Improving the Security of Wavelet-based Watermarking Systems

Diplomarbeit
zur Erlangung des Diplomgrades
an der Naturwissenschaftlichen Fakultät
der Universität Salzburg

eingereicht von

Werner Michael Dietl

Salzburg, am 10. Dezember 2002

Acknowledgments

Deepest thanks to all the members of my family and to all the people who made me the person that I am today.
Special thanks to my advisor Andreas Uhl, without whom this thesis would not exist. Peter Meerwald got me interested in the field of watermarking and without the countless discussions with him and his friendship I would have never finished this thesis. Thanks a lot, Peter! Thanks to all the members of the Ganesh group for answering my questions and supporting me, especially to Andreas Pommer.
I am deeply indebted to all my friends around the world that gave me emotional support and were there to listen to me. Thanks to Anu, Steven, Kasey, Tuan, Corey, Bente and Daniela!
Part of the work on this thesis was funded by the FWF project P15170 “Sicherheit für Bilddaten in Waveletdarstellung”.

Contents

Abstract **xi**

1 Introduction **1**
1.1 Copyright Protection for the Information Society . . . 1
1.2 Wavelet Analysis . . . 2
1.2.1 Basic Wavelet Theory for Image Processing . . . 2
1.3 Watermarking . . . 7
1.3.1 Kim Algorithm . . . 8
1.3.2 Wang Algorithm . . . 10
1.3.3 Xia Algorithm . . . 11
1.3.4 Correlation and Implementation . . . 12
1.4 Robustness and Security . . . 13
1.4.1 Applied Assessments . . . 14

2 Filter Parametrization **17**
2.1 Previous Work . . . 17
2.2 Filter Parametrization . . . 18
2.3 Two Parameters . . . 18
2.3.1 Security Assessment . . . 18
2.3.2 Quality Assessment . . . 24
2.3.3 Parameter Quality Assessment . . . 27
2.4 Different Combinations of Six Parameters . . . 29
2.4.1 Security Assessment . . . 29
2.4.2 Quality Assessment . . . 46
2.5 1, 2, 3, 5, 6 and 9 Parameters . . . 51
2.6 Combined System . . . 55
2.6.1 Security Assessment . . . 55
2.6.2 Quality Assessment . . . 60
2.7 Example Images . . . 63
2.8 Conclusions . . . 67

3 Wavelet Packet Decomposition **69**
3.1 Theory and Previous Work . . . 69
3.2 Proposed Method . . . 70
3.2.1 Tree Decompositions . . . 71
3.2.2 Embedding Variations . . . 72
3.3 Security Assessment . . . 73
3.3.1 7 Levels, Watermark Length 1000 . . . 73
3.3.2 4 Levels, Watermark Length 1000 . . . 82
3.3.3 Watermark Length 5000 — Decomposition 2 . . . 84
3.3.4 Watermark Length 20000 — Decomposition 2 . . . 86
3.4 Quality Assessment . . . 88

3.4.1 7 Levels, Watermark Length 1000 . . . 89
3.4.2 4 Levels, Watermark Length 1000 . . . 95
3.4.3 7 Levels, Watermark Length 5000 . . . 97
3.4.4 4 Levels, Watermark Length 5000 . . . 100
3.4.5 7 Levels, Watermark Length 20000 . . . 101
3.4.6 4 Levels, Watermark Length 20000 . . . 104
3.4.7 Length Comparison . . . 106
3.4.8 Levels Comparison . . . 108
3.4.9 Picture Comparison . . . 110
3.5 Example Decompositions and Images . . . 112
3.5.1 Tree Decompositions . . . 112
3.5.2 7 Levels, Watermark Length 1000 . . . 114
3.5.3 4 Levels, Watermark Length 1000 . . . 118
3.5.4 7 Levels, Watermark Length 5000 . . . 119
3.5.5 4 Levels, Watermark Length 5000 . . . 120
3.5.6 7 Levels, Watermark Length 20000 . . . 121
3.5.7 4 Levels, Watermark Length 20000 . . . 122
3.6 Conclusions . . . 123

4 Attacks on Watermark Coefficients 125
4.1 Introduction . . . 125
4.2 Description of Attacks . . . 125
4.2.1 Setting Coefficients to Zero . . . 126
4.2.2 Fixed Quantization of Coefficients . . . 126
4.2.3 10% Scaling of Coefficients . . . 126
4.2.4 Barni-Lewis Perceptual Quantization of Coefficients . . . 126
4.3 Attacks on Watermarks . . . 126
4.3.1 Setting Coefficients to Zero . . . 126
4.3.2 Fixed Quantization of Coefficients . . . 130
4.3.3 10% Scaling of Coefficients . . . 133
4.3.4 Barni-Lewis Perceptual Quantization of Coefficients . . . 134
4.4 Example Images . . . 135
4.5 Conclusions . . . 138

A Example Images 139
A.1 Original Images . . . 140
A.2 Compressed Lena . . . 141
A.3 Compressed Barbara . . . 144
A.4 Decomposed Lena . . . 147
A.5 Decomposed Barbara . . . 148

B Development Environment 149

Bibliography 151

Curriculum Vitae 157

List of Figures

1.1 One decomposition step of an input image into four sub-images 3
1.2 One decomposition step 4
1.3 One composition step of the four sub-images into one output image 4
1.4 Three decomposition steps of an input image into 10 sub-images 4
1.5 Pyramid after three decomposition steps 5
1.6 Composing the original image from the sub-images 5
1.7 Decomposing "Lena" 1, 2 and 3 times 5
1.8 Tiling of the time-frequency domain 6

2.1 Overview of watermark embedding procedure using parametrized wavelet filters . . 18
2.2 Comparison of Daubechies 6 and parametrized filter 19
2.3 Kim security evaluation overview 20
2.4 Kim security evaluation details 21
2.5 Wang security evaluation 22
2.6 Xia security evaluation 23
2.7 Kim compression behavior 24
2.8 Wang compression behavior 25
2.9 Xia compression behavior 26
2.10 Kim parameter quality assessment 27
2.11 Wang parameter quality assessment 28
2.12 Xia parameter quality assessment 28
2.13 Kim — One filter with six parameters — Overview 30
2.14 Kim — One filter with six parameters — Details 31
2.15 Kim — Six filters each with one parameter — Overview 32
2.16 Kim — Six filters each with one parameter — Details 33
2.17 Kim — Mix 3-2-1 — Overview 34
2.18 Kim — Mix 3-2-1 — Details 35
2.19 Kim — Mix 1-3-2 — Overview 36
2.20 Kim — Mix 1-3-2 — Details 37
2.21 Wang — One filter with six parameters — Overview 38
2.22 Wang — One filter with six parameters — Details 39
2.23 Wang — Six filters each with one parameter — Overview 40
2.24 Wang — Six filters each with one parameter — Details 41
2.25 Wang — Mix 3-2-1 — Overview 42
2.26 Wang — Mix 3-2-1 — Details 43
2.27 Wang — Mix 1-3-2 — Overview 44
2.28 Wang — Mix 1-3-2 — Details 45
2.29 Kim — Correlation and PSNR under JPEG2000 compression — Mixed 46
2.30 Kim — Correlation and PSNR under JPEG2000 compression — Pure 47
2.31 Kim — Correlation and PSNR under JPEG2000 compression — One filter 47
2.32 Kim — Correlation and PSNR under JPEG2000 compression — Six filters 47
2.33 Wang — Correlation and PSNR under JPEG2000 compression — Mixed 48

2.34 Wang — Correlation and PSNR under JPEG2000 compression — Pure 49
2.35 Wang — Correlation and PSNR under JPEG compression — Mixed 49
2.36 Wang — Correlation and PSNR under JPEG compression — Pure 49
2.37 Wang — Correlation and PSNR under JPEG2000 compression — One filter 50
2.38 Wang — Correlation and PSNR under JPEG2000 compression — Six filters 50
2.39 Correlation and PSNR under JPEG2000 compression 53
2.40 Correlation and PSNR under JPEG compression 54
2.41 Attacks on all four levels 56
2.42 Variations of all levels 57
2.43 Variations of single parameters 58
2.44 Kim — Attacks on first and second level 59
2.45 Average correlation under JPEG2000 and JPEG compression 60
2.46 Minimum correlation under JPEG2000 and JPEG compression 61
2.47 Maximum correlation under JPEG2000 and JPEG compression 61
2.48 Average PSNR under JPEG2000 and JPEG compression 61
2.49 Comparison of correlation and PSNR under JPEG2000 compression 62
2.50 Comparison of correlation and PSNR under JPEG compression 62
2.51 One Parameter: 1.90 63
2.52 Three Parameters: 1.50 -2.25 2.25 64
2.53 Five Parameters: 1.00 2.00 -1.75 0.75 -1.25 64
2.54 Nine Parameters – Working 65
2.55 Nine Parameters – Broken 65
2.56 20 Parameters – Example 1 66
2.57 20 Parameters – Example 2 66

3.1 System design of two complete decomposition steps 69
3.2 Subband structure after two complete decomposition steps 70
3.3 Basic system design 71
3.4 Decomposition 1, 7 levels, watermark length 1000, no variation 73
3.5 Decomposition 1, 7 levels, watermark length 1000, variation 1 74
3.6 Decomposition 1, 7 levels, watermark length 1000, variation 2 75
3.7 Decomposition 1, 7 levels, watermark length 1000, variation 3 76
3.8 Decomposition 1, 7 levels, watermark length 1000, variation 1 & 3 77
3.9 Decomposition 2, 7 levels, watermark length 1000, no variation 78
3.10 Decomposition 2, 7 levels, watermark length 1000, variation 1 79
3.11 Decomposition 2, 7 levels, watermark length 1000, variation 2 80
3.12 Decomposition 2, 7 levels, watermark length 1000, variation 3 81
3.13 Decomposition 2, 7 levels, watermark length 1000, variation 1 & 3 81
3.14 Decomposition 1, 4 levels, watermark length 1000, no variation 82
3.15 Decomposition 1, 4 levels, watermark length 1000, variation 1 & 3 82
3.16 Decomposition 2, 4 levels, watermark length 1000, no variation 83
3.17 Decomposition 2, 4 levels, watermark length 1000, variation 1 & 3 83
3.18 Decomposition 2, 7 levels, watermark length 5000, no variation 84
3.19 Decomposition 2, 7 levels, watermark length 5000, variation 1 & 3 84
3.20 Decomposition 2, 4 levels, watermark length 5000, no variation 85
3.21 Decomposition 2, 4 levels, watermark length 5000, variation 1 & 3 85
3.22 Decomposition 2, 7 levels, watermark length 20000, no variation 86
3.23 Decomposition 2, 7 levels, watermark length 20000, variation 1 & 3 86
3.24 Decomposition 2, 4 levels, watermark length 20000, no variation 87
3.25 Decomposition 2, 4 levels, watermark length 20000, variation 1 & 3 87
3.26 Lena: 7 levels, watermark length 1000, comparison of methods 89
3.27 Lena: decomposition 1, 7 levels, watermark length 1000, no variation 90
3.28 Lena: decomposition 2, 7 levels, watermark length 1000, no variation 91
3.29 Lena: decomposition 2, 7 levels, watermark length 1000, comparison of variations . 92

3.30 Barbara: 7 levels, watermark length 1000, comparison of methods 93
3.31 Barbara: decomposition 2, 7 levels, watermark length 1000, no variation 94
3.32 Lena: 4 levels, watermark length 1000, comparison of methods 95
3.33 Lena: decomposition 2, 4 levels, watermark length 1000, no variation 96
3.34 Lena: 7 levels, watermark length 5000, comparison of methods 97
3.35 Lena: decomposition 2, 7 levels, watermark length 5000, no variation 98
3.36 Barbara: 7 levels, watermark length 5000, comparison of methods 99
3.37 Lena: 4 levels, watermark length 5000, comparison of methods 100
3.38 Lena: 7 levels, watermark length 20000, comparison of methods 101
3.39 Lena: decomposition 2, 7 levels, watermark length 20000, no variation 102
3.40 Barbara: 7 levels, watermark length 20000, comparison of methods 103
3.41 Lena: 4 levels, watermark length 20000, comparison of methods 104
3.42 Lena: decomposition 2, 4 levels, watermark length 20000, no variation 105
3.43 Lena: 7 levels, length comparison 106
3.44 Barbara: 7 levels, length comparison 107
3.45 Lena: watermark length 1000, levels comparison 108
3.46 Lena: watermark length 5000, levels comparison 108
3.47 Lena: watermark length 20000, levels comparison 109
3.48 Decomposition 1, picture comparison 110
3.49 Decomposition 2, picture comparison 111
3.50 Decomposition 1, 4 levels 112
3.51 Decomposition 2, 4 levels 112
3.52 Decomposition 1, 7 levels 113
3.53 Decomposition 2, 7 levels 113
3.54 Lena: decomposition 1, tree 150000, no variation 114
3.55 Lena: decomposition 1, tree 150000, coefficient skipping 114
3.56 Lena: decomposition 1, tree 150000, watermark shuffling 115
3.57 Lena: decomposition 1, tree 150000, skipping and shuffling 115
3.58 Lena: decomposition 2, tree 150000, no variation 115
3.59 Lena: decomposition 2, tree 150000, coefficient skipping 116
3.60 Lena: decomposition 2, tree 150000, watermark shuffling 116
3.61 Lena: decomposition 2, tree 150000, skipping and shuffling 116
3.62 Lena: decomposition 2, tree 200000, coefficient skipping 117
3.63 Barbara: decomposition 2, tree 150000, no variation 117
3.64 Lena: decomposition 1, tree 150000, coefficient skipping 118
3.65 Lena: decomposition 2, tree 150000, coefficient skipping 118
3.66 Lena: decomposition 1, tree 150000, coefficient skipping 119
3.67 Lena: decomposition 2, tree 150000, coefficient skipping 119
3.68 Lena: decomposition 1, tree 150000, coefficient skipping 120
3.69 Lena: decomposition 2, tree 150000, coefficient skipping 120
3.70 Lena: decomposition 1, tree 150000, coefficient skipping 121
3.71 Lena: decomposition 2, tree 150000, coefficient skipping 121
3.72 Lena: decomposition 1, tree 150000, coefficient skipping 122
3.73 Lena: decomposition 2, tree 150000, coefficient skipping 122

4.1 Attack with Biorthogonal 7/9; Zeroing coefficients 127
4.2 Attack with Biorthogonal 7/9; Zeroing coefficients; Correlation without reference . 128
4.3 Attack with Daubechies 6; Zeroing coefficients 129
4.4 Attack with Daubechies 6; Zeroing coefficients; Correlation without reference . . . 129
4.5 Attack with Biorthogonal 7/9; Quantization step size 100 130
4.6 Attack with Biorthogonal 7/9; Quantization step size 100; Correlation without reference 131
4.7 Attack with Daubechies 6; Quantization step size 100 132
4.8 Attack with Daubechies 6; Quantization step size 100; Correlation without reference 132

4.9 Attack with Biorthogonal 7/9; 10% Scaling; Without reference 133
4.10 Attack with Daubechies 6; 10% Scaling; Without reference 133
4.11 Attack with Biorthogonal 7/9; Perceptual Quantization; Without reference 134
4.12 Attack with Daubechies 6; Perceptual Quantization; Without reference 134
4.13 Lena after zeroing 100 coefficients; PSNR = 20.39 dB 135
4.14 Lena after zeroing 1000 coefficients; PSNR = 19.63 dB 135
4.15 Lena after zeroing 20000 coefficients; PSNR = 19.41 dB 136
4.16 Lena after quantizing 20000 coefficients with step size 100; PSNR = 33.08 dB . . . 136
4.17 Lena after scaling 20000 coefficients ±10%; PSNR = 35.94 dB 136
4.18 Lena after perceptually quantizing 20000 coefficients; PSNR = 35.45 dB 137

A.1 Uncompressed Lena; 512 x 512 pixels; grayscale, 8 bits per pixel 140
A.2 Uncompressed Barbara; 512 x 512 pixels; grayscale, 8 bits per pixel 140
A.3 Lena: PSNR under JPEG2000 Compression 141
A.4 Lena: PSNR under JPEG Compression 141
A.5 Lena under JPEG2000 compression 142
A.6 Lena under JPEG compression 143
A.7 Barbara: PSNR under JPEG2000 Compression 144
A.8 Barbara: PSNR under JPEG Compression 144
A.9 Barbara under JPEG2000 compression 145
A.10 Barbara under JPEG compression 146
A.11 Decomposed Lena 147
A.12 Decomposed Barbara 148

Abstract

Over the recent years research into copy protection schemes has seen widespread attention from both academia and content owners. The ease of content distribution over the Internet has allowed malicious users to quickly spread illegal copies of music, videos and images. Embedding invisible watermarks is used to prove the rightful owner of media and to deter illegitimate use.
In this thesis we present two methods to improve the security of wavelet-based watermarking systems and analyze their properties in detail. In chapter 1 we give an introduction to watermarking, wavelet analysis and describe the watermarking algorithms we use.
Chapter 2 introduces filter parametrization with between 2 and 20 parameters and uses non-stationary multi-resolution analysis to get a secure keyspace. The robustness to JPEG and JPEG2000 compression and the security against unauthorized detection is assessed.
Chapter 3 describes how random wavelet packet decompositions can be used to increase the security. We describe two procedures to create random wavelet packet trees and three embedding variations and analyze the security and robustness of the proposed system.
Finally, in chapter 4, we perform malicious attacks on the wavelet coefficients and try to remove the watermarks. We compare the behavior of the two proposed systems with the standard systems that use the pyramidal decomposition and either the Biorthogonal 7/9 or the Daubechies 6 filter. Both proposed systems show superior performance.
We conclude the thesis with appendix A, example images, appendix B, a description of the development environment, the bibliography and a curriculum vitae.

Parts of chapter 2 have been presented at the following conference and have been published in the conference proceedings:

> Werner Dietl, Peter Meerwald, and Andreas Uhl.
> Watermark security via high-resolution wavelet filter parametrization.
> In Kmet' Stanislav and Pavluš Miron, editors,
> *Proceedings of 7th International Scientific Conference, Section 1: Applied Mathematics*,
> pages 21–28, Košice, Slovakia, May 2002.

The following paper will be presented in January 2003 and will appear in the conference proceedings:

> Werner Dietl, Peter Meerwald, and Andreas Uhl.
> Key-dependent pyramidal wavelet domains for secure watermark embedding.
> In Edward J. Delp and Ping Wah Wong, editors,
> *Proceedings of SPIE Volume 5020*, Santa Clara, CA, USA, January 2003.

Papers based on other parts are currently being reviewed or may be published in the future.

Chapter 1

Introduction

1.1 Copyright Protection for the Information Society

The protection of Intellectual Property Rights has become one of the main interests of content owners. Many different terms are in use: Intellectual Property Rights IPR, Digital Rights Management DRM, Copyright Protection, Access Control Schemes and others.
The basic goal of the content owners is to regain their power over the media they own.
In the time of analog reproduction private copies of media were not a big problem. The quality of analog copies degraded significantly and therefore widespread distribution of works did not occur. Digital technology changed this. The use of the Internet and the availability of powerful personal computers made it possible for the layman to download digital content from the Internet, manipulate it on the local machine and redistribute it. Peer-to-Peer systems allow easy and fast distribution of works over the Internet.
This has effects on all kinds of media: audio, images and videos.
Over the last several years a widespread discussion about the consumers' and producers' rights has occured. No real solution has been found yet and the very political discussion is likely to continue for some time.
An overview of the involved parties for multimedia protection in consumer electronics is given in [30, 31].
The audio industry claims to be the hardest hit by the digital revolution. Organizations like the Recording Industry Association of America (RIAA, `http://www.riaa.com/`) sue Peer-to-Peer networks like Napster and try to shut down the illegal distribution of digital music. Technologies like the Secure Digital Music Initiative (SDMI, `http://www.sdmi.org/`) were developed for copyright protection of digital audio. [18, 99] show successful attacks on this system.
The movie industry is equally concerned about the widespread illegal distribution of movies over the Internet. The introduction of DVDs, large storage capacities, high-speed Internet-connections and advanced audio and video coders made the distribution of full-length movies over the Internet possible. Organisations like the Motion Picture Association of America (MPAA, `http://www.mpaa.org/`) are trying to protect their members' Intellectual Properties Rights. Technology like the Content Scramble System CSS for DVDs were already hacked and software is available to decode encrypted movies.
The protection of still images is in the interest of the copyright holders, stock photography and image providers and image galleries. Examples include Corbis (`http://www.corbis.com/`), the State Hermitage Museum (`http://www.hermitagemuseum.org/`) or the Vatican Library (`http://www-3.ibm.com/software/is/dig-lib/vatican/manuscript.html`, `http://www.vatican.va/`).
The legal aspects are gaining importance as well. The Digital Millennium Copyright Act DMCA in the USA and the European Union Copyright Directive (`http://uk.eurorights.org/issues/eucd/`) are both intended to better protect the copyright in this time of digital technologies.
The Electronic Frontier Foundation (EFF, `http://www.eff.org/`) has many news and references

about privacy, intellectual property and security in the digital age. Especially the EFF "Intellectual Property Online: Patent, Trademark, Copyright" archive at `http://www.eff.org/IP/` has a lot of information and many references to copyright protection systems.
In this thesis we focus our attention to increasing the security of invisible still-image watermarking systems that work in the wavelet domain. The remainder of this chapter gives an introduction to the technologies that are used. Chapters 2 and 3 explain two methods for increasing the security of watermarks and chapter 4 investigates the robustness to attacks on the wavelet coefficients.

1.2 Wavelet Analysis

Over the last few decades wavelet analysis emerged from a variety of different fields like signal processing, digital speech recognition, seismology and pyramidal image processing.
Many books have been written about wavelet analysis and books about different application areas usually have sections on the basics of Wavelets and how they can be applied.
The following books assume different backgrounds and use different application areas for introducing and using wavelets:

- Daubechies' "Ten Lectures on Wavelets" [22] and Mallat's "Wavelet Tour of Signal Processing" [59] are two of the classic books that introduce Wavelets from a mathematical point of view.
- [38] introduces theory and algorithms for Wavelets and also shows some applications.
- [1] describes multiresolution signal decomposition methods including Wavelets. [10] shows the use of Wavelets for signal analysis.
- [98] introduces the theory of Wavelets and shows how they can be implemented in software.
- [63] looks at the connection between filter banks and Wavelets.
- One very important application area for Wavelets is the new JPEG2000 still image compression standard. [86] introduces all components of JPEG2000 and also contains good information about Wavelets.
- [50] is a collection of case studies that show different uses for Wavelets.
- Image processing and computer graphics is another application area for Wavelets. Books like [8], [37] and [90] contain chapters about the use of Wavelets and [79] and [83] are dedicated to the application of Wavelets to this area.

All of these books include extensive bibliographies and should make it easy to find further references. A quick search on Amazon returns more than 150 books about wavelet analysis and there is a countless number of research papers about Wavelets and their different applications. Of course the Internet is a good source for finding information as well. Three possible starting points are `http://www.mathsoft.com/wavelets.html`, `http://www.wavelet.org/links.html` and `http://www.cosy.sbg.ac.at/~uhl/wav.html`.

1.2.1 Basic Wavelet Theory for Image Processing

Because we do not need all the details of wavelet theory we will only give a very application oriented introduction to the wavelet transformation, how to use it and some of the important facts about it. This introduction should be enough to understand the use of Wavelets for image watermarking. An introduction to wavelet filter parametrization can be found in section 2.2 and the wavelet packet decomposition is described in section 3.1.
The name *Wavelet* is used because of the roots in seismology and at the core of the analysis are functions with finite support that look like waves. These functions are used to decompose a signal and represent the information in a compact way.
Some commonly used abbreviations are

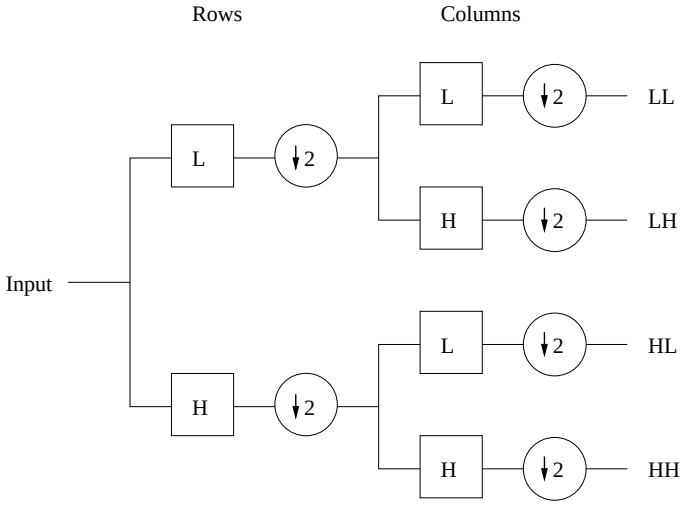


Figure 1.1: One decomposition step of an input image into four sub-images

CWT Continuous Wavelet Transformation: General form of wavelet analysis operating on continuous functions.

DWT Discrete Wavelet Transformation: Transformation of sampled data, e.g. values in an array, into wavelet coefficients.

IDWT Inverse Discrete Wavelet Transformation: The inverse procedure that converts wavelet coefficients into the original sampled data.

FWT Fast Wavelet Transformation: A fast method for the DWT.

We introduce Wavelets for two dimensional images, because that is the application area we need. Let us limit our attention to square sized images with a width that is a power of two, i.e. N by N pixel images with $N = 2^n$. We also limit our observations to gray-level images with 256 grays, i.e. we need 8 bits or 1 byte for every image pixel. Color images can be analyzed on a per-plane basis, either in RGB, CMYK or another color domain.
For us it suffices to think of the Wavelet analysis as the application of special high- and low-pass filters that separate the original image into four sub-images that contain the different frequency components. For an N by N image this process generates N^2 wavelet coefficients.
Figure 1.1 shows the first decomposition step. The image is high- and low-pass filtered along the rows and the results of each filter are down-sampled by two. Those two sub-signals correspond to the high- and low-frequency components along the rows and are each of size N by N/2. Each of those sub-signals is then again high- and low-pass filtered, but this time along the column data. The results are again down-sampled by two.
In this way the original data is split into four sub-images each of size N/2 by N/2 containing information from different frequency components. Figure 1.2 shows the four sub-bands in the typical arrangement.
The LL subband is the result of low-pass filtering both the rows and columns and contains a rough description of the image. Therefore the LL subband is also called the approximation subband.
The HH subband was high-pass filtered in both directions and contains the high-frequency components along the diagonals. The HL and LH images are the result of low-pass filtering in one direction and high-pass filtering in the other direction. LH contains mostly the vertical detail information, which corresponds to horizontal edges. HL represents the horizontal detail information from the vertical edges. All three subbands HL, LH and HH are called the detail subbands, because they add the high-frequency detail to the approximation image.

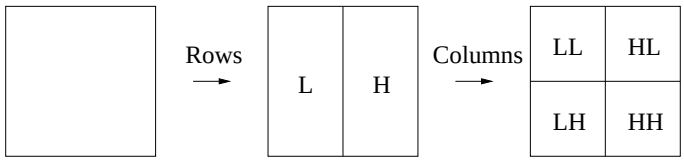


Figure 1.2: One decomposition step

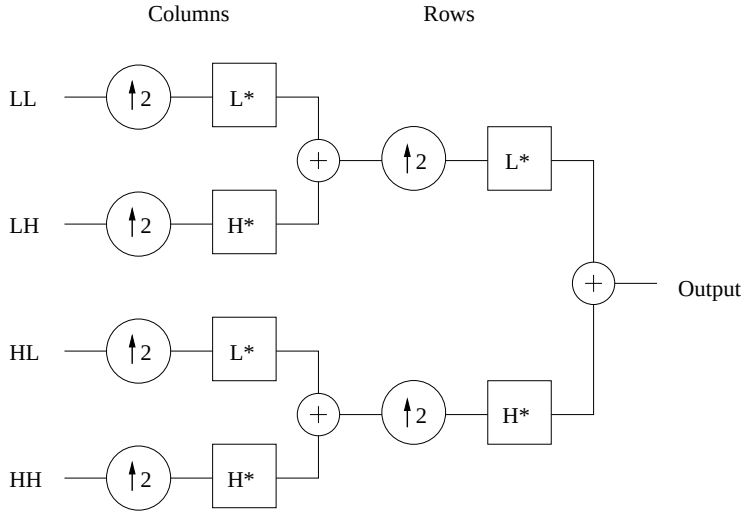


Figure 1.3: One composition step of the four sub-images into one output image

The inverse process is shown in figure 1.3. The information from the four sub-images is up-sampled and then filtered with the corresponding inverse filters along the columns. The two results that belong together are added and then again up-sampled and filtered with the corresponding inverse filters. The result of the last step is added together and we have the original image again.
Note that there is no loss of information when the image is decomposed and then composed again at full precision.
Different filters can be used for the decomposition step. A short introduction about filters can be found in section 2.2.
Usually the image is decomposed more than one time. The simplest and most common way is the pyramidal decomposition that we will explain here. An alternative method is the Wavelet Packet decomposition that will be explained in section 3.1.
For the pyramidal decomposition we only apply further decompositions to the LL subband. Figure 1.4 shows a systematic diagram of three decomposition steps. For simplicity the decomposition from figure 1.1 is the DWT box with one input and four output images. At each level the detail subbands are the final results and only the approximation subband is further decomposed.

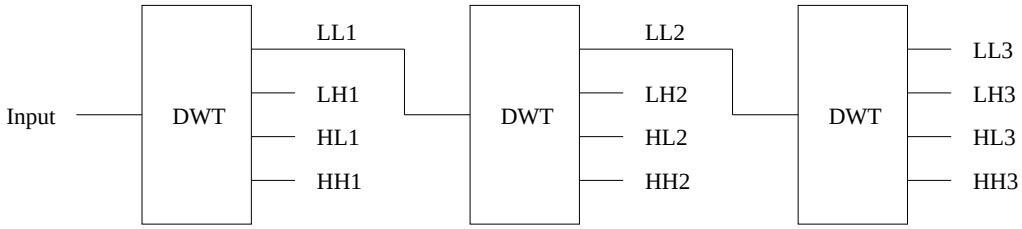


Figure 1.4: Three decomposition steps of an input image into 10 sub-images

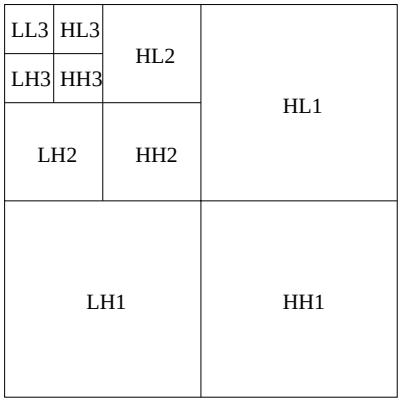


Figure 1.5: Pyramid after three decomposition steps

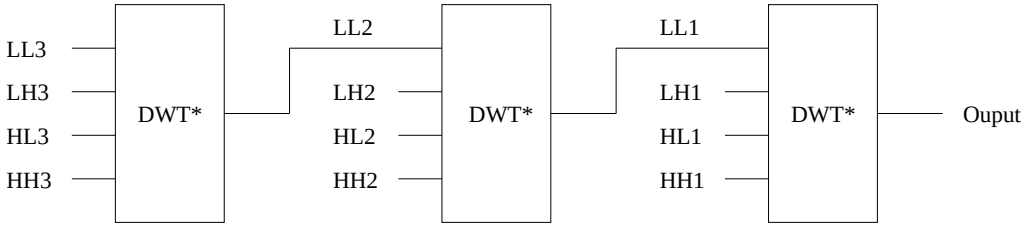
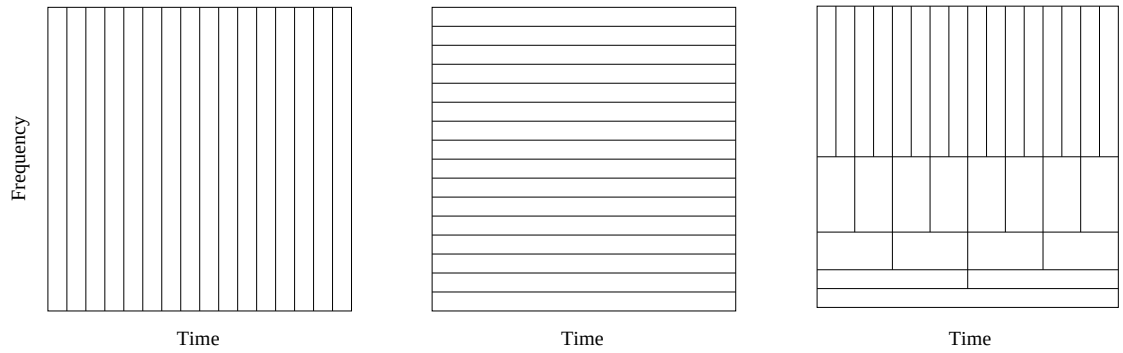


Figure 1.6: Composing the original image from the sub-images

Figure 1.5 shows the pyramidal structure that results from this decomposition. At the lowest level there is one approximation subband and there are a total of nine detail subbands at the different levels. After L decompositions we have a total of $D(L) = 3 * L + 1$ subbands.
The composition to a complete image is shown in figure 1.6. The DWT* box is the composition from figure 1.3 used as a basic building block.
Figure 1.7 is an example of this decomposition process. It shows the "Lena" image after one, two and three pyramidal decomposition steps using the standard Biorthogonal 7/9 filter. The original "Lena" image and other example decompositions can be seen in Appendix A.
In summary the advantages of the wavelet analysis are:

Multiresolution analysis After the wavelet transformation we can access the data at different resolutions and thereby focus our attention to different levels of detail. Fine detail can be analyzed in the detail subbands, whereas big image components can be detected in the approximation image and in the detail information at different levels.

Figure 1.7: Decomposing "Lena" 1, 2 and 3 times



(a) Sampled Data (b) FFT (c) FWT

Figure 1.8: Tiling of the time-frequency domain [37]

Better time-frequency localisation In figure 1.8 we show the one-dimensional time-frequency correspondence for sampled data, the Fast Fourier Transform FFT and the Fast Wavelet Transform FWT.

For sampled data we have the complete information for every time location, but no frequency information.

The FFT gives the frequency information of the signal, but we lose the location of the occurences.

In figure 1.8(c) we see the tiling for the FWT with a standard pyramidal decomposition. For the high-frequency components we have a fine time-resolution, but only for a wider frequency band. With every subband division we get finer resolution of the frequencies, but lose the time resolution. The lowest subband gives detailed information about the lowest frequency for the complete time axis.

The same is true for two-dimensional data. The detail subbands at the highest level have the best localization of the frequency information, but only for a large frequency band. Subbands at lower decomposition levels have better frequency resolution, but lose the spatial resolution. The wavelet packet decomposition allows for an adaptation to the specific requirements of the signal.

Human Visual System HVS Wavelet analysis has properties that make it easy to model the HVS. The block-based DCT analysis used in the JPEG compression results in block artifacts at high compression rates. Wavelet analysis is used for the JPEG2000 standard and has less visible artifacts even at high compression rates.

Energy compaction The approximation subband contains the highest energy. Most of the detail coefficients are close to zero and can therefore be quantized easily. This property is very important for compression applications.

Lower time complexity The Fast Fourier Transformation FFT of n elements is $O(n\,log_2(n))$. In contrast to that the Fast Wavelet Transformation FWT is only $O(n)$.

This is just a very basic introduction to the wavelet analysis. There are many other topics that need to be addressed for a working system. But for our understanding of the watermarking methods this should be enough for now.

1.3 Watermarking

Beginning in the early 1990's the area of watermarking began to surface in the research community. Since then it has become a very active field with hundreds of publications each year and many conferences dedicated solely to watermarking. Table 1.1 shows an overview of the number of publications over the last decade [69, 28, 60].

Year	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001
Publications	2	2	4	13	29	64	103	200+	250+	300+

Table 1.1: Number of digital watermarking related publications according to [69, 28, 60] and our own INSPEC research

Digital watermarking is also a very multi-disciplinary research area and people with backgrounds in cryptography, information theory, image processing, the human visual system and other areas have contributed to its development.
There is a number of recent publications that give a survey of the state of digital watermarking and show possible applications [46, 69, 85, 16, 11, 64, 65, 42].
One recent book that gives a great introduction to many different areas of watermarking is [17] by Cox, Miller and Bloom. [24, 43] look at different aspects of watermarking, its history, existing tools, attacks and countermeasures. For an introduction to cryptography see [81, 61], two popular books on the subject.
Cox et al. [17] define *Watermarking* as "the practice of imperceptibly altering a Work to embed a message about that Work". The general term *Work* is used to mean any media object, including images, videos and music.
Let us quickly introduce some terms and properties. For a discussion see e.g. [70, 69, 82, 16, 17].

- The most general term used is *Information Hiding*, which encompasses all methods for (more or less) covert communication.
- *Steganography*, which comes from the Greek words "steganos graphia" which mean "hidden writing", is one form of information hiding. The aim of steganography is to hide the fact that a communication is taking place at all [33, 71]. High channel capacity is very important and in most cases robustness to attacks is less important.
- *Fragile Watermarking* is used to verify that a Work is not modified [93, 56, 2]. It is designed to be robust against some simple image modifications like light compression or channel noise, but not to be robust to content modifications. It is usually desirable to be able to determine the regions that were changed.
- *Fingerprinting* is used to invisibly mark multiple copies of one Work. The mark is used to determine the recipient of the Work. In case of copyright infringement the fingerprint can be used to determine the adversary. Fingerprinting can be seen as a special case of watermarking where unique identification numbers are embedded into multiple copies of the same Work. High robustness to attacks is desirable, especially against collusion attacks that use multiple copies of the same Work to remove the watermark [6, 25, 23, 62]. The capacity is less important, because only a unique user identification has to be embedded.

There are different ways to distinguish between watermarking systems. The first characteristic is whether the watermark is visible or not:

- What we are looking at in this thesis is *Imperceptible Watermarking*. The goal is to robustly embed information into a Work in a (for the human user) invisible way. The measure of invisibility can either be experimental with a group of people or automated calculations like the Peak Signal to Noise Ratio PSNR.

- In contrast *Visible Watermarking* is used in some applications. The watermark could be a company logo or copyright statement that is robustly embedded into a Work. For example the watermarking system from IBM (`http://www-3.ibm.com/software/is/dig-lib/vatican/manuscript.html`), that is used for the Vatican Library (`http://www.vatican.va/`), embeds a logo of the Vatican in images.

Another characteristic is whether we need the original Work or not:

- *Blind Watermarking* means that the watermarking system does not need the original Work to extract the watermark. For many application areas this is the only possibility, because it is not feasible to get access to the original Work, e.g. in video watermarking.
- In *Non-Blind Watermarking* we have the original Work at our disposal. Being able to use the original allows for more robust watermarks, because we can more easily correct modifications an attacker has performed.

Finally we can differentiate the domain that is used for embedding the watermark:

- *Spatial-Domain Watermarking* embeds the watermark directly into the cover data. Examples include modifying the Least Significant Bit LSB of every image pixel [9, 36, 53] or modifying the brightness of random pixel pairs. Because the watermark information is embedded in the insignificant bits of the cover the effects of simple manipulations like lossy compression are usually severe.
- In contrast *Transform-Domain Watermarking* applies the modifications after the application of a transformation. Examples are the Discrete Cosine Transformation DCT [12, 13] or the wavelet transformation [48, 96, 100].

Other properties of watermarking systems according to [17] are the embedding effectiveness, the fidelity, the data payload, the false positive rate, the robustness, the security, the cipher and watermark keys that are used, the ability to modify the watermark or embed multiple watermarks and the cost of the watermarking system.
Two very important concepts were first introduced by Cox et al. [12, 13, 14]. Firstly, in order to be robust against attacks the watermark should be embedded into the perceptually most significant components of a Work. This ensures that severe modifications are necessary to remove the watermark and that the Work will most likely lose its value by attacks.
Secondly, to ensure that embedding a watermark in the most significant parts of a Work does not leave perceptible artifacts, concepts from spread spectrum communication are used. In spread spectrum communication a narrow information band is spread over a wide carrier band. This minimizes the modifications per frequency that are necessary and also increases the security, because without the knowledge of the spreading function the signal is very hard to detect or remove. For an introduction to spread spectrum communications see [49].
In their work Cox et al. embed the watermark in the discrete cosine transform domain, but since then many algorithms have been developed that use other transformations, for example the wavelet transformation or the log-polar Fourier transformation.
Peter Meerwald's master thesis [60] has an overview of digital image watermarking in the wavelet transform domain and describes many algorithms that embed watermarks using the wavelet domain. We choose three algorithms proposed by Kim, Wang and Xia and look at methods to improve their security. All three algorithms are non-blind, imperceptible watermarking systems that embed a random sequence in the wavelet coefficients of the image. For easy reference to the original papers we keep the respective notations. Let us first have a detailed look at those three algorithms.

1.3.1 Kim Algorithm

The first method we want to decribe is from Jong Ryul Kim and Young Shik Moon [48].
A watermark is embedded by the following procedure:

- The watermark is a gaussian-distributed random vector X_i of length 1000. We use a mean of zero and a standard deviation of one.
- The original image is wavelet transformed into 3 levels using a biorthogonal filter.
- For each subband a threshold value is computed depending on the decomposition level. First the largest coefficient C_i of the three detail subbands is calculated. Then the corresponding threshold value T_i is calculated by

$$T_i = 2^{\lfloor log_2 C_i \rfloor - 1} \tag{1.1}$$

- This threshold value is used to determine the perceptually significant coefficients of the image. Only coefficients that are larger than the corresponding T_i are used for embedding the watermark.
- For each significant coefficient V_i the corresponding watermark elemente X_i is embedded by the modification

$$V_i' = V_i + \alpha V_i X_i \tag{1.2}$$

  This way the new wavelet coefficient V_i' depends on the original value V_i and a scaled version of the watermark element X_i. Multiplication of X_i with V_i adds a dependency of the watermark modification on the current coefficient value.
- α is a level-dependent scale factor to influence the watermark strength. For the aproximation subband LL a small scale factor of 0.04 is used. For the detail subbands values of 0.1, 0.2 and 0.4 are used for the third, second and first level of the decomposition.
- After all watermark elements have been embedded into the significant coefficients the inverse wavelet transformation is applied to the modified coefficients to produce the watermarked image.

To extract the watermark from a possibly attacked image we follow the following steps:

- First we calculate the same three-level wavelet transformation of the original and of the watermarked images.
- Then we subtract the wavelet coefficients of the original image from the coefficients of the watermarked image.
- The remaining values are the modifications that were added by the watermark and any possible modifications from attacks on the watermark.
- We now measure the similarity between the extracted watermark X^* and the embedded watermark X by one of two methods proposed in [48].

  The first method is a vector projection defined by

$$sim_1(X, X^*) = \frac{X \cdot X^*}{\sqrt{X^* \cdot X^*}} \tag{1.3}$$

- The second measure is a normalization of the previous value and corresponds to a percentage of similarity between the two watermarks

$$sim_2(X, X^*) = \frac{X \cdot X^*}{\sqrt{X^* \cdot X^*}} \Big/ \frac{X \cdot X}{\sqrt{X \cdot X}} \times 100 \tag{1.4}$$

- If the similarity measure is larger than a specified threshold then we say that the watermark was present in the image.

[48] presents invisibility and robustness results. This method shows superior invisibility properties and higher robustness to the examined attacks.

1.3.2 Wang Algorithm

The next method is from Houng-Jyh Wang and C.-C. Jay Kuo [96].
The embedding procedure is:

- The watermark is again a gaussian distributed random vector W_k with a mean of zero and a standard deviation of one. As upper bound for the watermark length they suggest half the number of wavelet coefficients. For their experiments they use $\frac{1}{32}$ of all coefficients.
- The search for the significant wavelet coefficients is based on Successive Subband Quantization (SSQ) from the Multi-Threshold Wavelet Codec (MTWC) which is from the same authors [94, 95].
- For each subband s we calculate the maximum absolute coefficient value $C_{max,s}$ and initialize the subband threshold $T_s = C_{max,s}/2$.
- For every wavelet coefficient we have to store whether that coefficient was already used to embed a watermark element. In the beginning we set all coefficients to be unselected.
- For every subband we have a weighting factor β_s that determines whether a subband should be used for embedding watermark information and whether some subbands should be used with a higher priority. The authors use 1.0 for all subbands except for the approximation subband. No information is embedded in the approximation subband.
- The user-adjustable factor α_s is used to control the strength of the watermark and influences the visibility and robustness of the algorithm.
- Now to embed the watermark sequence into the wavelet coefficients we first find the subband with the highest threshold value $\beta_s \times T_s$. Within this subband we use all unselected coefficients that are larger than T_s to embed a watermark element according to

$$C'_{s,k}(x,y) = C_s(x,y) + \alpha_s \beta_s T_s W_k \tag{1.5}$$

  and set the corresponding coefficient to be selected.
- Once there are no more unselected coefficients larger than the threshold T_s we change the threshold to $T_s^{new} = T_s/2$ and start again by looking for the subband with the highest threshold.
- This process is repeated until all the watermark elements are embedded into the wavelet coefficients. Then we apply the inverse wavelet transformation to the changed coefficients to get the watermarked image.

The watermark detection works as follows:

- We apply the wavelet analysis to both the original image I and the attacked image I^*.
- For every wavelet coefficient we calculate the difference to the original value

$$E^*_{s,k}(x,y) = C^*_{s,k}(x,y) - C_s(x,y) \tag{1.6}$$

- We know that the difference introduced by the watermarking procedure is

$$E_{s,k}(x,y) = \alpha_s \beta_s T_s W_k \tag{1.7}$$

- As similarity measure we use the normalized correlation between the embedded and the extracted watermarks multiplied by the number of watermark elements N_w

$$sim(I^*, I) = N_w \frac{\sum_{k=1}^{N_w} E^*_{s,k}(x,y) \cdot E_{s,k}(x,y)}{\|E^*_{s,k}(x,y)\| \, \|E_{s,k}(x,y)\|} \tag{1.8}$$

- Depending on $sim(I^*, I)$ and a threshold value we decide whether there is a watermark in the image or not.

The paper goes on to discuss methods to protect the embedded watermark.

1. The transformation structure that was used for the wavelet transformation can be kept secret. E.g. whether a pyramidal decomposition or a wavelet packet decomposition was used and how many levels have been decomposed.
2. The wavelet filter that was used for the decomposition could be used as key.
3. A seed number for significant coefficient skipping could introduce randomness.

In later chapters we will investigate filter parametrization to get a large space of possible filters and the use of randomly generated wavelet packets to get different decomposition structures. We also use significant coefficient skipping as one way to increase the security of our wavelet packet system.
The paper also discusses how the initial threshold values can be hidden and gives experimental evidence of the robustness of the proposed system.

1.3.3 Xia Algorithm

The last method we want to present in detail is from Xiang-Gen Xia, Charles G. Boncelet and Gonzalo R. Arce [100].
The embedding works like this:

- For an image x with dimensions m by n we first calculate the wavelet coefficients y[m, n]. The authors use the Haar wavelet and apply two decompositions steps.
- We have a watermark N[m, n] that is a gaussian pseudo-noise sequence with a mean of zero and a deviation of 1.
- The watermark is added to all detail subbands, but not to the approximation subband. The modification formula is

$$\tilde{y}[m,n] = y[m,n] + \alpha y^2[m,n] N[m,n] \tag{1.9}$$

  Again α is a user-defined scale factor that determines the embedding strength. The original coefficient value is squared to get higher influence from large coefficients.
- After all wavelet coefficients have been modified we apply the inverse transformation to get an intermediate image $\tilde{x}$.
- The last step is a correction of the dynamic range. For this the pixels are modified by

$$\hat{x}[m,n] = min\left(max(x[m,n]), max\left\{\tilde{x}[m,n], min(x[m,n])\right\}\right) \tag{1.10}$$

- $\hat{x}$ is the watermarked version of the image x.

The decoding process works in a hierarchical way:

- The original and watermarked image are not completely decomposed in the beginning. We only apply one decomposition step at a time.
- We calculate the difference between the HH_1 coefficients of the watermarked image and the original image and calculate the cross correlation to the original watermark.
- If there is a peak in the correlation then the watermark is detected and we can stop.
- If the correlation remains low, then we calculate the difference for the HH_1 and the LH_1 subbands and then calculate the correlation. Again, if there is a peak in the correlation we can stop.
- The same happens for the HH_1, LH_1 and HL_1 subbands. Only if we still do not have a peak in the correlation after checking all three detail subbands do we need to decompose the approximation subband LL_1 to get the HH_2, LH_2, HL_2 and LL_2 subbands.
- We continue this process of adding additional subbands until we get a peak in the correlation or run out of additional subbands. If we never find a peak in the correlation we have to conclude that there is no watermark in the image.

1.3.4 Correlation and Implementation

For all three algorithms we use the same normalized correlation measure between the extracted and the embedded watermark sequences. If X is the array of length n that contains the watermark that was embedded and Y is the watermark that was extracted from the image then we calculate

$$sim(X,Y) = \frac{X\,Y}{\sqrt{X^2\,Y^2}} = \frac{\sum_{i=0}^{n} X_i\,Y_i}{\sqrt{\sum_{i=0}^{n} X_i^2 \sum_{i=0}^{n} Y_i^2}} \tag{1.11}$$

For the Xia algorithm we calculate the normalized correlation per subband and also determine the maximum correlation that occured. We use the maximum correlation for further analysis.
The correlation measure $sim(X,Y)$ is in $[-1,1]$.

- If $sim(X,Y) = 1$ then the two sequences are exactly the same.
- If $sim(X,Y) = -1$ then the two sequences are exactly opposite, meaning that $X_i = -Y_i \quad \forall i$.
- If $sim(X,Y) = 0$ then there is no correlation between the two sequences. The correlation between two random sequences is close to zero.

For an application scenario you need to determine a threshold correlation value. If the correlation is above the threshold the watermark is said to be detected, otherwise it is not detected. The selected threshold determines the false positive and false negative rate. A false positive happens when the watermark is detected, even though there is no watermark present. A false negative occurs when the watermark is not detected, even though there is a watermark present. Acceptable values for the false positive and negative rate depend on the application scenario. See [57, 17] for a treatment of the subject.
In our results we present the normalized correlation value and do not apply a threshold.
We implemented the Kim, Wang and Xia algorithm using the programming language C. Because the descriptions in some of the papers were not detailed enough for implementation, we filled the gaps as good as possible. For the experiments in chapters 3 and 4 we developed a C++ implementation of the Wang algorithm. The basic wavelet and wavelet packet algorithms were provided by the Ganesh-library. For a complete account of the software used see appendix B.

1.4 Robustness and Security

Cox et al. [17] make the following distinction between robustness and security.
A system is *Robust* if it is resistant to common signal processing operations that are expected to happen. For example this can include channel noise, cropping or lossy compression. Depending on the application scenario the watermark needs to be resistant to different operations.
Security on the other hand is the resistance to malicious attacks on the watermark operation. All possible attacks are usually not known in advance.
There are four categories of attacks:

Unauthorized Detection: For some application scenarios it is detrimental if unauthorized people can detect the watermark. Knowledge of the existence of a watermark, for example, makes sensitivity attacks [58] possible.

Unauthorized Embedding: Unauthorized users should not be able to embed a watermark into images. In the copy attack [52] the watermark from one image is estimated and then embedded into another image. If the watermarking system is not secure against unauthorized embedding it might now appear that the second image contains a legitimate watermark.

Unauthorized Removal: This attack tries to remove the watermark from the image and make it undetectable.

System Attacks: Are attacks on the system as a whole, not targeted at the image as such. For example the inversion attack [19, 20, 21] tries to create a deadlock between the real copyright holder and the attacker. Neither one is able to prove rightful ownership.

There are other possible classifications possible and different authors use different terms. An other possible attack classification is the following [52]:

Removal Attacks: Signal processing operations like denoising, remodulation, lossy compression and quantization. The watermark information is removed from the work and can not be detected.

Geometrical Attacks: Imperceptible geometric operations that make the watermark undetectable. The watermark is still in the image data, but the current detector is unable to detect it. Maybe more advanced detectors can be used, that estimate the geometric modifications and can then detect the watermark. Examples are global or local warps and transfoms and jittering.

Cryptographic Attacks: Use similar methods used for cryptanalysis and apply them to the field of watermarking. Examples are brute force key search, the collusion and the averaging attacks.

Protocol Attacks: Compromises to the system as a whole. Examples are watermark inversion or the copy attack.

A system needs to be robust in order to be secure, because if the watermark can be removed by simple modifications the malicious attacker simply uses them. But a robust system does not need to be secure, because the application scenario might not include malicious attacks.
One important principle is *Kerckhoff's Assumption* [47], which is very important for cryptography. It should be assumed that the method of encryption is known to an attacker. The security should only lie in the choice of the key that was used. The advantage of a publicly known encryption system is that other researchers can investigate the system and find any flaws that might exist. If the system is kept secret there is always the danger that some malicious attacker might find a flaw and exploit it in an already deployed system.
In cryptography there is the discipline of *Cryptanalysis* that analyzes cryptographic systems, tries to find flaws in them and tries to break them. The cycle of developing a new crypto-system,

breaking the system, developing an advanced system that is resistant to previous attacks and then finding new weaknesses, led to many highly developed and very secure cryptographic systems. This would not have been possible, if the details of the systems would have been secret.

The same process happened with watermarking and data hiding in general. One term that is used in this area is *Steganalysis*. The first attacks were simple signal manipulations like lossy compression, low-pass filtering or adding white gaussian noise. For a survey on watermarking applications and possible attacks see [64]. Introductions to steganalysis, attacks and countermeasures can be found in [44, 91, 43]. An overview of different attacks and benchmarking utilities is presented in [92].

Benchmarking applications were developed to automatically assess the robustness and security of watermarking algorithms. The benchmarking program gets a watermarked Work as input and tries different modifications to remove the watermark. One popular tool is Fabien Petitcolas' Stirmark [68, 66, 67] (`http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/index.html`). Another tool to benchmark watermarking technology is Checkmark (`http://watermarking.unige.ch/Checkmark/`). Most early watermarking systems can be defeated by applying Stirmark or Checkmark.

Other discussions on possible watermark attacks, robustness, security and countermeasures can, for example, be found in [3, 15, 40, 84, 27, 26, 29].

1.4.1 Applied Assessments

In the following chapters we will at least perform two assessments for every proposed system: we apply lossy compression to test basic robustness and determine the keyspace to protect against unauthorized detection. In chapter 4 we test the resistance of the proposed systems against removal attacks.

For chapters 2 and 3 we have different parameters that influence the watermark embedding. In chapter 2 we use different filter parametrization values to embed the watermark in a parametrized wavelet domain. In chapter 3 we use different wavelet packet decompositions for the same purpose. In the following we simply speak of the key and either mean the filter parametrization or the wavelet packet decomposition.

Quality Assessment – Robustness

We study the robustness against lossy compression with both JPEG and JPEG2000. For JPEG compression we use quality factors of $95, 90, 80, \ldots, 20, 10, 5$. And for JPEG2000 compression we use compression factors between 0.250 and 0.001.

We only modify the compression rate and keep all other possible parameters constant. Fotopoulos et al. [32] note that the JPEG2000 compression parameter is not the only influence that can be taken into account for assessing the robustness of watermarking systems. Other possible parameters include the tile size, regions of interest and wavelet filter kernels.

We embed the watermark with a specific key into the original image and repeatedly modify the embedding strength parameter until the difference between the original image and the watermarked image is below 40dB, as measured by the Peak Signal to Noise Ratio (PSNR).

Then we take the watermarked image and compress it with one compression method at a specified compression rate. First we calculate the PSNR of the compressed image to see how many artifacts were introduced. We also try to detect the watermark in the compressed image. Because the compression removes part of the watermark information, the correlation will be lower for higher compression rates.

Appendix A shows the original, unwatermarked images Lena and Barbara and also shows the PSNR behavior under compression. We also included a few images after compression to make the occuring artifacts more graspable.

Security Assessment – Unauthorized Detection

To analyze the available keyspace we embed the watermark with one specific key and then try to detect the watermark with a large set of incorrect keys.
The ideal system would have a normalized correlation of zero everywhere except for the key location where the correlation would be 1.0. Because of rounding errors and random similarities that perfect system is not possible. In reality you have to decide on a threshold level and say that the watermark is embedded, if the correlation is above the threshold.
If possible we look at different resolutions to determine the smallest difference between key values that still create a clear difference between the key location and wrong key values.

Removal Attack

In chapter 4 we analyze the security of the two systems we propose in chapters 2 and 3 against malicious modification of the wavelet coefficients.
We did not find any previous work on malicious attacks directly conducted against wavelet coefficients. For details on the attacks we applied see chapter 4.

Chapter 2

Filter Parametrization

In this chapter we explore the use of parametrized wavelet filters as a way to increase the security of watermarking systems.

We give an overview of previous work in section 2.1 and in section 2.2 we give an introduction to wavelet filter parametrization and explain how to use it to increase security.

Section 2.3 explores the properties of a system with two parameters and then in section 2.4 we look at systems that use six parameters that are distributed over the different decomposition levels. Section 2.5 discusses the properties of systems with 1, 2, 3, 5, 6 and 9 filter parameters.

Finally, in section 2.6 on page 55 we present a combined system with a total of 20 parameters. 5 parameters are used for each filter parametrization and we use 4 different filters for the decomposition levels.

In section 2.7 we present some example images and in section 2.8 we draw some final conclusions about wavelet filter parametrization.

2.1 Previous Work

In previous work the following techniques to enhance the security of watermarks have been proposed.

Pseudo-random skipping of coefficients has been proposed by Wang [96] and Kundur [51].

Fridrich [35] introduced the concept of key-dependent basis functions in order to protect a watermark from hostile attacks. By embedding the watermark information in a secret transform domain, Fridrich's algorithm can better withstand attacks such as those described by Kalker [45] employing a public watermark detector device. However, Fridrich's approach suffers from the computational complexity and the storage requirements for generating numerous orthogonal patterns of the size of the host image. In a later paper Fridrich reduced the computational complexity of the system [34].

Wang [97] uses randomly generated orthonormal filter banks as one part of the private key of his watermarking system.

Pollen [75] describes a method to parametrize wavelets using two free parameters and in [87] a lifting based integer wavelet transform with one free parameter is presented. Hartenstein [39] developed a parametrization of biorthogonal filters.

In the following we will describe and use the filter parametrization developed by Schneid [80], based on work by Zou [101].

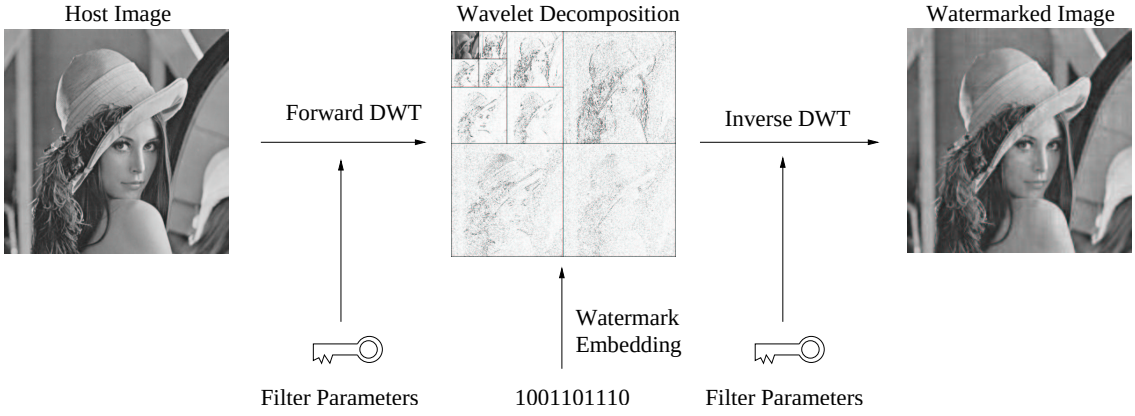


Figure 2.1: Overview of watermark embedding procedure using parametrized wavelet filters

2.2 Filter Parametrization

In order to construct compactly supported orthonormal wavelets, solutions for the dilation equation

$$\phi(t) = \sum_{k\in} c_k \phi(2t-k),$$

with $c_k \in$, have to be derived, satisfying two conditions on the coefficients c_k [22]. Schneid [80] describes a parametrization for suitable coefficients c_k based on the work of Zou [101] to facilitate construction of such wavelets. Given N parameter values $-\pi \leq \alpha_i < \pi$, $0 \leq i < N$, the recursion

$$\begin{aligned} c_0^0 &= \tfrac{1}{\sqrt{2}} \quad \text{and} \quad c_1^0 = \tfrac{1}{\sqrt{2}} \\ c_k^n &= \tfrac{1}{2}((c_{k-2}^{n-1} + c_k^{n-1}) \cdot (1+\cos\alpha_{n-1}) + \\ &\quad (c_{2(n+1)-k-1}^{n-1} - c_{2(n+1)-k-3}^{n-1})(-1)^k \sin\alpha_{n-1}) \end{aligned}$$

can be used to determine the filter coefficients c_k^N, $0 \leq k < 2N+2$. We set $c_k = 0$ for $k < 0$ and $k \geq 2N+2$.

We propose to decompose the host image using wavelet filters constructed with the above parametrization. The parameter values α_i used for construction and the resulting wavelet filter coefficients are kept secret. Hence, the watermark information can be embedded in a secret multi-resolution transform domain, making it difficult to mount a hostile attack that seeks to destroy or remove watermark information at specific locations.

Our concept is illustrated in figure 2.1. Figure 2.2 compares a standard Daubechies 6 filter with a parametric filter with $N = 2$ that was generated using $\alpha_0 = -0.4815$ and $\alpha_1 = 2.6585$, resulting in a 6-tap filter.

2.3 Two Parameters

For this first analysis we use two parameters to create the wavelet filters. The first investigation will look at the security of the system. Then we examine the correlation and PSNR values under compression with JPEG and JPEG2000 at different compression rates. Finally, we look at what effect the parameter selection has on correlation and image quality.

2.3.1 Security Assessment

The goal of this assessment is to analyze the effectiveness of the filter parameters to protect against unauthorized detection and to determine the smallest difference in parameter values that still allows a clear distinction between correct and incorrect parameter values.

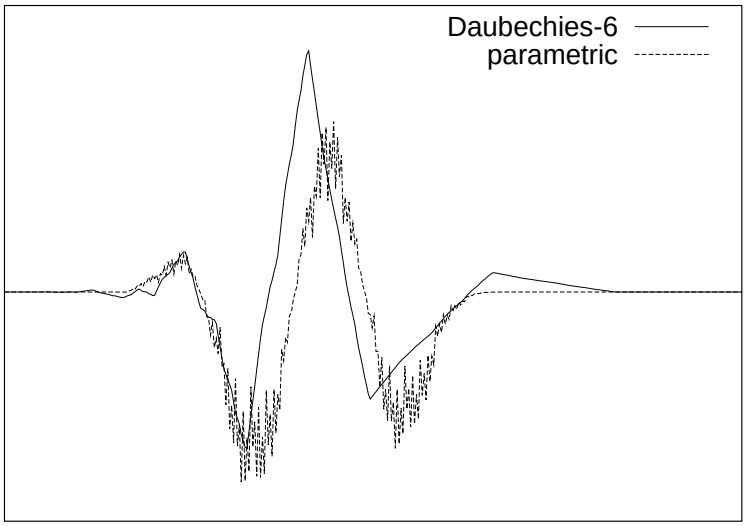


Figure 2.2: Comparison of standard Daubechies 6 and a parametrized wavelet filter using $\alpha_0 = -0.4815$ and $\alpha_1 = 2.6585$

The filter parametrization is the value that we use as key for the embedding process. Other possible keys, like the seed for the spread spectrum sequence or a seed for significant coefficient skipping, are kept constant for the following experiments.

We embed one specific watermark with one filter parametrization and then try to detect the watermark with other filter parameters. Our goal in introducing the parametrized wavelet filters is to increase the security of the watermarking systems. Therefore the correlation between the embedded watermark and the watermark extracted with the wrong filter parametrization should be very low.

The ideal system would have a normalized correlation of zero everywhere except for the key location where the correlation would be 1.0. Because of rounding errors and random similarities that perfect system is not possible. In reality you have to decide on a threshold level and say that the watermark is embedded, if the correlation is above the threshold. The selection of the threshold depends on the watermarking system, expected attacks and the desired false-positive and false-negative rates.

If there is high correlation even when the filter parameters are set to wrong values close to the correct values, then the security improvement is not as high as expected, because even if you are only in the proximity of the correct key you already know that the watermark is embedded.

We look for the smallest difference between parameter values that is necessary to have a clear separation in the correlation values of the correct filter parameters and incorrect ones. If we choose this difference too small, then we might get high correlation even though the wrong filter parameters are used. If we choose it too large, then the number of possible filter parameters is small and therefore limits the key space of the system.

In all the following experiments we embed the watermark with the parameters $\alpha_0 = 0.150$ and $\alpha_1 = 0.650$ and an embedding strength that results in 40dB PSNR into the well known “Lena” image. Then we use different parameter ranges to vary the filter parameters and measure the resulting correlation between the embedded watermark and the watermark that is extracted with this filter.

Kim Algorithm

The results of the first experiment with the Kim algorithm are shown in figure 2.3. Here we select $\alpha_0, \alpha_1 \in \{-3.14, -3.13, \ldots, 3.13, 3.14\}$ with $\Delta = 0.01$. This results in 395641 samples, i.e. potentially different filter parameters. There is only one peak with a correlation of 0.903694 at the location of the embedding parameters. For the other samples the correlation is always smaller than 0.25, most of the time being below 0.10.

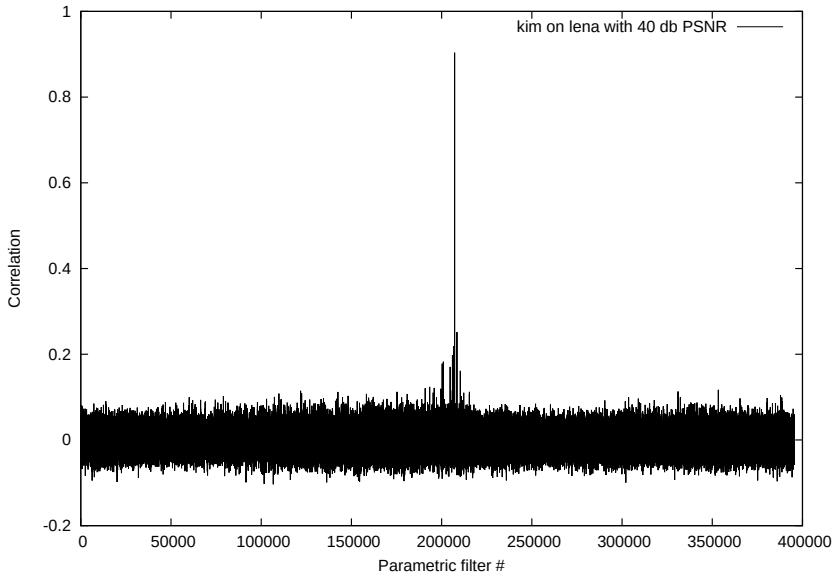


Figure 2.3: Security evaluation of Kim algorithm. There is only one peak at the exact parameter values used for embedding. $\alpha_0, \alpha_1 \in \{-3.14, -3.13, \ldots, 3.13, 3.14\}$ with $\Delta = 0.01$

This result is very promising. We have high correlation only with the parameters used for embedding the watermark and the correlation is low for all other filter parameters. Because the only increase in correlation is around the embedding parameters we will confine our further examinations to this area.

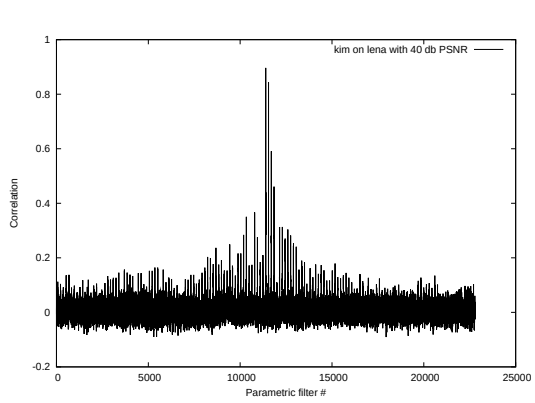
Figure 2.4(a) shows the correlation when $\alpha_0 \in \{0.000, \ldots, 0.300\}$ and $\alpha_1 \in \{0.500, \ldots, 0.800\}$ with $\Delta = 0.002$ resulting in 22801 measurements. Applied to the whole parameter space using $\Delta = 0.002$ would result in 9865881 possible filter keys. There are two peaks in the correlation — at the embedding position and in very close proximity.

To better visualize this result we use the parameter values α_0 and α_1 as X- and Y-axis and map the correlation to a shade of gray. A correlation of zero corresponds to black and a correlation of one to white. In figure 2.4(b) you can see one white dot at the center of the image.

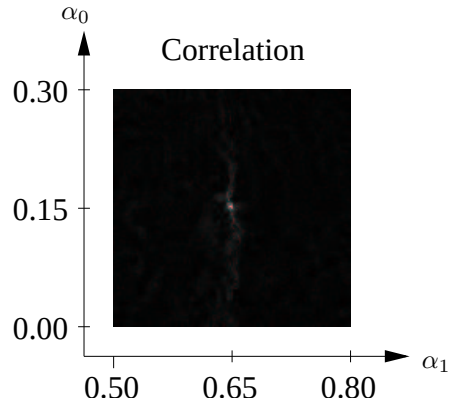
This picture helps us see that there is a higher correlation around the Y-axis for $\alpha_1 = 0.650$. To further analyse this we look at the sensitivity of one parameter if the other parameter is already set to the right value.

Diagram 2.4(c) shows the correlation when $\alpha_1 = 0.650$ and $\alpha_0 \in \{0.000, \ldots, 0.300\}$; diagram 2.4(d) sets α_0 to 0.150 and varies $\alpha_1 \in \{0.500, \ldots, 0.800\}$; both diagrams use $\Delta = 0.0005$ which results in 601 measurements each, which would be equivalent to 157778721 different filter keys.

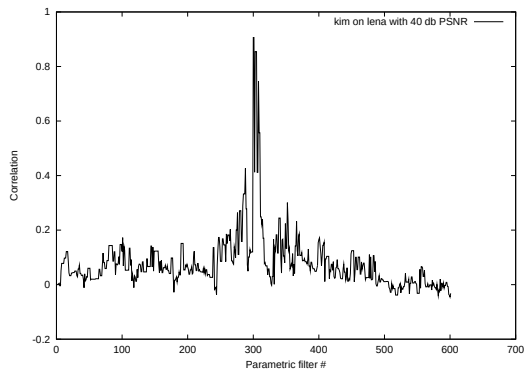
At this fine resolution we see that there are several peaks close to the embedding value. Diagram 2.4(c) also has a broader region with high correlation than diagram 2.4(d). This results in the vertical "stripe" in diagram 2.4(b). With the step size Δ at around 0.0005 we are approaching the finest meaningful resolution, further refinement would lead to a large range of high correlation around the true embedding key.



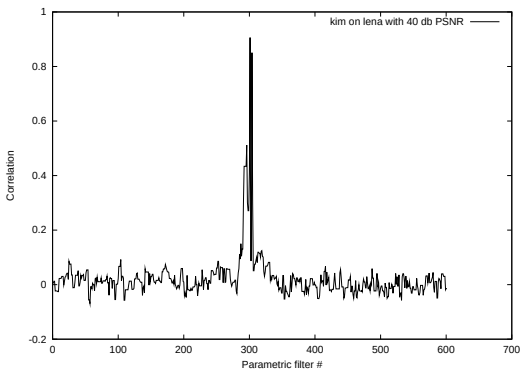
(a) Resolution $\Delta = 0.002$



(b) Resolution $\Delta = 0.002$

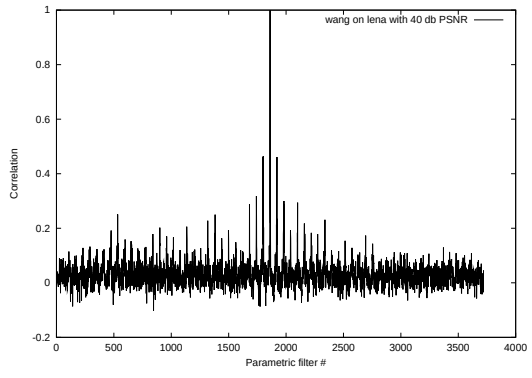


(c) Resolution $\Delta = 0.0005$

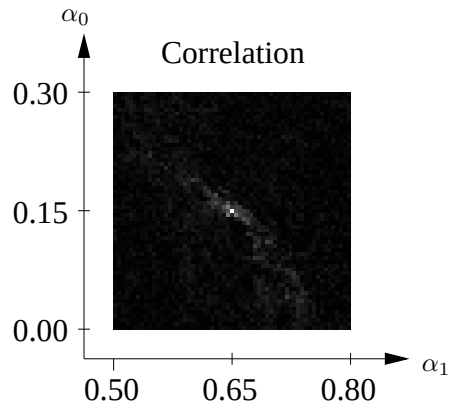


(d) Resolution $\Delta = 0.0005$

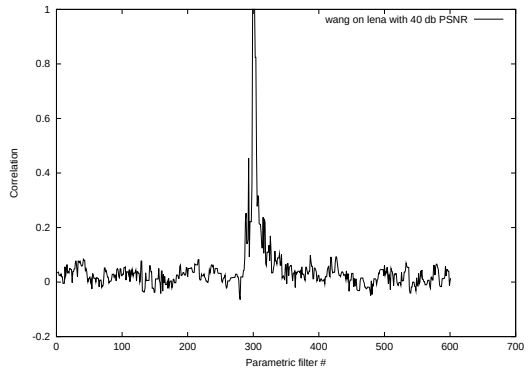
Figure 2.4: Parameter security evaluation for Kim algorithm. The watermark was embedded with a PSNR of 40db at $\alpha_0 = 0.150$ and $\alpha_1 = 0.650$



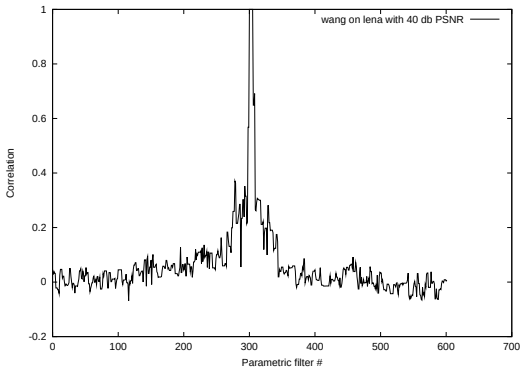
(a) Resolution $\Delta = 0.005$



(b) Resolution $\Delta = 0.005$



(c) Resolution $\Delta = 0.0005$

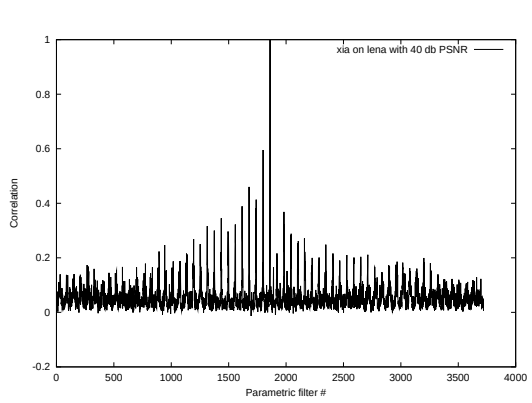


(d) Resolution $\Delta = 0.0005$

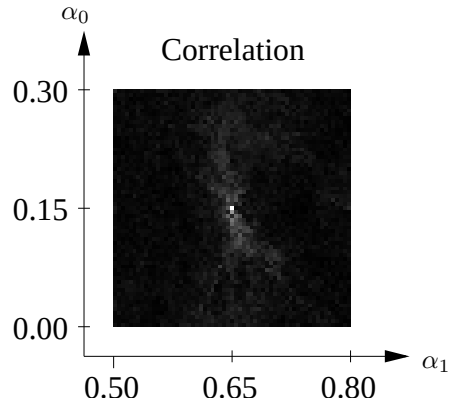
Figure 2.5: Parameter security evaluation for Wang algorithm. The watermark was embedded with a PSNR of 40db at $\alpha_0 = 0.150$ and $\alpha_1 = 0.650$

Wang and Xia Algorithms

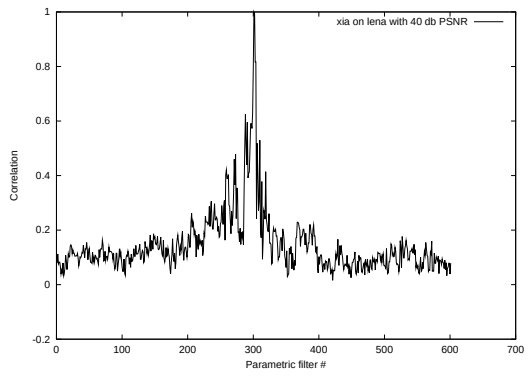
Figures 2.5 and 2.6 show the corresponding results for the Wang and Xia algorithms.
Diagrams (a) and (b) show the correlation when $\alpha_0 \in \{0.000, \ldots, 0.300\}$ and $\alpha_1 \in \{0.500, \ldots, 0.800\}$ with $\Delta = 0.005$ resulting in 3721 measurements. At this resolution we have 1580049 possible filter keys.
For both algorithms the correlation has only one clear maximum at the embedding values.
The diagrams (c) show the correlation when $\alpha_1 = 0.650$ and $\alpha_0 \in \{0.000, \ldots, 0.300\}$; diagrams (d) set α_0 to 0.150 and vary $\alpha_1 \in \{0.500, \ldots, 0.800\}$; both diagrams use $\Delta = 0.0005$ which results in 601 measurements each, which would be equivalent to 157778721 different filter keys.
A good choice for Δ therefore is in the range of 0.005 and 0.01 which results in a large number of possible keys and either one clear peak or a very small area of high correlation.



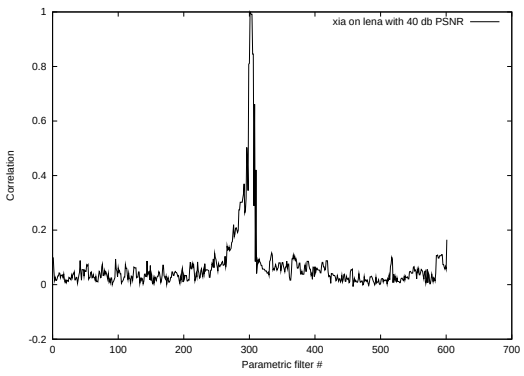
(a) Resolution $\Delta = 0.005$



(b) Resolution $\Delta = 0.005$



(c) Resolution $\Delta = 0.0005$



(d) Resolution $\Delta = 0.0005$

Figure 2.6: Parameter security evaluation for Xia algorithm. The watermark was embedded with a PSNR of 40db at $\alpha_0 = 0.150$ and $\alpha_1 = 0.650$

2.3.2 Quality Assessment

Next we explore the difference between the parametric filters and the Daubechies 6 and the Biorthogonal 7/9 filters with regard to correlation and PSNR under JPEG and JPEG2000 compression.

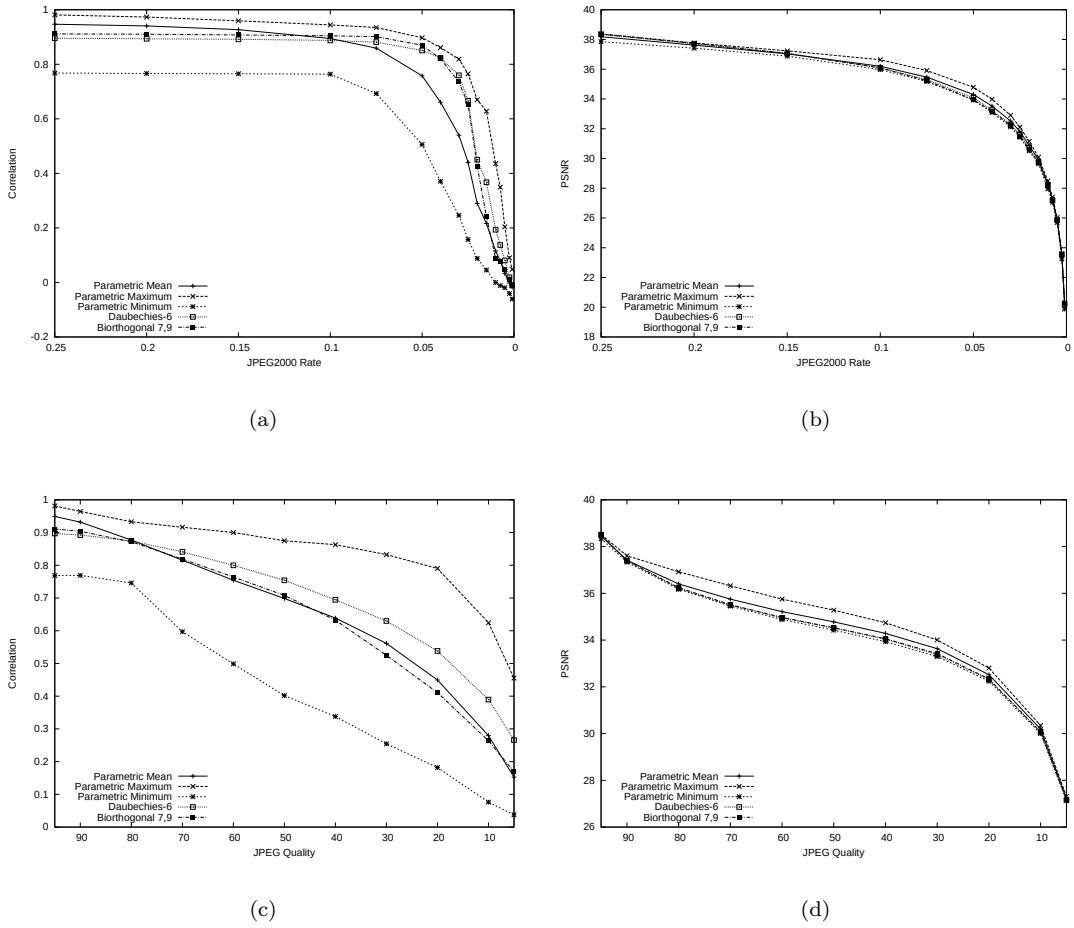


Figure 2.7: Correlation and PSNR of Kim watermarks under JPEG and JPEG2000 compression

We watermark the "Lena" image with each parametrized filter with $\alpha_0, \alpha_1 \in \{-3.00, \ldots, 3.00\}$ and $\Delta = 0.20$. This generates 961 different wavelet filters. The watermark is embedded with an embedding strength that results in a 40dB PSNR. Each picture is JPEG and JPEG2000 compressed with different quality levels and then the correlation and PSNR is measured.
Figure 2.7 shows the results for the Kim algorithm. Diagram (a) shows the correlation after JPEG2000 compression of the watermarked image, diagram (b) the PSNR after JPEG2000 compression. Diagram (c) shows the correlation, diagram (d) the PSNR after JPEG compression.
The diagrams contain the average of all measured parametrized filters and the minimum and maximum for each compression rate. The average parametrized filters PSNR values are slightly above the standard filters and behave very similarly to them. The influence of parametrization on image quality is therefore minimal. The correlation values vary in a wider range. The average parametric correlation is close to that of the standard filters.
Diagram 2.8 contains the results for the Wang algorithm, diagram 2.9 the results for the Xia algorithm. The behavior of these algorithms under compression is very similar. For both the average correlation is very close to the two standard filters. The difference between the minimum

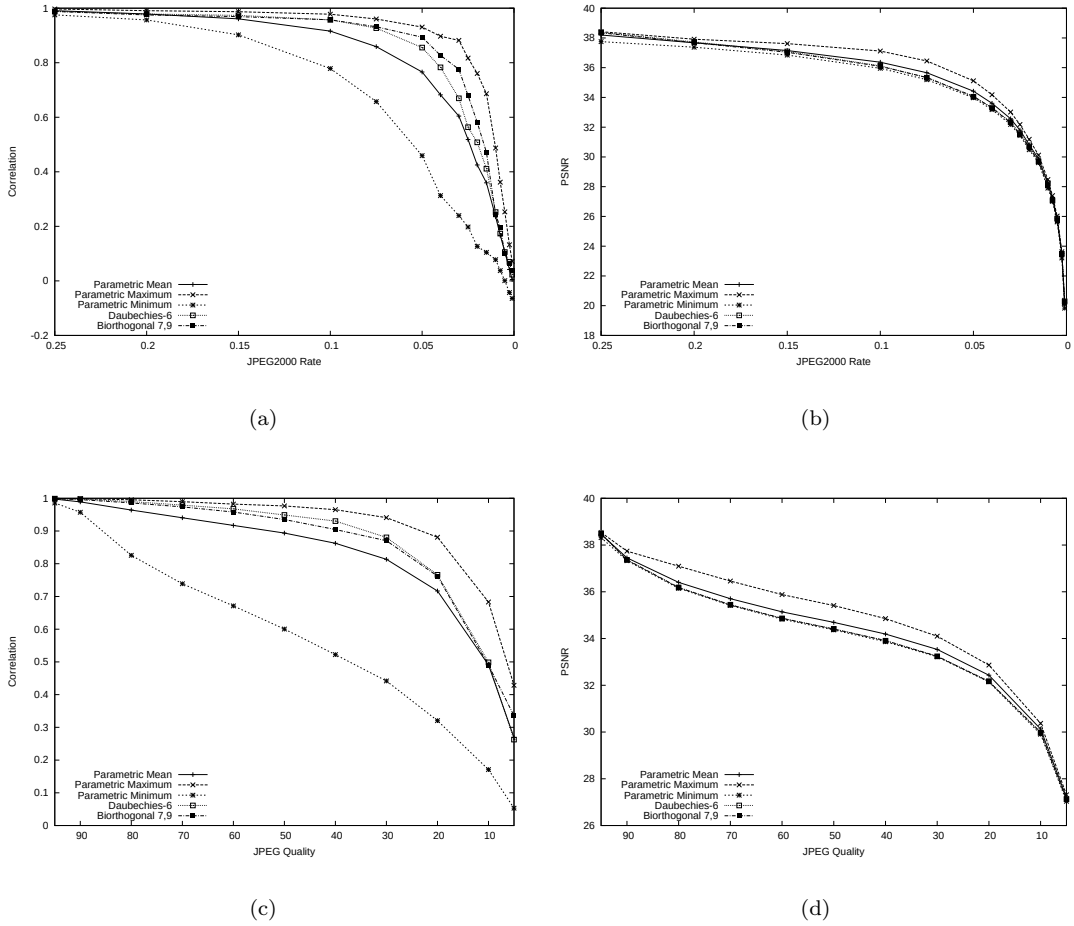


Figure 2.8: Correlation and PSNR of Wang watermarks under JPEG and JPEG2000 compression

and the maximum of the measured values is larger for the Wang algorithm. This bandwidth of possible behavior under compression is smaller for the Xia algorithm.

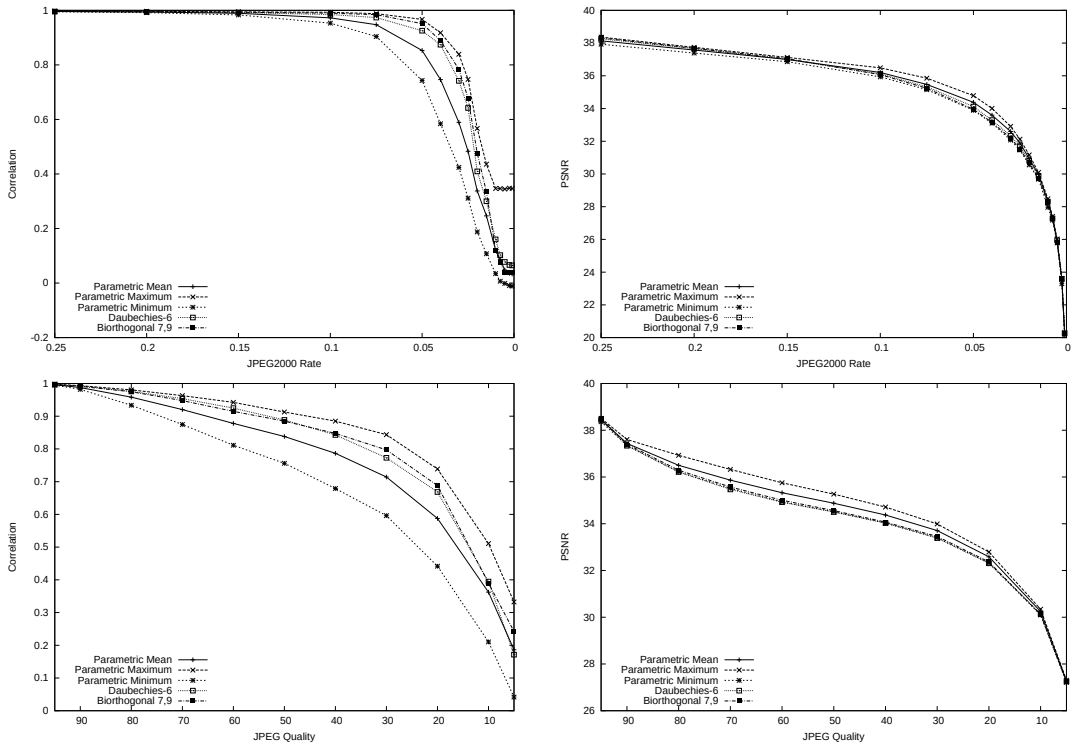
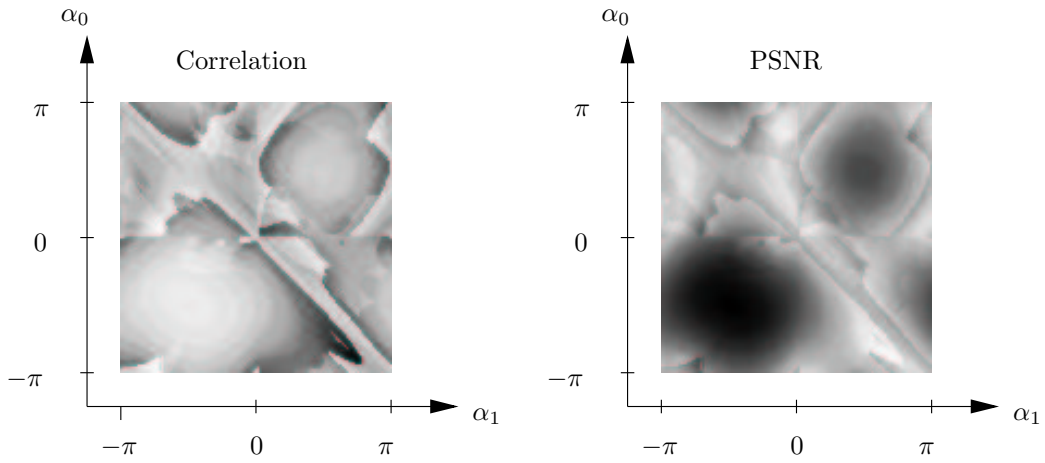


Figure 2.9: Correlation and PSNR of Xia watermarks under JPEG and JPEG2000 compression

2.3.3 Parameter Quality Assessment

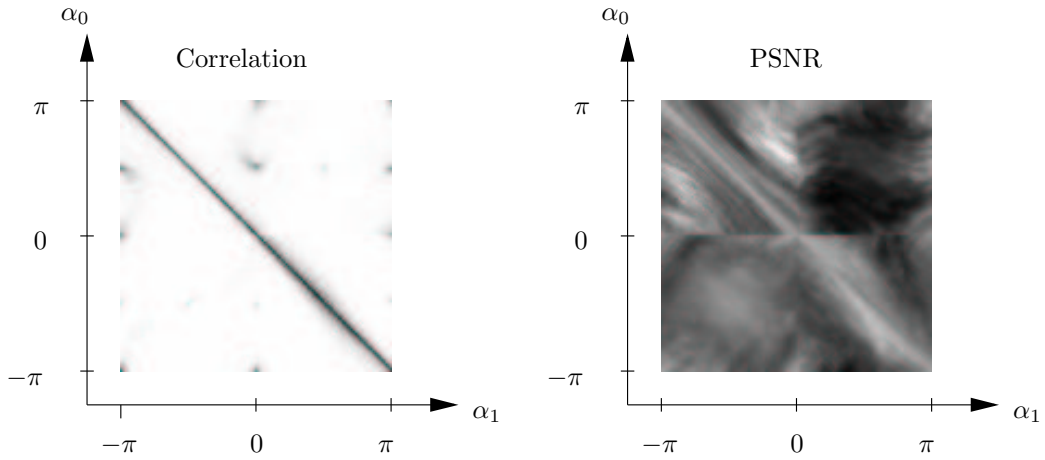
For the last test we embed a watermark with 40dB PSNR with the wavelet filters that are generated by α_0 and α_1 both equal to zero. This parameter selection results in the well-known Haar Wavelet. We save the embedding strength that was used to achieve this PSNR and then use it for the other parametrized filters with $\alpha_0, \alpha_1 \in \{-3.10, \ldots, 3.10\}$ and $\Delta = 0.05$. For each filter we measure the correlation and PSNR that results from using this fixed embedding strength.
Figures 2.10, 2.11 and 2.12 show the correlation and PSNR over the parameter values as two-dimensional plots. For these plots we scale the shades of gray in such a way that black means the minimum and white means the maximum of all measured values. We also show the maximum, average and minimum values measured.



Max. Correlation: 0.988097 Avg. Correlation: 0.950132 Min. Correlation: 0.854893
Max. PSNR: 42.309034 Avg. PSNR: 38.741370 Min. PSNR: 34.180642

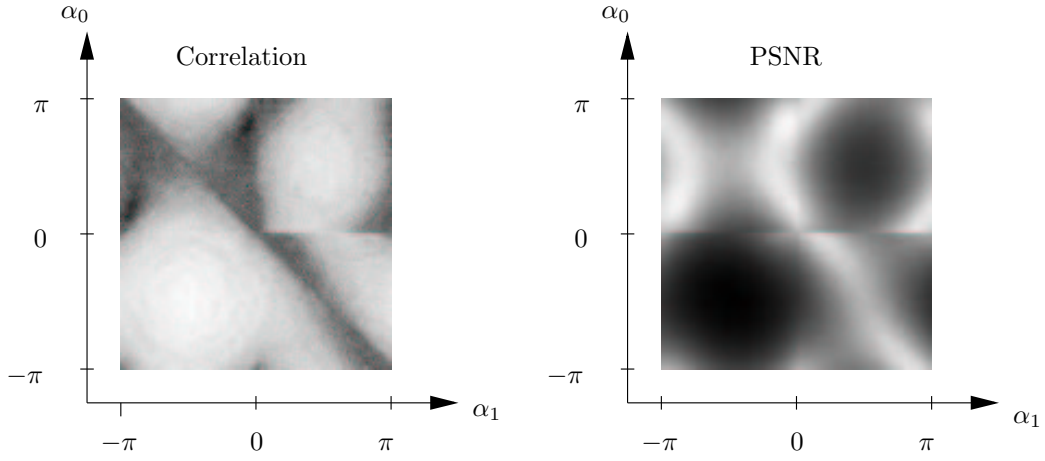
Figure 2.10: Kim watermark embedded with strength 0.37501

The correlation values are very good for all possible values and the PSNR is in a reasonable range. What should be noted is that the minima and maxima of correlation and PSNR are complementary. This means that the correlation is lower in regions with a high PSNR and the other way around. Therefore a desired PSNR should always be possible by adjusting the embedding strength to the selected filter values.



Max. Correlation: 0.999966 Avg. Correlation: 0.999554 Min. Correlation: 0.989557
Max. PSNR: 41.050860 Avg. PSNR: 39.356588 Min. PSNR: 38.252465

Figure 2.11: Wang watermark embedded with strength 0.16568



Max. Correlation: 0.999675 Avg. Correlation: 0.999015 Min. Correlation: 0.997325
Max. PSNR: 43.275802 Avg. PSNR: 37.636727 Min. PSNR: 32.510788

Figure 2.12: Xia watermark embedded with strength 0.10710

2.4 Different Combinations of Six Parameters

In the previous section we analyzed the security and quality aspects of using two filter parameters as watermarking key. We have seen that with two parameters we have a keyspace from a few hundred-thousands to a few millions, depending on what resolution is selected.
We have two options if we are interested in a larger possible keyspace. We could use more than two parameters to create a filter or use different filters for the different wavelet decomposition levels, which is also called Non-Stationary Multi-Resolution Analysis (NSMRA). Of course we can also combine those two possibilities.
In [89] non-stationary MRA is used for improving image compression. Here we use non-stationary MRA as a method to increase security. To get a larger parameter space we use different parametrized filters for the decomposition levels. We hope that the number of parameters for the different levels add up and produce a system with a large number of parameters.
In this section we are going to look at different possibilities to distribute six parameters over the decomposition levels. We analyze different parameter numbers in detail in section 2.5 and look at a combined system in section 2.6.
The parameter combinations we look at now are:

Parameter	1	2	3	4	5	6
mix-1-3-2	A	B	B	B	C	C
mix-2-1-1-1-1	A	A	B	C	D	E
mix-2-2-1-1	A	A	B	B	C	D
mix-3-1-1-1	A	A	A	B	C	D
mix-3-2-1	A	A	A	B	B	C
mix-4-1-1	A	A	A	A	B	C
mix-4-2	A	A	A	A	B	B
one-filter	A	A	A	A	A	A
two-filters	A	A	A	B	B	B
three-filters	A	A	B	B	C	C
six-filters	A	B	C	D	E	F

Parameters that have the same letter assigned are used as parameters for the same decomposition level. The filter parameters that are used for the last given decomposition level are also used for all deeper decompositions, if any.
For example the "three-filters" combination is "A A B B C C". This means that the first two parameters are used to parametrize the filter for the first decomposition level. The third and fourth parameters are used for the second level and the last two parameters are used to decompose the third and all deeper decomposition levels.
The Xia algorithm is designed to extract the watermark after as few decompositions as possible. Therefore the embedding with non-stationary filters does not improve the security of the system. The experiments for the Xia algorithm behave very similarly to the experiments for the Kim algorithm and therefore are not shown seperatly.

2.4.1 Security Assessment

To test the security of the system we embed a fixed watermark with one key parametrization and then try to detect the watermark with other parametrizations. We embed the watermark into the well-known "Lena" image with an embedding strength that results in 40dB PSNR.
We use the parameters "-1.5 2.5 -1.0 1.5 0.5 -2.5" to embed the watermark.
The overview diagrams vary all the parameters with ± 0.1 and $\Delta = 0.05$. This results in 5 possibilities for every parameter and a total of $5^6 = 15625$ analyzed filter parametrizations. Using $\Delta = 0.05$ over the complete parameter space results in more than 2^{40} possible filter parametrizations.

The detail diagrams vary only one parameter with ± 0.2 and $\Delta = 0.002$; all the other parameters are set to the correct key position. This results in $0.4/0.002 + 1 = 201$ analyzed filters. "Detail A" means that only the first of the six parameters is varied, "Detail B" that only the second parameter is varied and so on.
Because of space restrictions we are only going to show the result diagrams for the "one-filter", "six-filters", "mix-3-2-1" and "mix-1-3-2" systems with the Kim and Wang watermarking algorithms. The main problems of the different parameter distributions can be seen from those four systems.

Kim — One Filter with Six Parameters

Here all six parameters are used to generate one filter and that filter is used for all decomposition levels. Diagram 2.13 shows the behavior when all parameters are varied and diagrams 2.14(a)-(f) show the response when just one parameter is modified.
The results of this system are very similar to the two parameter case. There is one clear peak at the embedding position and all other correlation values are below 0.4. All 6 detail diagrams show just a very small area of high correlation.

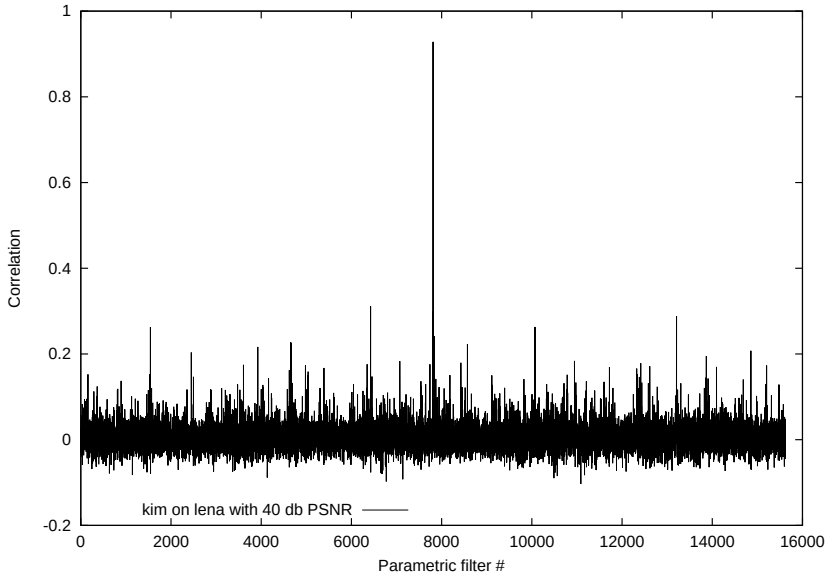
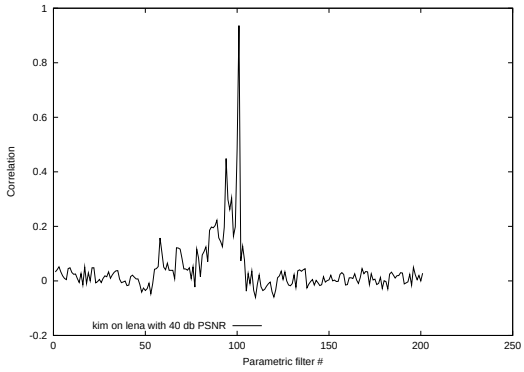
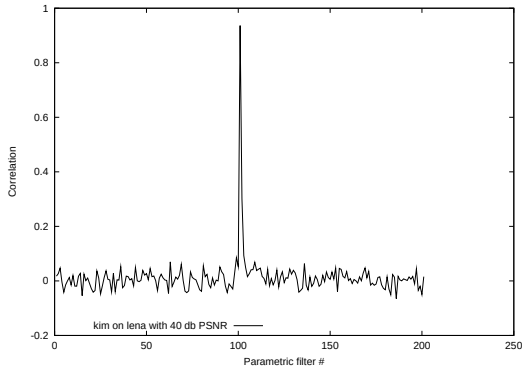


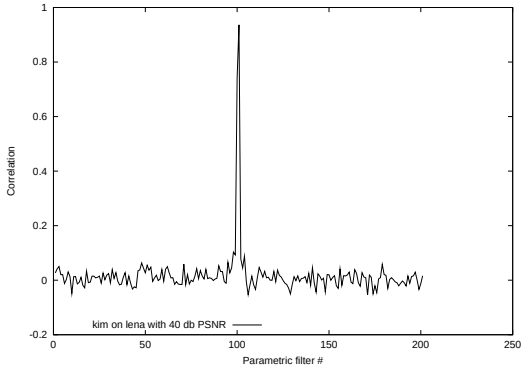
Figure 2.13: Kim — One filter with six parameters — Overview



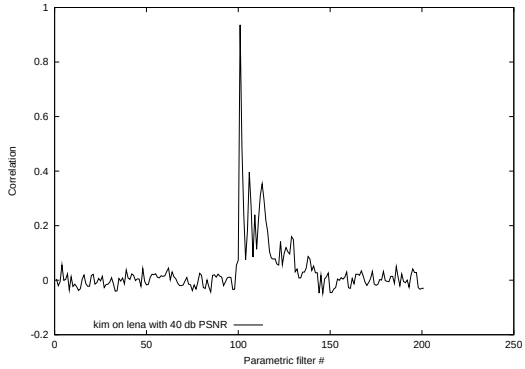
(a) Detail A



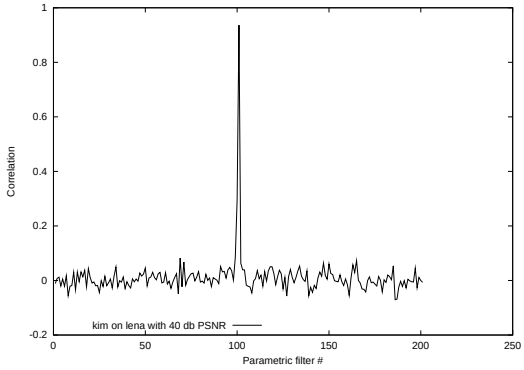
(b) Detail B



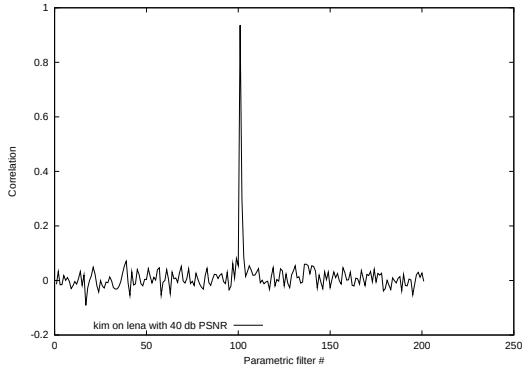
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

Figure 2.14: Kim — One filter with six parameters — Details

Kim — Six Filters each with One Parameter

The other extreme case is that every parameter is used for a different filter, generating six filters for six decomposition levels. Diagram 2.15 gives the overview and diagram 2.16 the single parameter behavior.
We can already see from diagram 2.15 that there is a security problem. There is a range from roughly parametrization 6000 to 9000 that has a correlation of more than 0.70. This means that even if you are only close to the correct embedding parameters you already get a very high correlation value.
By looking at diagram 2.16 we see what the problem is. Diagram (a) shows good behavior and we only have one peak in the correlation. But as soon as the first parameter has the correct value we have an increase in correlation to more than 0.70. Diagram (b) has a correlation of nearly 0.80 over the complete parameter range. Diagrams (c) through (f) look even worse. There is no significant difference in correlation if you vary the parameters for the fifth and sixth decomposition levels.
Clearly the security of this system is not what we expect. From all six parameters only the first one has real significance. The other five parameters do not influence the correlation in the expected way.

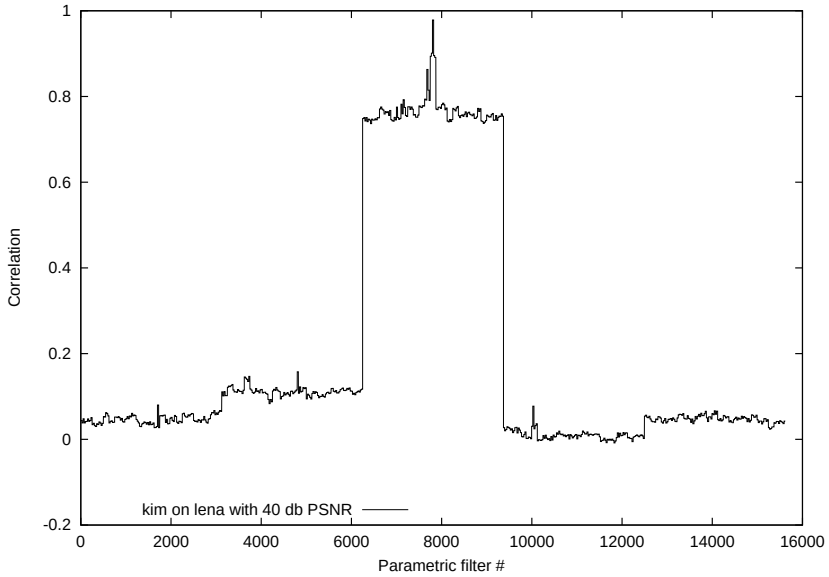
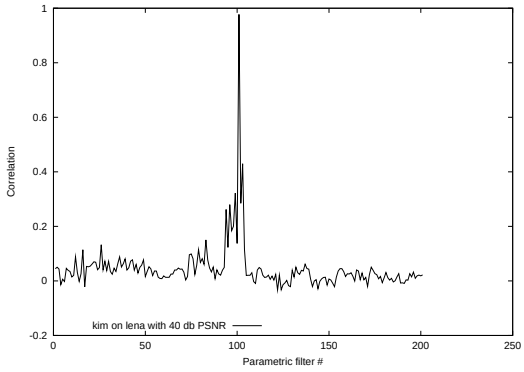
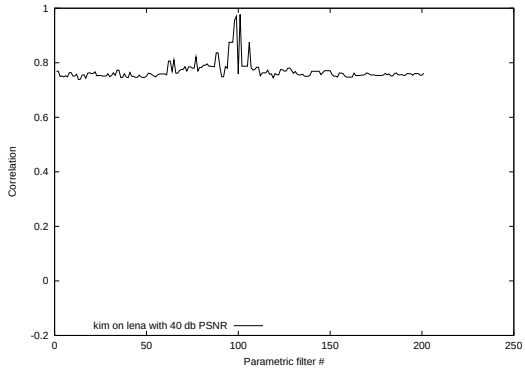


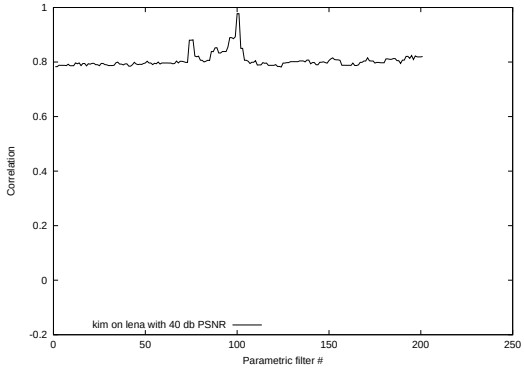
Figure 2.15: Kim — Six filters each with one parameter — Overview



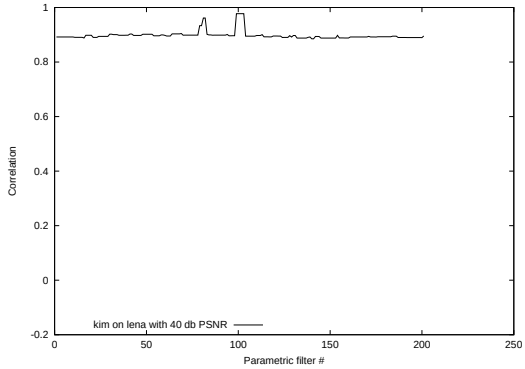
(a) Detail A



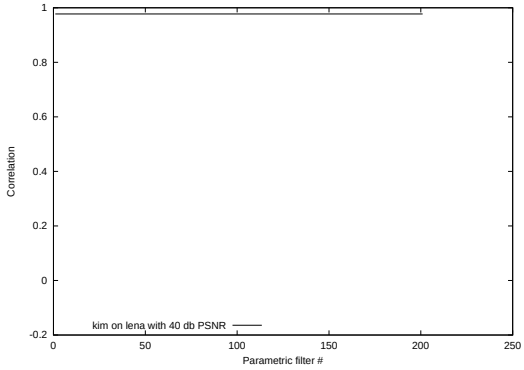
(b) Detail B



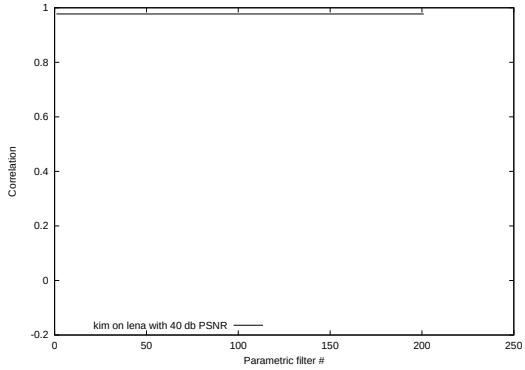
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

Figure 2.16: Kim — Six filters each with one parameter — Details

Kim — Mix 3-2-1

In diagram 2.17 we see a small area of high correlation and one clear peak. But when we look at diagrams 2.18 (a) through (f) we see that as soon as the first three parameters are set to the key values we have a jump in the correlation. The next two parameters are used for the second level and have a correlation of more than 0.60 over the complete range. The last parameter is used for the third and all higher decomposition levels and always has a correlation of around 0.90.
The added security of the last three parameters is therefore minimal. As soon as the first three parameters are correct we have a correlation of more than 0.60.

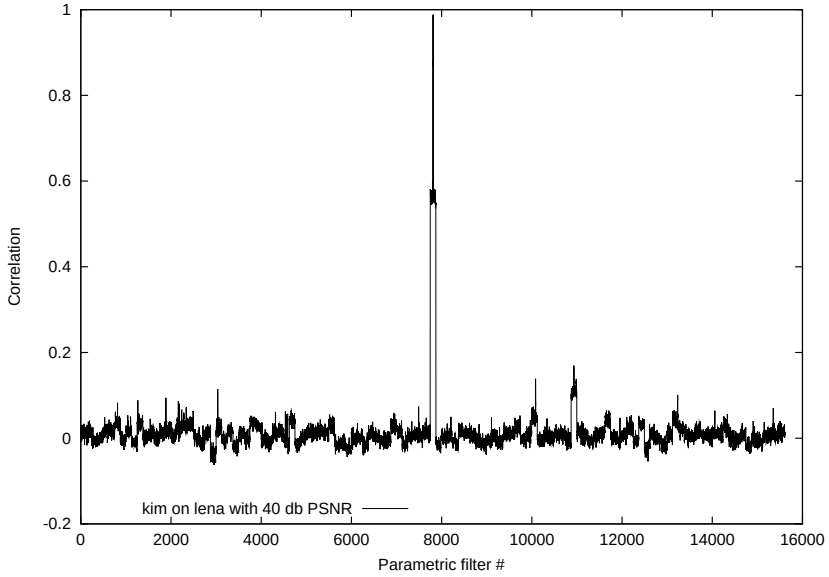
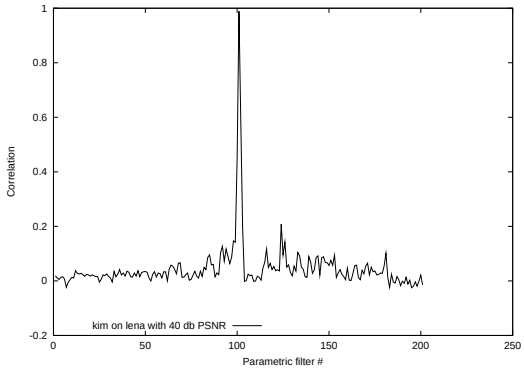
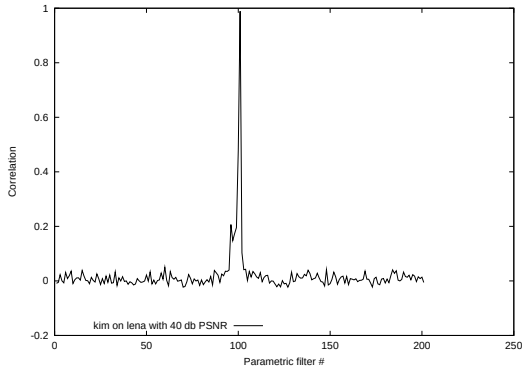


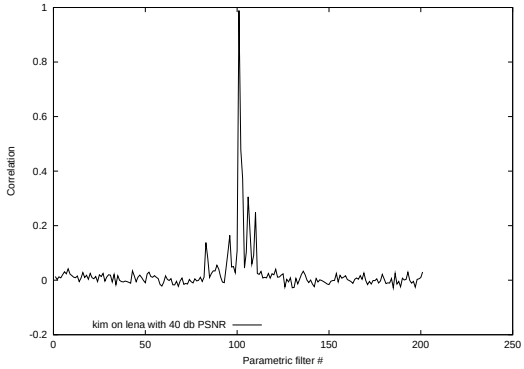
Figure 2.17: Kim — Mix 3-2-1 — Overview



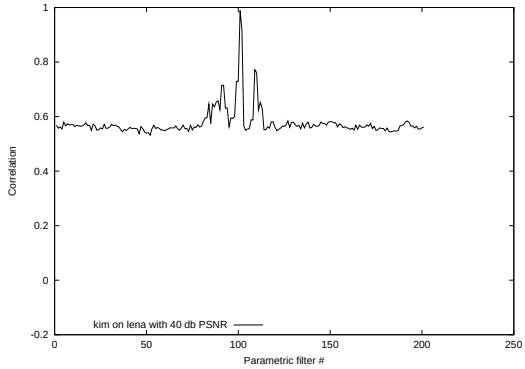
(a) Detail A



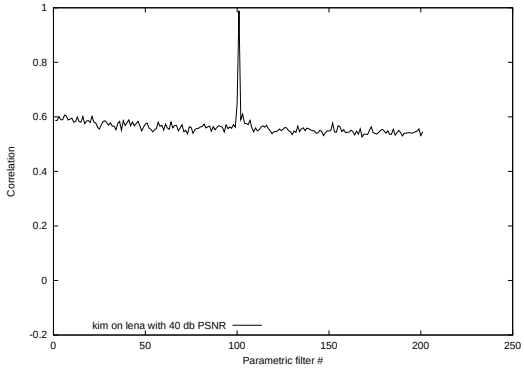
(b) Detail B



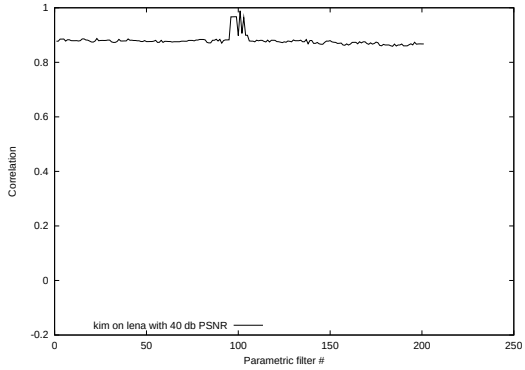
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

Figure 2.18: Kim — Mix 3-2-1 — Details

Kim — Mix 1-3-2

Diagram 2.19 looks very similar to diagram 2.15. There is a large range of around 3000 different parametrizations that result in a correlation of 0.60. In diagram 2.20 we again see that as soon as the first parameter is set to the correct key position we see very little effect from the remaining parameters and have a high correlation for all parametrizations.
From these results we conclude that using the Kim algorithm with non-stationary MRA does not result in the expected security increase. As soon as the filter for the first decomposition is known, there is minimal influence from the other decomposition levels. Therefore there is no real increase in security if more than one filter is used.

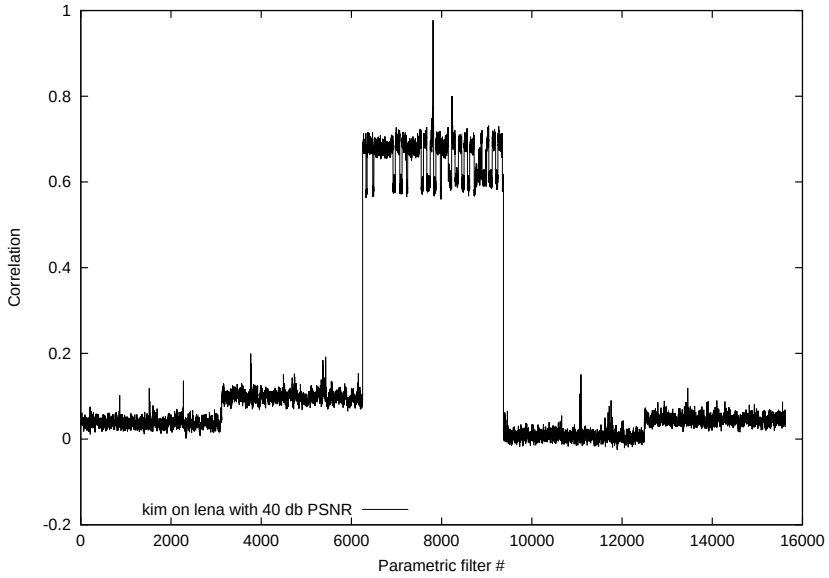
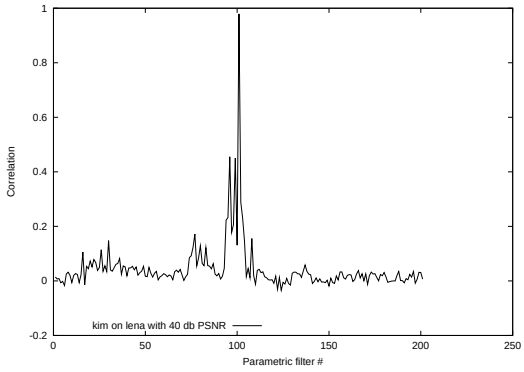
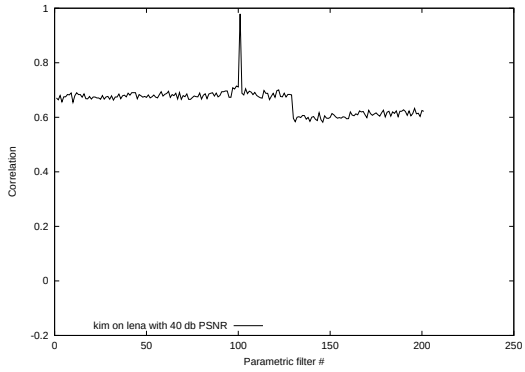


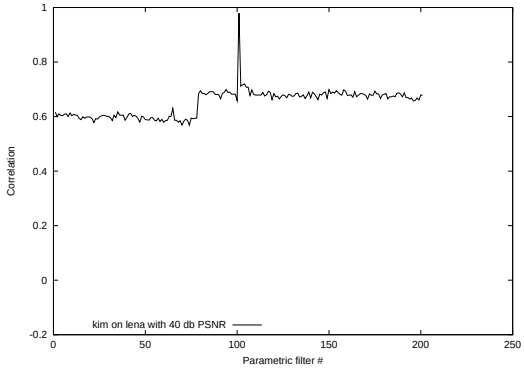
Figure 2.19: Kim — Mix 1-3-2 — Overview



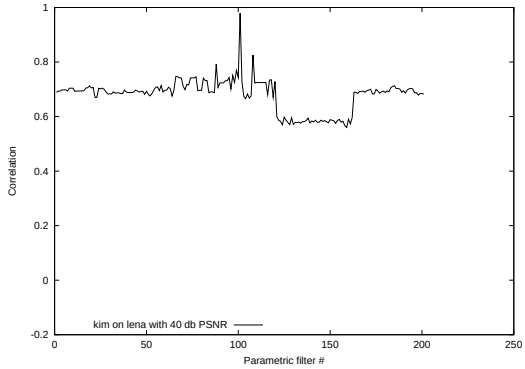
(a) Detail A



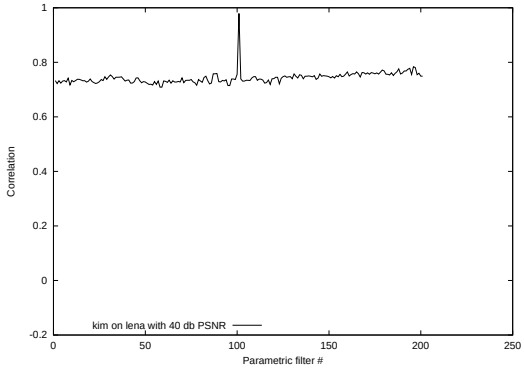
(b) Detail B



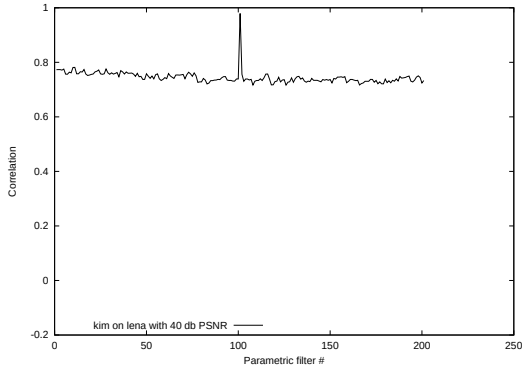
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

Figure 2.20: Kim — Mix 1-3-2 — Details

Wang — One Filter with Six Parameters

Diagrams 2.21 and 2.22 show the results for using all six parameters for one filter and embedding the watermark using the Wang algorithm. As expected there is one clear peak and generally very low correlation everywhere else.
Also the influence from each of the six parameters looks similar and each has a very small area of high correlation.

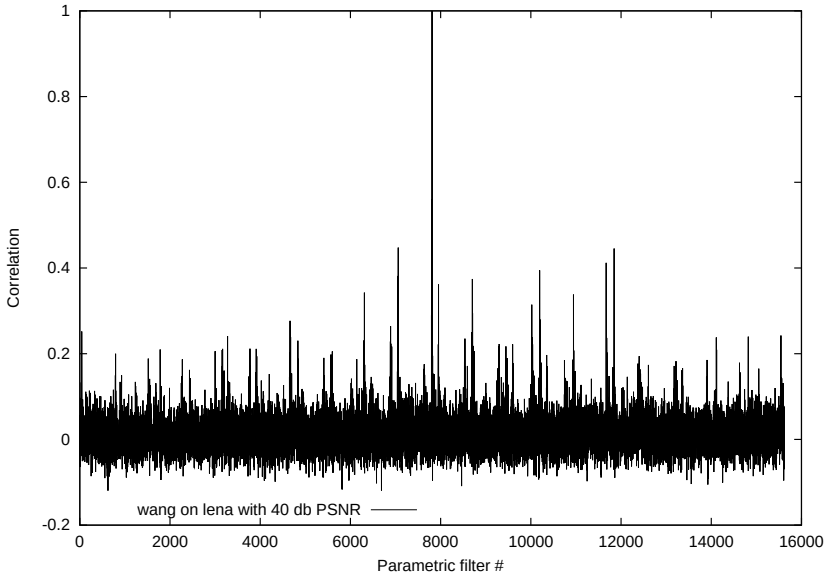
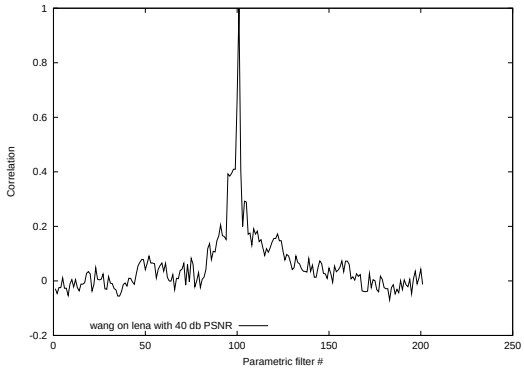
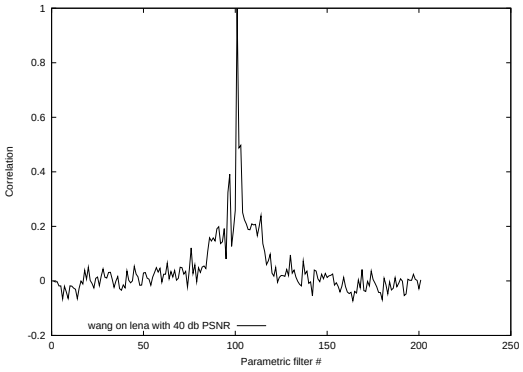


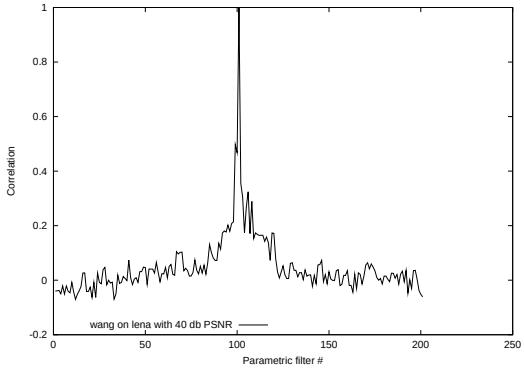
Figure 2.21: Wang — One filter with six parameters — Overview



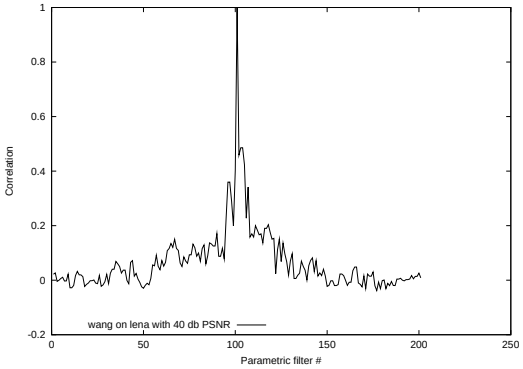
(a) Detail A



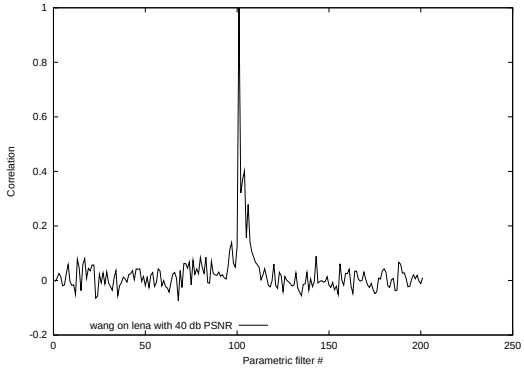
(b) Detail B



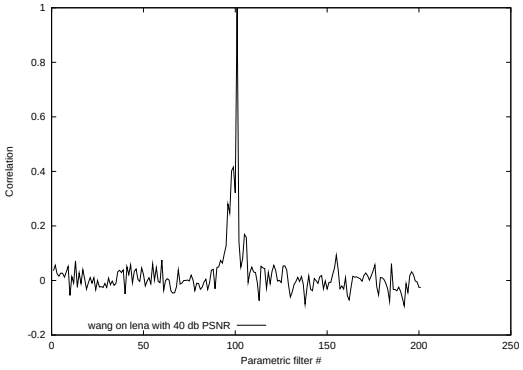
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

Figure 2.22: Wang — One filter with six parameters — Details

Wang — Six Filters each with One Parameter

Diagrams 2.23 and 2.24 show the six parameters distributed over six filters. With the Wang embedding scheme we see very good results. Diagrams 2.24 (a)-(c) look very good and only have one clear peak. Diagrams 2.24 (d)-(e) have larger areas of high correlation, but still show that even the last parameter has an effect on the correlation. The security of using the Wang method therefore is greater than using the Kim method, because we can use the parameters on all decomposition levels as keys.

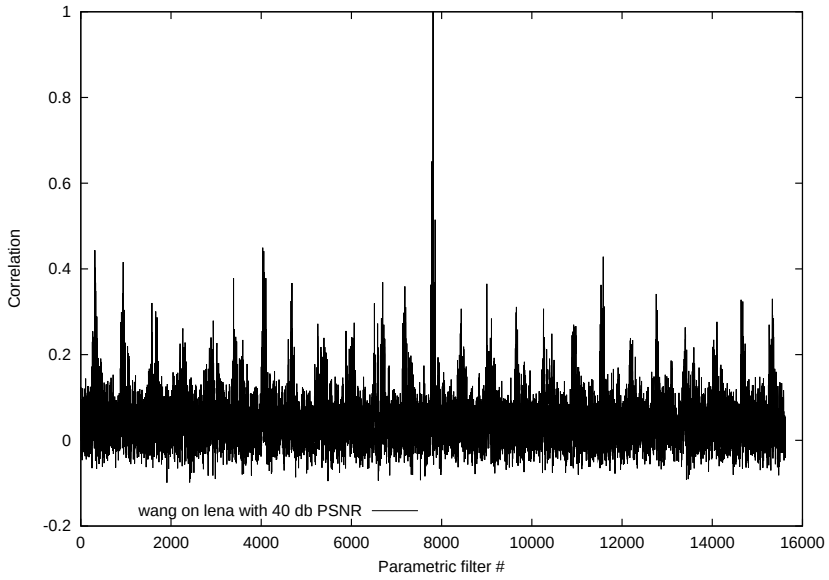
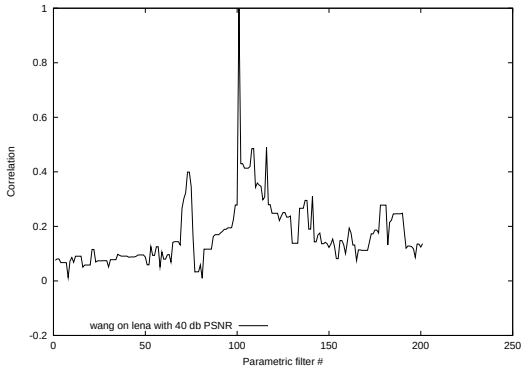
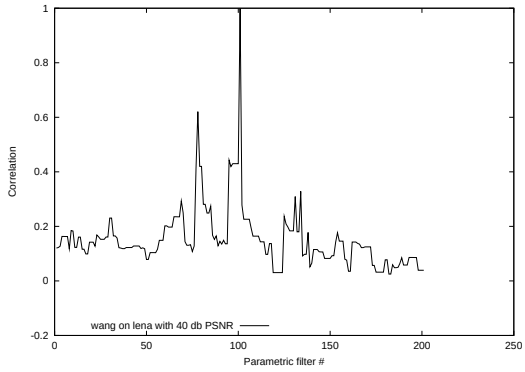


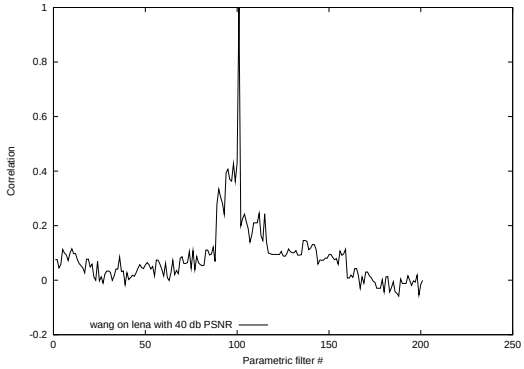
Figure 2.23: Wang — Six filters each with one parameter — Overview



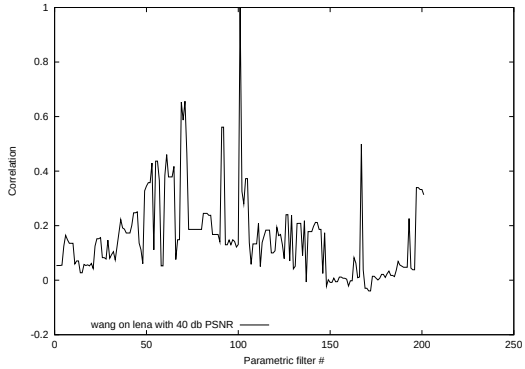
(a) Detail A



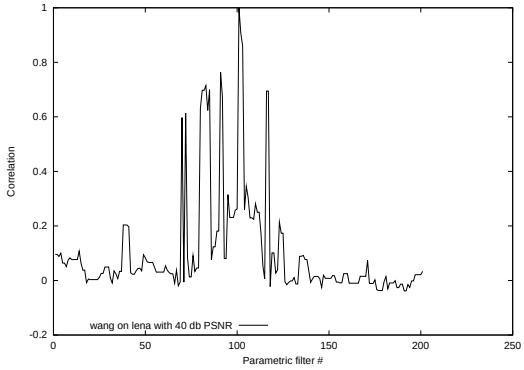
(b) Detail B



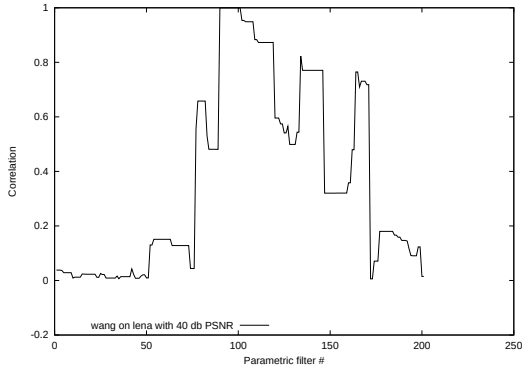
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

Figure 2.24: Wang — Six filters each with one parameter — Details

Wang — Mix 3-2-1

Diagrams 2.25 and 2.26 show the overview and detail results for the "mix-3-2-1" system. There are four parametrizations with a correlation above 0.60. In comparison to the Kim algorithm we see that all 6 parameters have a clear influence on the correlation, although the average correlation seems to be higher for the Wang method.

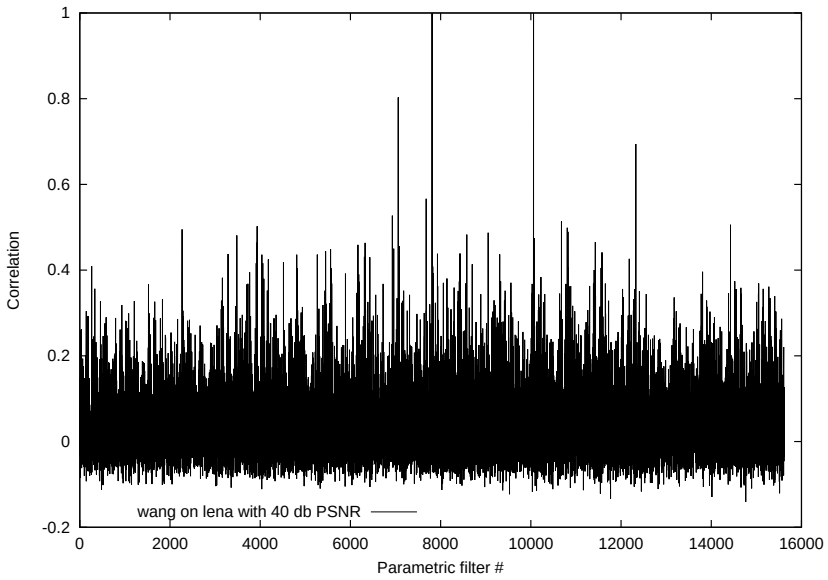
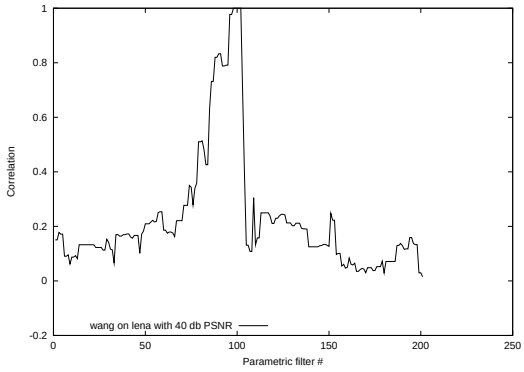
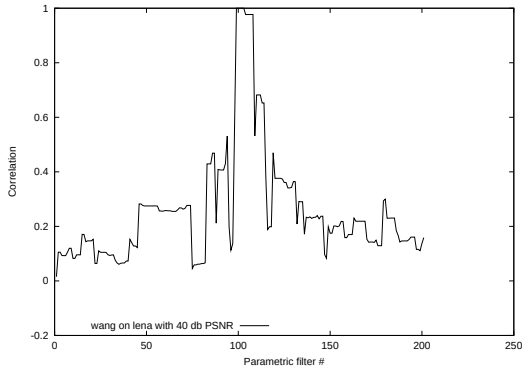


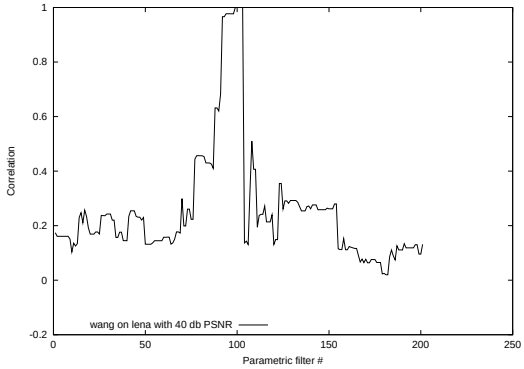
Figure 2.25: Wang — Mix 3-2-1 — Overview



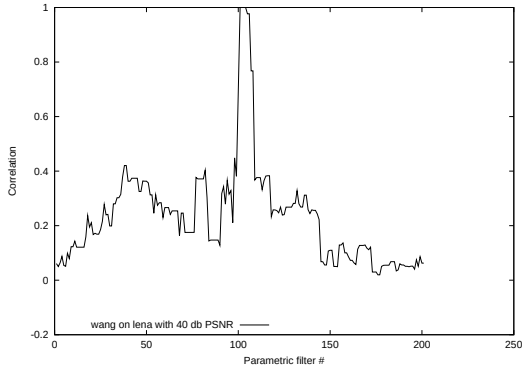
(a) Detail A



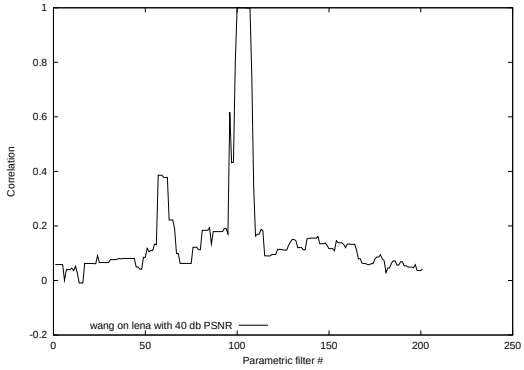
(b) Detail B



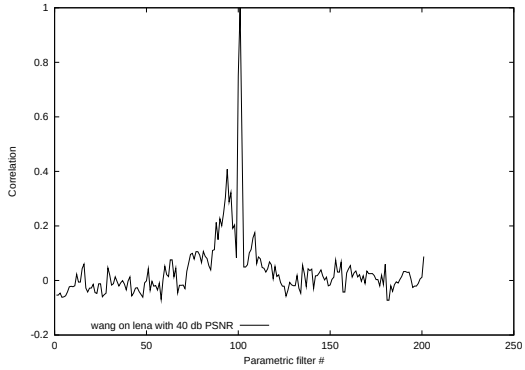
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

Figure 2.26: Wang — Mix 3-2-1 — Details

Wang — Mix 1-3-2

The behavior of the "mix-1-3-2" system is shown in diagrams 2.27 and 2.28. There are only a few peaks with a correlation above 0.70 and all six parameters show influence on the correlation value. We see that using the Wang watermarking method with non-stationary MRA shows higher security than using the Kim method. All parameters from all decomposition levels have an influence on the correlation result. NSMRA can be used to increase the parameter-space and enhance the security of the system.

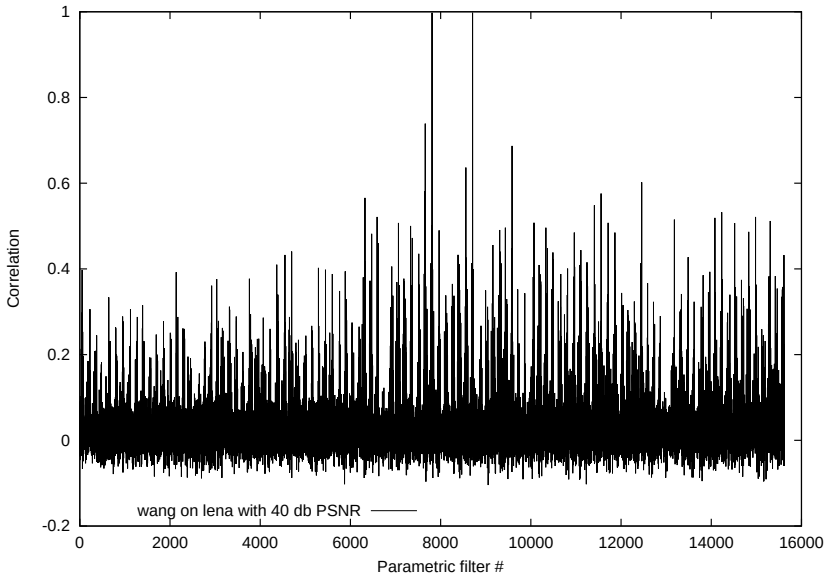
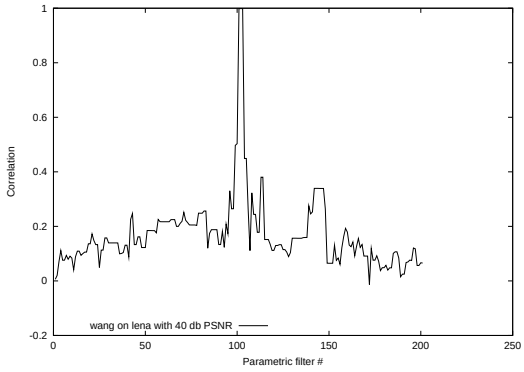
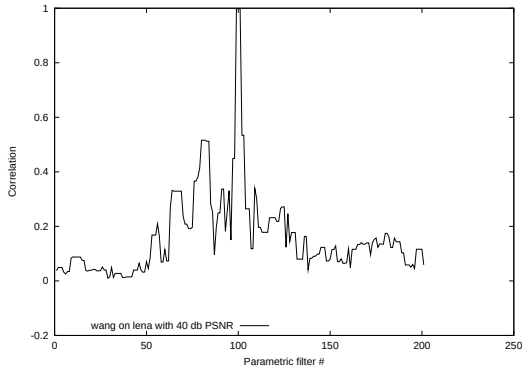


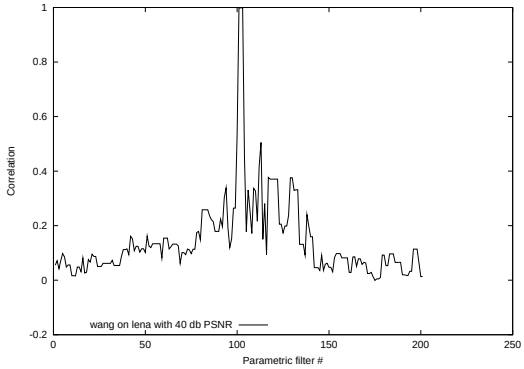
Figure 2.27: Wang — Mix 1-3-2 — Overview



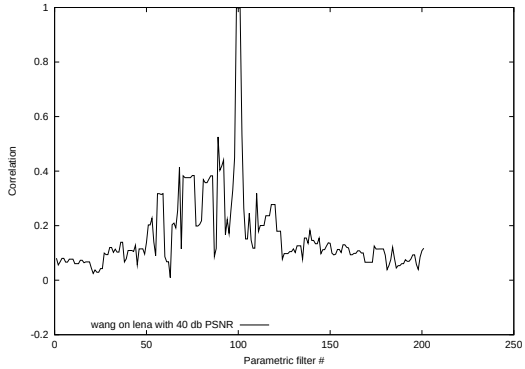
(a) Detail A



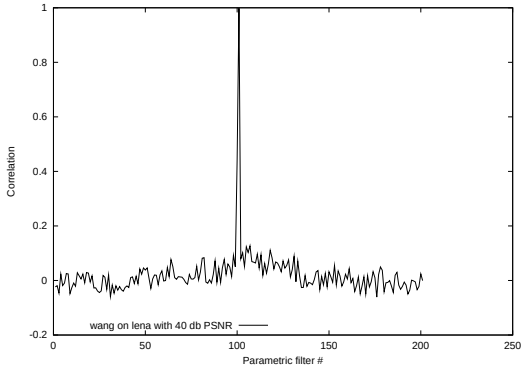
(b) Detail B



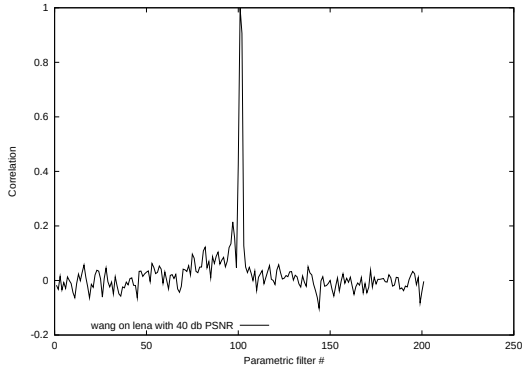
(c) Detail C



(d) Detail D



(e) Detail E



(f) Detail F

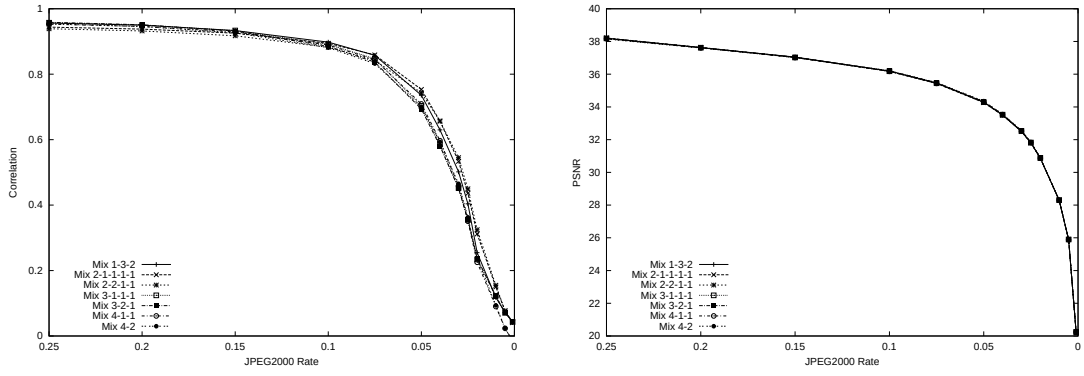
Figure 2.28: Wang — Mix 1-3-2 — Details

2.4.2 Quality Assessment

To analyze the influence of the parametrized filters on the image quality and correlation under compression we embed a watermark with different filters. Each filter parameter α_i is chosen from $\{-1.5, 0.50, 2.5\}$, therefore we assess $3^6 = 729$ different parametrizations. Again we choose an embedding strength that results in 40dB PSNR with the "Lena" image. Then we compress the watermarked images using JPEG and JPEG2000 with different quality levels.
We measure the PSNR of the compressed image to see the effect of the parameterized filters on the image quality. We also try to detect the watermark in the compressed image with the known parameters and measure the resulting correlation between the embedded and the extracted watermarks.
Then the minimum, maximum and average of all parametrized filters are calculated and used for comparing the different parameter distributions.

Kim Algorithm

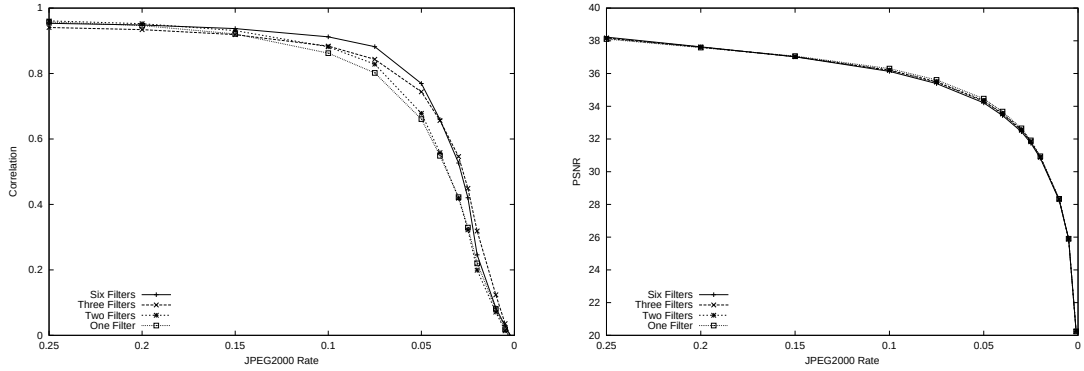
Diagram 2.29(a) shows the average correlation behavior under JPEG2000 compression for the seven systems that use a different number of parameters for different levels. For short we call these systems the "Mixed" systems. In contrast diagram 2.30(a) shows the average behavior of the "Pure" systems that use the same number of parameters for all decomposition levels.
In both diagrams you can see that the correlation after strong compression is better when the filter was generated from fewer parameters. Especially diagram 2.30(a) for the pure systems makes this clear. For a JPEG2000 compression rate of 0.05 the system that uses all six parameters to create the filters has a correlation of around 0.1 below the correlation for the system that uses each parameter to create six different filters.
Diagrams 2.29(b) and 2.30(b) show the average PSNR after compression with JPEG2000 at different rates. The difference between the systems is very small.
Finally diagrams 2.31(a) and 2.32(a) show the minimum, maximum and average correlation values under JPEG2000 compression. Diagram 2.31(a) is for the system that uses all parameters to create one filter and diagram 2.32(a) is for the system that uses every parameter to create a different filter. The corresponding PSNR values can be seen in diagrams 2.31(b) and 2.32(b).



(a) Correlation

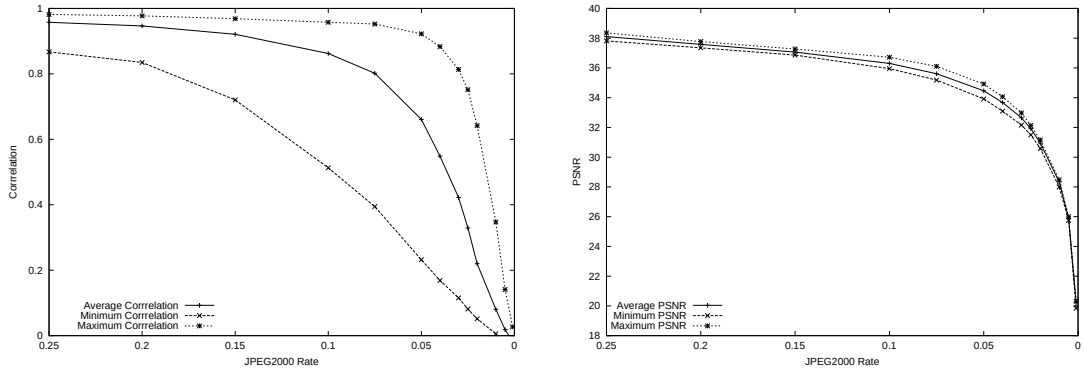
(b) PSNR

Figure 2.29: Kim — Correlation and PSNR under JPEG2000 compression — Mixed



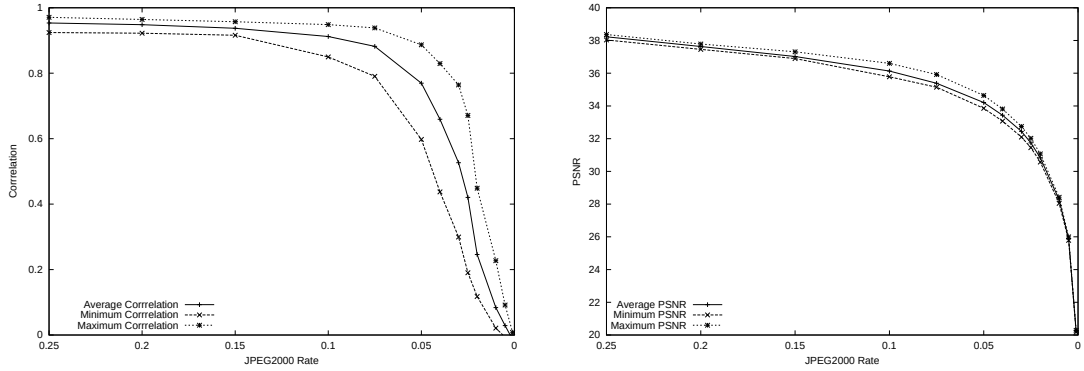
(a) Correlation (b) PSNR

Figure 2.30: Kim — Correlation and PSNR under JPEG2000 compression — Pure



(a) Correlation (b) PSNR

Figure 2.31: Kim — Correlation and PSNR under JPEG2000 compression — One filter

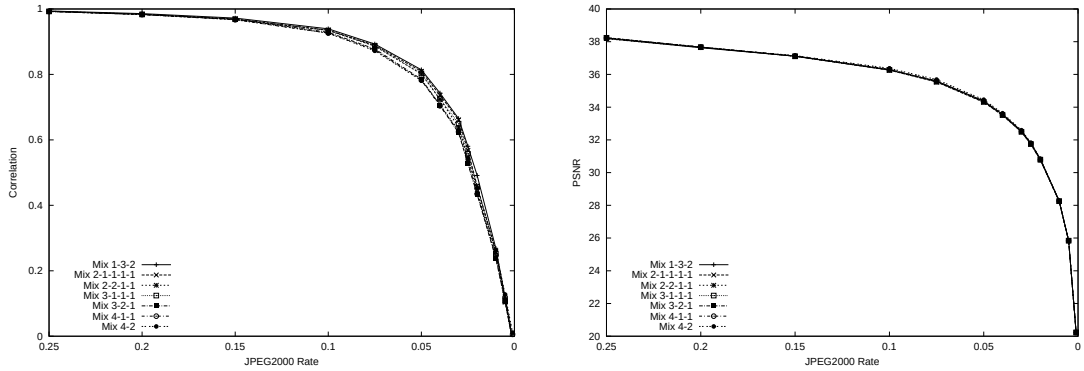


(a) Correlation (b) PSNR

Figure 2.32: Kim — Correlation and PSNR under JPEG2000 compression — Six filters

Wang Algorithm

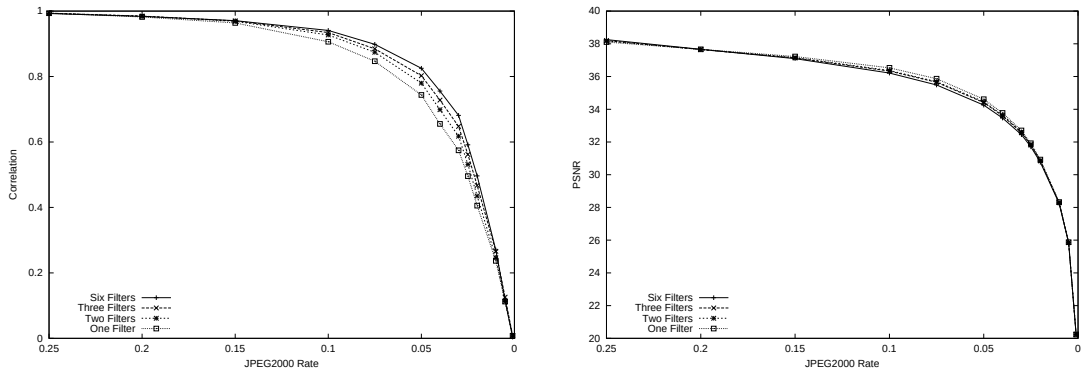
The results for the Wang algorithm look very similar to those for the Kim algorithm. But because we already saw that the Wang algorithm offers significantly higher security we are going to show more quality results for this algorithm.
Diagram 2.33(a) shows the average correlation behavior under JPEG2000 compression for the mixed systems. The behavior of the pure systems is shown in diagram 2.34(a). It is again clear to see that for compression rates between 0.15 and 0.025 the shorter filters show better resistance to the compression. For a JPEG2000 rate of 0.05 the correlation for the single-parameter filters is around 0.1 higher than the correlation for the six-parameter filter.
The average PSNR under JPEG2000 compression is shown in diagram 2.33(b) for the mixed systems and in diagram 2.34(b) for the pure systems. Again there is little difference in the PSNR values. In diagram 2.34(b) you can see that the difference between the one-filter and the six-filter systems at a compression rate of 0.05 is only 0.35 dB in favour of the one-filter system.
In diagram 2.35(a) you can see the average correlation of the mixed systems under JPEG compression. Diagram 2.36(a) shows the average correlation value under JPEG compression with different quality factors. Here the filters that were generated by one or two parameters are clearly above the filters generated by six parameters. The difference at a quality factor of 10 is 0.08 in advantage of the shorter filters.
The behavior of the average PSNR of the pure systems under JPEG compression is shown in diagram 2.36(b). Again the longer filters are only slightly better than the shorter filters. At a JPEG quality of 50 the one-filter system has a 0.34 dB higher PSNR.
The minimum, maximum and average correlation under JPEG2000 compression is shown in diagram 2.37(a) for the one-filter system and in diagram 2.38(a) for the six-filter system. Both systems vary with nearly the same bandwidth around their average value.
Diagrams 2.37(b) and 2.38(b) show the minimum, maximum and average PSNR under JPEG2000 compression for the one-filter system and the six-filter system. The difference between the minimum and maximum is 1.36 dB for the one-filter system at a compression rate of 0.1 and 1.34 dB for the six-filter system.



(a) Correlation

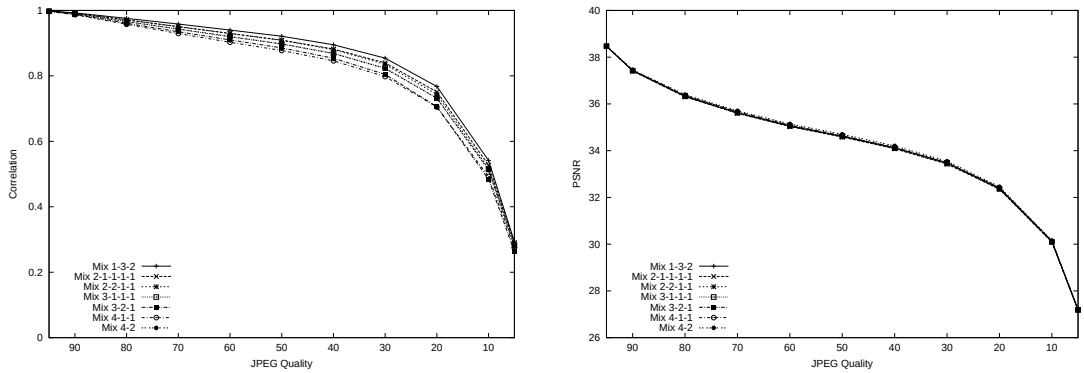
(b) PSNR

Figure 2.33: Wang — Correlation and PSNR under JPEG2000 compression — Mixed



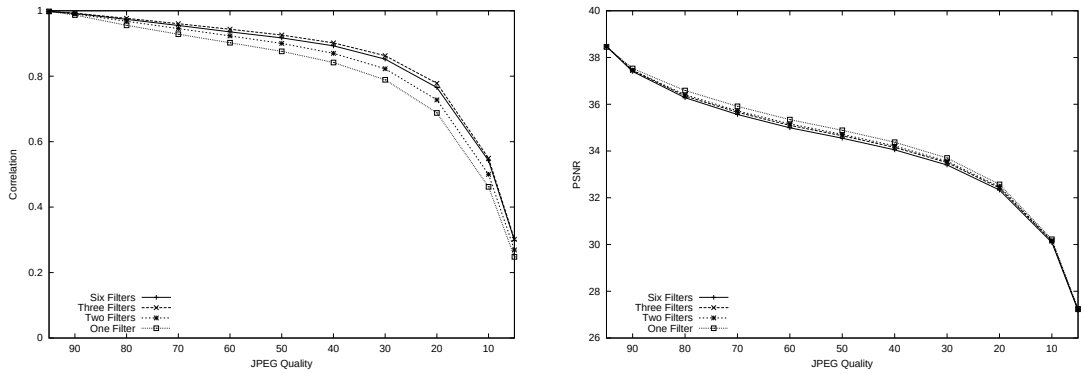
(a) Correlation (b) PSNR

Figure 2.34: Wang — Correlation and PSNR under JPEG2000 compression — Pure



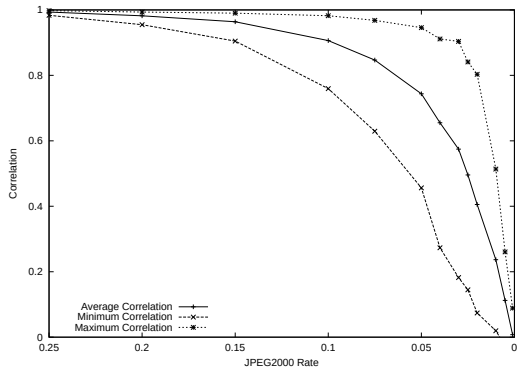
(a) Correlation (b) PSNR

Figure 2.35: Wang — Correlation and PSNR under JPEG compression — Mixed

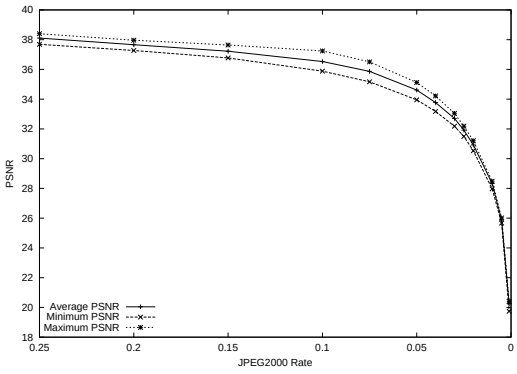


(a) Correlation (b) PSNR

Figure 2.36: Wang — Correlation and PSNR under JPEG compression — Pure

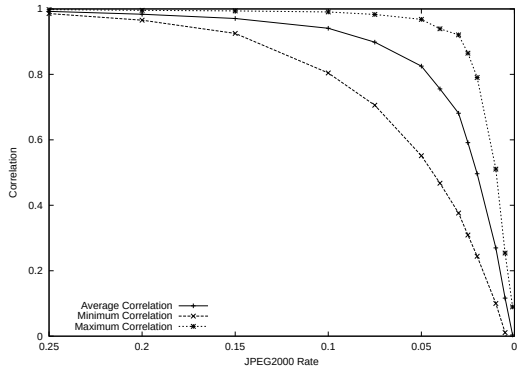


(a) Correlation

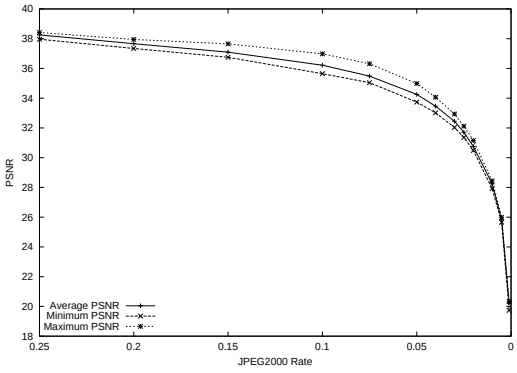


(b) PSNR

Figure 2.37: Wang — Correlation and PSNR under JPEG2000 compression — One filter



(a) Correlation



(b) PSNR

Figure 2.38: Wang — Correlation and PSNR under JPEG2000 compression — Six filters

2.5 1, 2, 3, 5, 6 and 9 Parameters

In the last section we looked at the distribution of six parameters over the different decomposition levels. For the quality assessment we varied the $\alpha_i \in \{-1.5, 0.50, 2.5\}$, which results in 729 different parametrizations. Now if the six parameters are used for one long filter we have $3^6 = 729$ different filters that are used for the decompositions. But if we use each single parameter to create a filter of length 4, then we only have 3 distinct filters that are distributed in different combinations over the six decomposition levels.

The different number of parametrizations used for the different filter lengths could result in skewed results. To make sure the different filter lengths produce comparable results we now look at filters with 1, 2, 3, 5, 6 and 9 parameters. For each length we choose parameters to get between 512 and 1024 different filter parametrizations. The same filter is used for all decomposition levels.

We now look at the results for the following filter parametrizations:

Parameters	Start Value	Stop Value	Delta	Parametrizations
1	-3.00	3.00	0.01	601
2	-3.00	3.00	0.25	625
3	-3.00	3.00	0.75	729

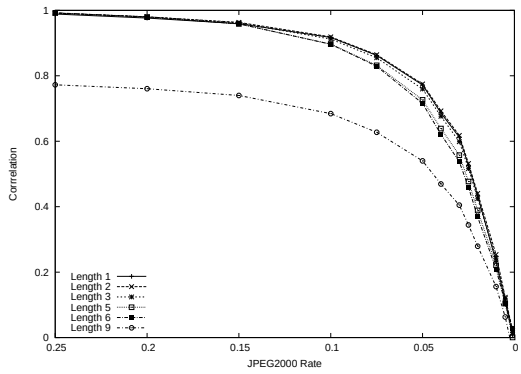
For 5, 6 and 9 parameters we can only use 4, 3 and 2 distinct parameter values per parameter to stay within the set number of different parametrizations. The following table shows the different values used for the parameters:

Parameters	Param	Start Value	Stop Value	Delta	Parametrizations
5					1024
	1	-3.00	3.00	2.00	-3.00, -1.00, 1.00, 3.00
	2	-2.50	3.00	1.50	-2.50, -1.00, 0.50, 2.00
	3	-1.75	3.00	1.50	-1.75, -0.25, 1.25, 2.75
	4	-2.25	3.00	1.50	-2.25, -0.75, 0.75, 2.25
	5	-1.25	3.00	1.25	-1.25, 0.00, 1.25, 2.50
6					729
	1	-2.50	3.00	2.00	-2.50, -0.50, 1.50
	2	-1.50	3.00	2.00	-1.50, 0.50, 2.50
	3	-2.10	3.00	2.00	-2.10, -0.10, 1.90
	4	-1.10	3.00	2.00	-1.10, 0.90, 2.90
	5	-2.90	3.00	2.00	-2.90, -0.90, 1.10
	6	-1.90	3.00	2.00	-1.90, 0.10, 2.10
9					512
	1	-2.50	3.00	3.00	-2.50, 0.50
	2	-1.50	3.00	3.00	-1.50, 1.50
	3	-0.50	3.00	3.00	-0.50, 2.50
	4	-2.50	3.00	3.00	-2.50, 0.50
	5	-0.50	3.00	3.00	-0.50, 2.50
	6	0.50	3.00	2.00	0.50, 2.50
	7	-2.50	3.00	3.00	-2.50, 0.50
	8	-0.50	3.00	3.00	-0.50, 2.50
	9	0.50	3.00	2.00	0.50, 2.50

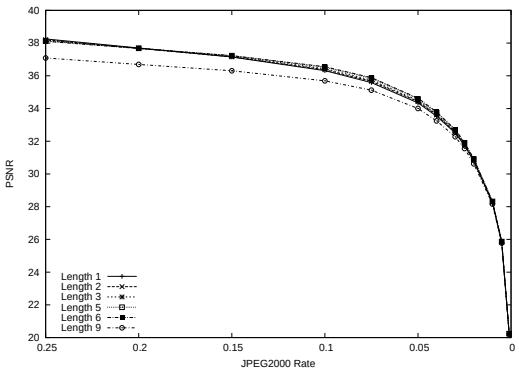
Each filter is used to embed a watermark with 40dB PSNR into the “Lena” image. Then the watermarked image is subjected to JPEG and JPEG2000 compression at different compression rates. We measure the correlation between the watermark signature that can be extracted from the compressed image and the orginally embedded watermark. We also measure the PSNR between the compressed image and the original.

From all the measurements for one filter length we calculate the average correlation and PSNR values and also the maximum and minimum values for each compression rate.

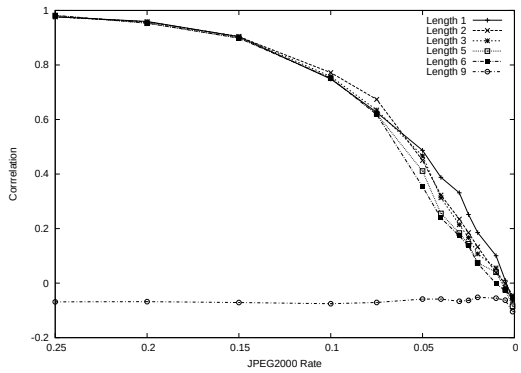
From previous results we know that the Kim and Xia embedding algorithms do not offer the same level of security as the Wang algorithm. Therefore we did all of the following experiments only for the Wang algorithm.
Diagram 2.39(a) shows the average correlation under JPEG2000 compression. As expected the longer filters show worse behavior under compression.
What needs to be noted is the behavior of the nine parameter filters. They show an average correlation of around 0.20 below the other filters. A look at the minimum correlation depicted in diagram 2.39(c) helps to understand this. There are filters with which no correlation could be detected. Close inspection showed that for more than 10% of the 9 parameter filters the wavelet decomposition and composition could not be performed with 40dB PSNR. Even without modification from the watermarking algorithms these filters created distorted images with a quality below 40dB PSNR.
The maximum correlation values are depicted in diagram 2.39(e). Again the shorter filters show the better resistance to compression with JPEG2000.
The same behavior can be seen under JPEG compression. Diagram 2.40(a) shows the average correlation, diagram 2.40(c) the minimum correlation and diagram 2.40(e) the maximum correlation under JPEG compression. Filters with 1, 2 and 3 parameters are very close together, but are clearly separated from the 5 and 6 parameter filters. We again see that the average value for the 9 parameter case is around 0.20 below the other values and that the minimum correlation is close to zero.
The influence of the different number of filter parameters on the image quality (as measured by the PSNR) is minimal. Diagram 2.39(b) shows the average PSNR under JPEG2000 compression. The longer filters show only a slight advantage for compression rates between 0.15 and 0.025. The minimum PSNR values are depicted in diagram 2.39(d). The worst behaving filter with nine parameters only achieved a PSNR of 20 dB. The maximum PSNR values are virtually on the same line and do not show a dependence on the filter length. They can be seen in diagram 2.39(f).
For JPEG compression the different behavior is more clearly visible. The average PSNR under JPEG compression is shown in diagram 2.40(b). From a compression quality of 90 down to 10 the longer filters have a higher PSNR than the shorter filters. But again the difference is very small.
The minimum PSNR behavior can be seen in diagram 2.40(d). Again we see that the worst nine parameter filters only achieve a PSNR of 20 dB. The maximum PSNR is shown in diagram 2.40(f). There is a clear difference between the long and the short filters.
From these results we see that the shorter filters do show better correlation behavior under compression. A comparable number of different filters was used for the different filter lengths to make sure that the results are not skewed. We also repeated the experiments with 512 random parameter selections and got the same results.



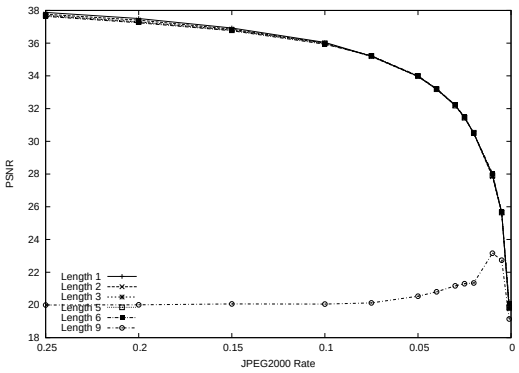
(a) Average Correlation



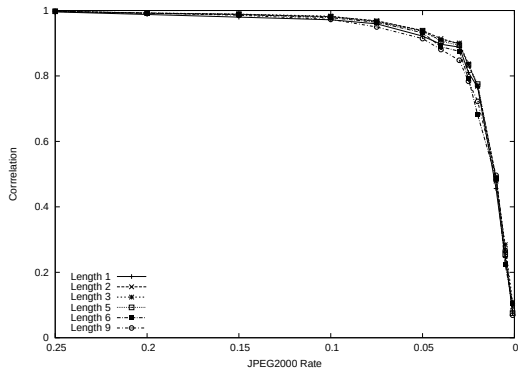
(b) Average PSNR



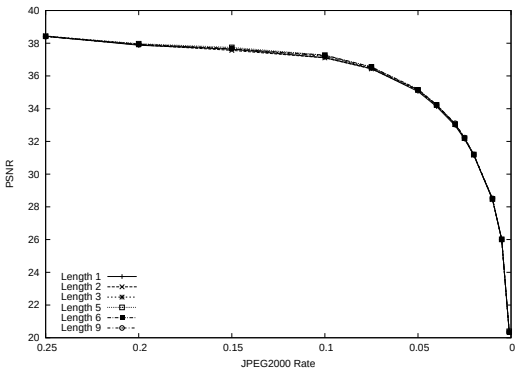
(c) Minimum Correlation



(d) Minimum PSNR

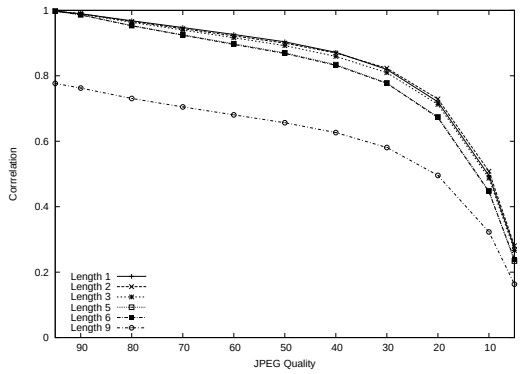


(e) Maximum Correlation

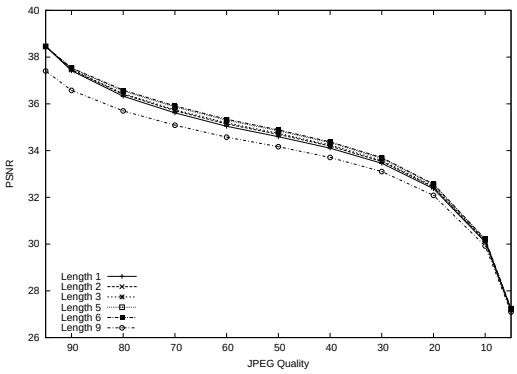


(f) Maximum PSNR

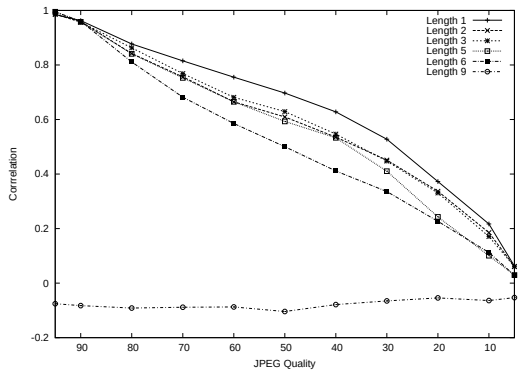
Figure 2.39: Correlation and PSNR under JPEG2000 compression



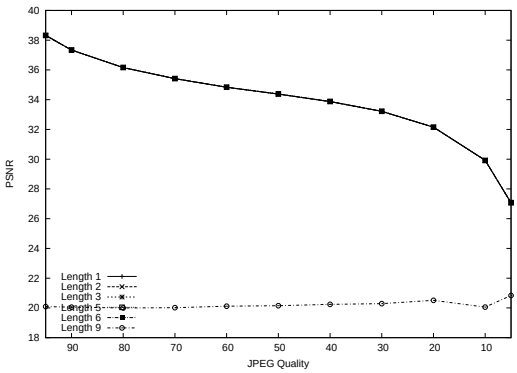
(a) Average Correlation



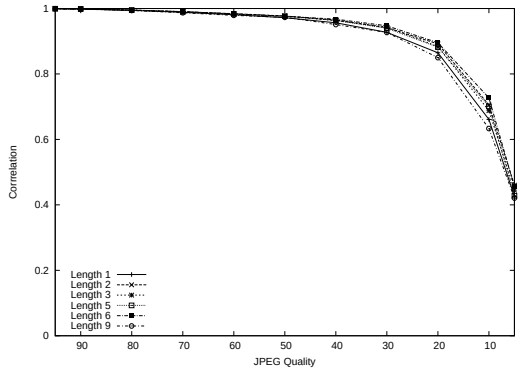
(b) Average PSNR



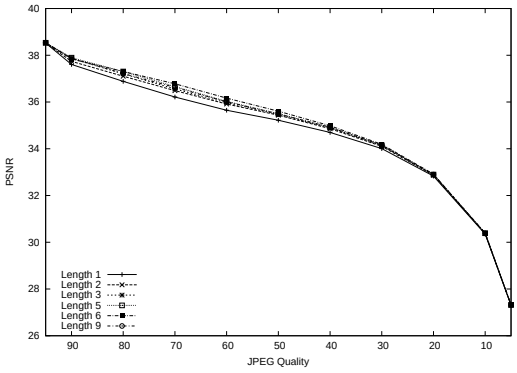
(c) Minimum Correlation



(d) Minimum PSNR



(e) Maximum Correlation



(f) Maximum PSNR

Figure 2.40: Correlation and PSNR under JPEG compression

2.6 Combined System

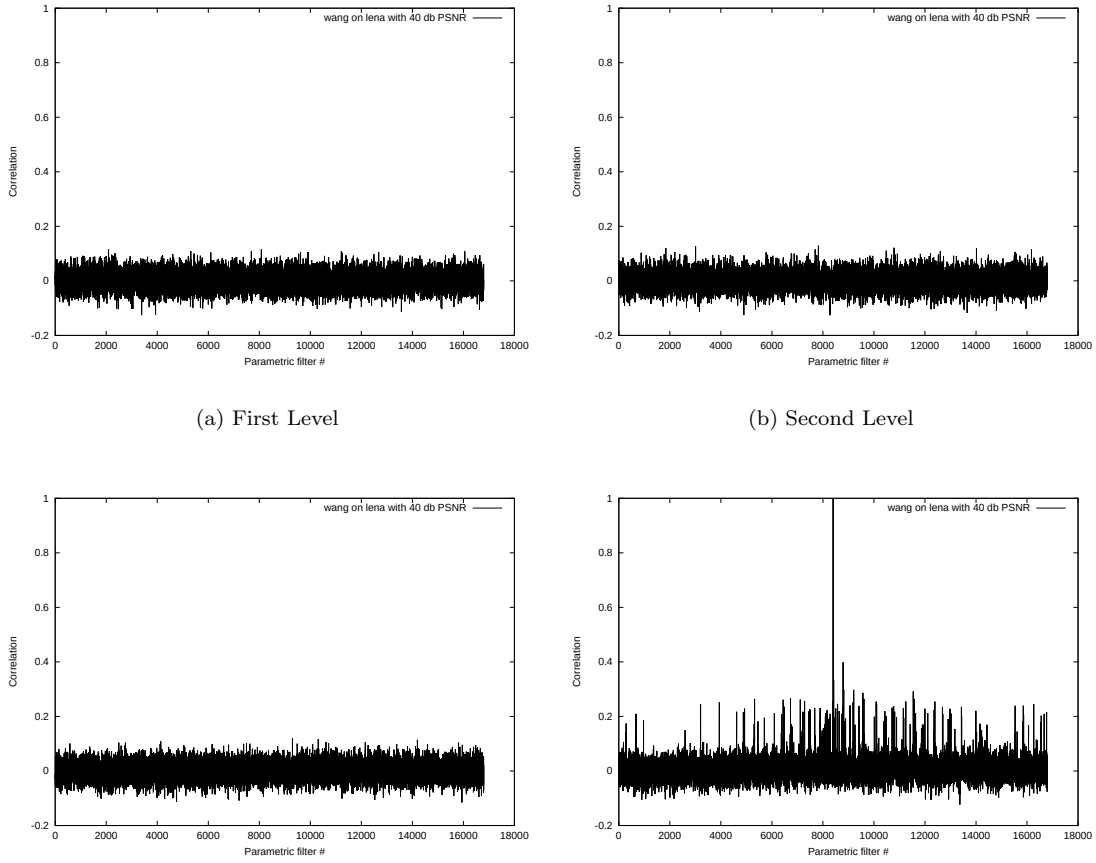
In the previous sections we investigated two methods to improve upon the security of wavelet-based watermarking systems. We have seen that with the Wang embedding system it is possible to use different filters for the different decomposition levels and have improved security. We also investigated the influence of using longer filters to have more parameters that can be used as embedding keys. We have seen that shorter filters with at most five or six parameters show better performance under JPEG and JPEG2000 compression than longer filters.
Now we investigate a system that combines the two approaches. We use filters that are generated by five parameters and use four filters for the first four decomposition levels. This results in a combined system with 20 parameters as embedding key.
In subsection 2.6.1 we look at the security of this combined system and in subsection 2.6.2 we look at the correlation and PSNR behavior under JPEG and JPEG2000 compression.

2.6.1 Security Assessment

For 20 parameters examining every parameter variation is not possible. Therefore we decided to look at the system and try to "attack" it. The attacker knows the basic design of the system, but does not know the key that was used for embedding. He in turn looks at the different decomposition levels and tries to independently guess the value of the 5 parameters used for that level. The parameters for the other levels are set to zero.
The watermark was embedded using the parameter values:

Parameter	1	2	3	4	5
Level 1	-0.5	2.5	-1.0	1.5	0.5
Level 2	-2.5	2.0	-2.0	0.5	1.0
Level 3	2.0	-1.5	0.5	2.5	-2.0
Level 4 + higher	-1.0	-2.0	1.0	-0.5	2.5

In the following diagrams we vary all five parameters for each level at the same time. We take the starting value 0.6 below the correct value for each of the five parameters. Then we increment the parameters by 0.2 until all parameters are 0.6 above the correct value. We therefore have $7^5 = 16807$ measurements for every level. If we would use a step size Δ of 0.2 for all 20 parameters and over the whole possible range we would get around $(2 * \pi/0.2)^{20} \approx 2^{99}$ possible filter parametrizations.
In diagram 2.41(a) we try to attack the first level. We vary the five parameters for the first decomposition level and keep the other parameters set to zero. From the correlation response to the different parametrizations there is no way for the attacker to guess the correct values.
If for some reason the attacker knows the correct parameter values for the first decomposition level, then he only needs to search for the remaining 15 parameters. Again we focus on the next set of five parameters used for the second decomposition level. The first decomposition is made with the correct filter and the third and higher filter parametrizations are unknown to the attacker. In diagram 2.41(b) we see that the attacker did not get any additional knowledge from knowing the first decomposition filter. The correlation is low over the complete range of guessed parametrizations, although the correct parametrization for the second decomposition level was tested.
The same is true for the third level. If we already have the first and second decomposition level parameters and only need to find the third and fourth level, then we will again first try to find the five parameters for the third decomposition. Diagram 2.41(c) shows the correlation when the first two levels are set to the correct keys, the fourth level is set to zero and the five parameters for the third level are varied over a set of parametrizations that contain the correct parameter values. Again there is no sign which of the tested parametrizations is the correct one and the attacker has no way of knowing that he has tested the correct parameters for the third level.



(a) First Level

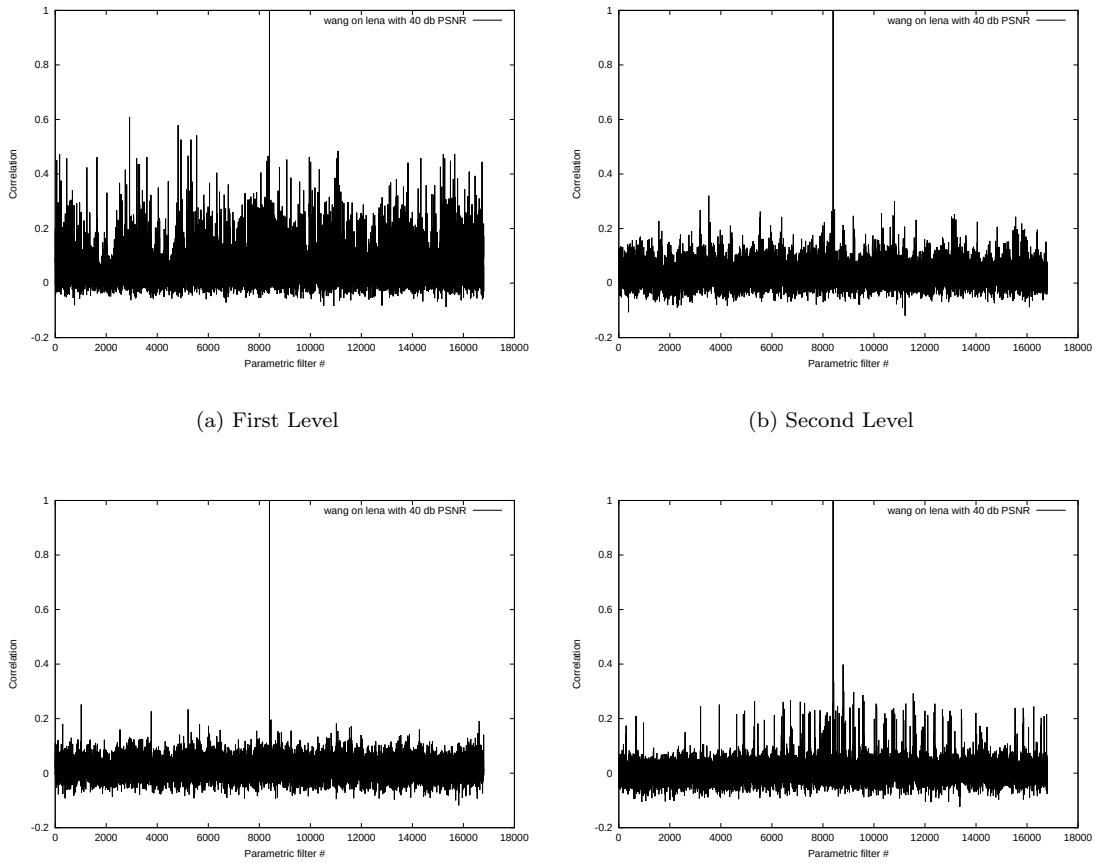
(b) Second Level

(c) Third Level

(d) Fourth Level

Figure 2.41: Attacks on all four levels

Only when the attacker already knows the decomposition parameters for the first three levels does he see the correct parameters for the fourth level. In diagram 2.41(d) the attacker already knows the first 15 keys and only varies the last five parameters for the fourth decomposition level. There is a clear peak at the location of the correct embedding parametrization.
So only after having all 20 parameters right does the attacker get a high correlation.
Next we look at the sensitivity of the different levels to parameter changes. We set all levels to the correct embedding parameters and only vary the parameters for one level and measure the correlation. In diagram 2.42(a) we vary the parameters for the first level and have the correct values for the other three levels. There is one peak at the embedding position and low variation everywhere else.
For levels two and three we see the respective results in diagrams 2.42(b) and 2.42(c). For the fourth level this is the same as the attack on the fourth level. The results are shown again in diagram 2.42(d).
There is higher correlation for the parameters on the first level, but there is one clear peak. For the second, third and fourth levels there is one peak and generally very low correlation.
We also investigated whether guessing the filter parameters "bottom-up" would work better. We set the parameters for the first three levels to zero and only investigated the fourth level. But there was no hightened correlation for the correct parametrization for that level.
To investigate the behavior at a finer resolution we performed the following experiment. We set all parameter values to the correct parameters and only vary the third parameter of every level,



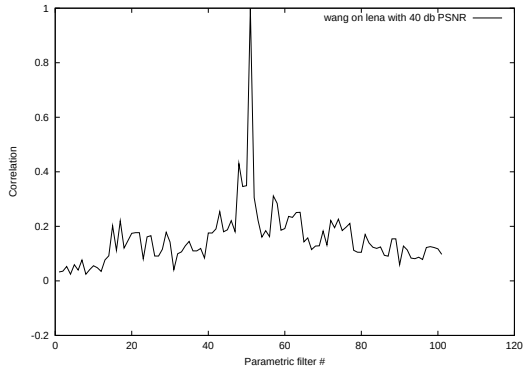
(a) First Level

(b) Second Level

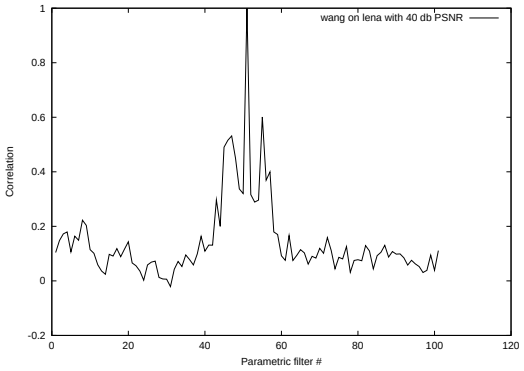
(c) Third Level

(d) Fourth Level

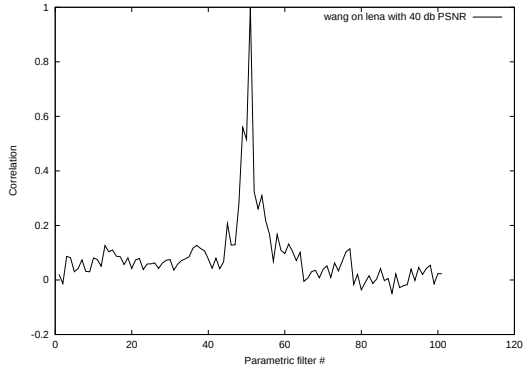
Figure 2.42: Variations of all levels



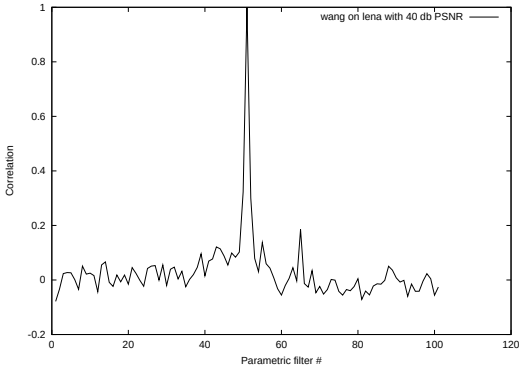
(a) Parameter 3



(b) Parameter 8



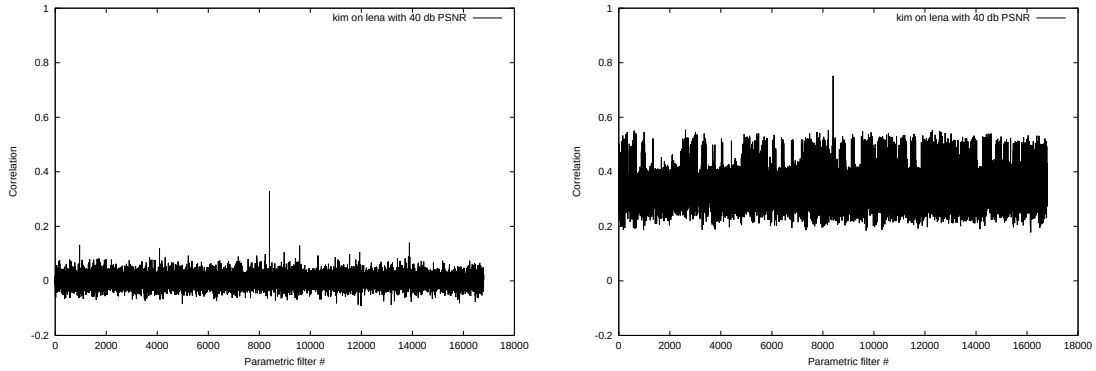
(c) Parameter 13



(d) Parameter 18

Figure 2.43: Variations of single parameters

corresponding to parameters 3, 8, 13 and 18. We start 0.5 below the correct parameter value and increment until the value is 0.5 above the correct value with a step size of 0.01. This results in 101 measurements. When we use $\Delta = 0.01$ for all 20 parameters, then we get $(2 * \pi/0.01)^{20} \approx 2^{185}$ possible filter parametrizations.
In diagrams 2.43 we see that there is a clear peak at the correct parameter value and a low correlation for other parameters.
To give you a clear view of the security advantage of using the Wang embedding scheme we show the attacks on the first and second levels for the Kim method in diagrams 2.44(a) and 2.44(b).
For diagram 2.44(a) we varied the five parameters for the first level and set all remaining 15 parameters to zero. There is one clear peak with a correlation of around 0.30. So although we do not get a 1.00 correlation by only guessing the first level we still see a significantly higher correlation for the correct parameters for the first level. Now by using the correct filters for the first level, setting the third and higher levels to zero and only varying the second level parameters we get diagram 2.44(b). The first thing to notice is that the correlation is above 0.20 for all tested parametrizations. But again there is a significantly higher correlation for the correct filter parametrization. We see the same behavior for the third and fourth levels, only that the general correlation gets higher and higher.
We see from this last experiment that only by using the Wang embedding algorithm we get the real security of all 20 parameters.
Using 20 filter parameters we get a vast key space. If we use a resolution of 0.20 we have around



(a) First Level

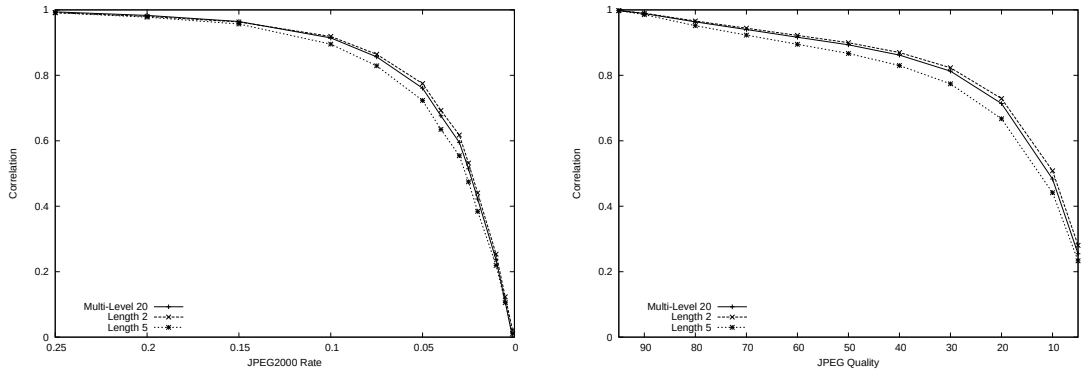
(b) Second Level

Figure 2.44: Kim — Attacks on first and second level

$(2 * \pi/0.20)^{20} \approx 2^{99}$ possible filter parametrizations. For the finer resolution of 0.01 we get $(2*\pi/0.01)^{20} \approx 2^{185}$ possible filters. If we choose a parameter resolution between 0.20 and 0.01 we have a very large key space and have very good separation between the correct key and incorrect embedding parametrizations.

2.6.2 Quality Assessment

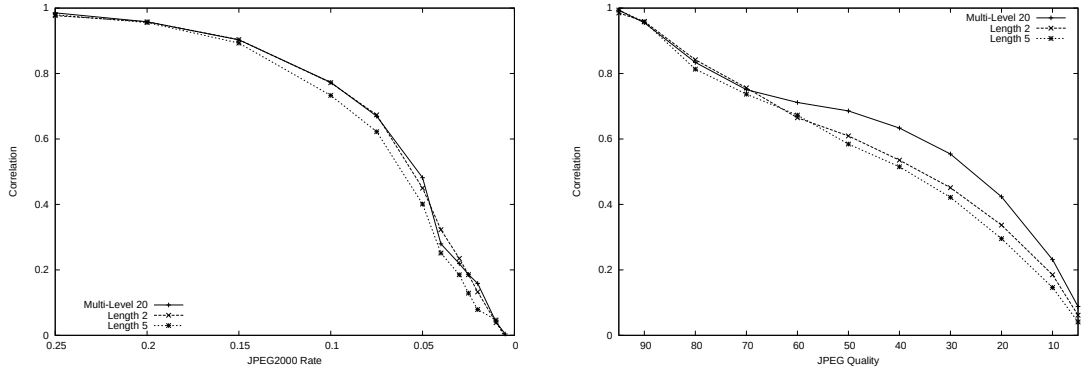
In this investigation we look at the correlation and PSNR behavior of the combined system under JPEG and JPEG2000 compression. We create 768 different filter parametrizations by randomly choosing values for the 20 parameters between -3.14 and 3.14. For each parametrization we embed a given watermark with an embedding strength that results in 40dB PSNR into the Lena image. Then we compress the watermarked image with JPEG and JPEG2000 at different compression rates. We try to detect the watermark in the compressed images and measure the correlation between the extracted watermark and the embedded one. Also the PSNR is measured to determine how strongly distorted the compressed image is.
We calculate the average, minimum and maximum values from all 768 different parametrizations and compare the results for the combined system to the results for the systems with two and five parameters.
Diagram 2.45(a) shows the average correlation under JPEG2000 compression. The behavior of the combined system is very similar to the other two systems. It is a bit above the five parameter curve and slightly below the two parameter system. Also the maximum and minimum correlation under JPEG2000 compression behave very similarly — the corresponding results are shown in figures 2.47(a) and 2.46(a).



(a) JPEG2000 (b) JPEG

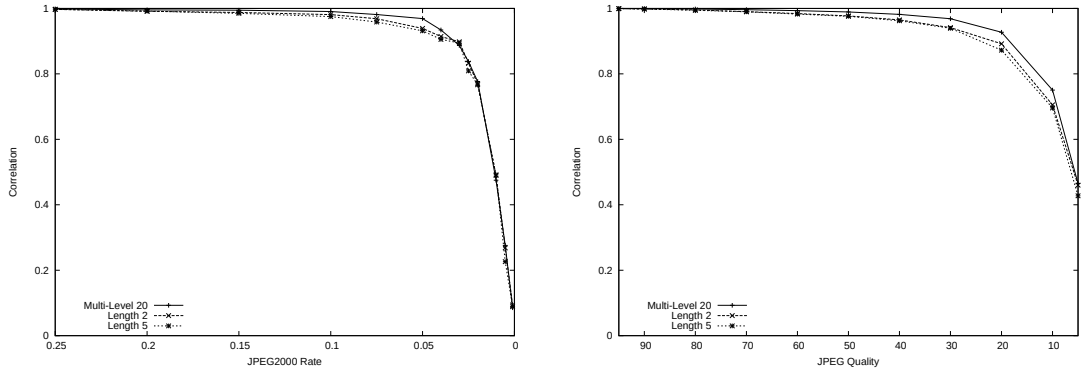
Figure 2.45: Average correlation under JPEG2000 and JPEG compression

The average correlation under JPEG compression is depicted in diagram 2.45(b). It also behaves close to the other systems. The maximum correlation is shown in diagram 2.47(b) and the minimum correlation in 2.46(b). Both are above the other two systems. Especially the minimum correlation of the combined system is around 0.10 above the other systems for JPEG compression qualities of 50 down to 20.
The PSNR behavior under compression shows very little dependency on the system used. Diagram 2.48(a) shows the average PSNR under JPEG2000 compression and diagram 2.48(b) the corresponding results under JPEG compression. All three curves are nearly identical for both JPEG and JPEG2000 compression. Also the maximum and minimum charts vary very little and are therefore not shown here.
In diagrams 2.49 and 2.50 we compare the minimum, maximum and average of the combined system with the behavior of the two standard filters Daubechies 6 and Biorthogonal 7/9.
Diagram 2.49(a) shows the correlation behavior under JPEG2000 compression, diagram (b) shows the PSNR. The corresponding values for JPEG compression are shown in diagram 2.50(a) and (b).
The average behavior of the combined system is very close to the two standard systems. The average correlation is a bit below the standard systems, but the PSNR is a little above the standard



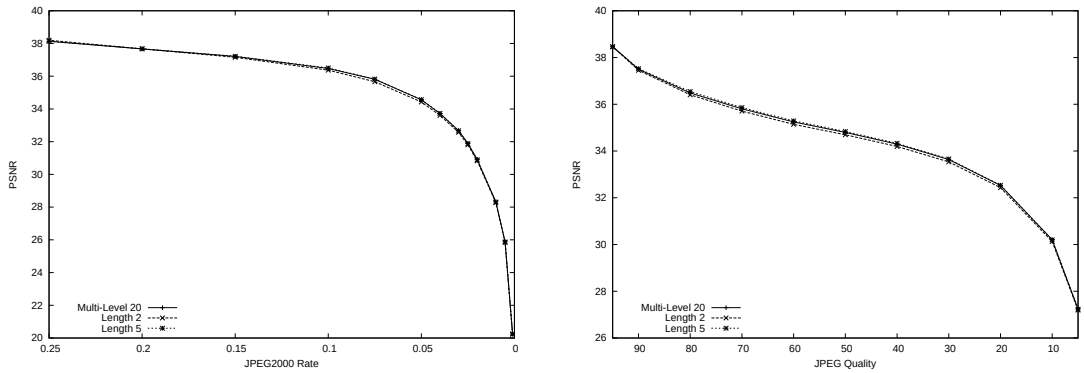
(a) JPEG2000 (b) JPEG

Figure 2.46: Minimum correlation under JPEG2000 and JPEG compression



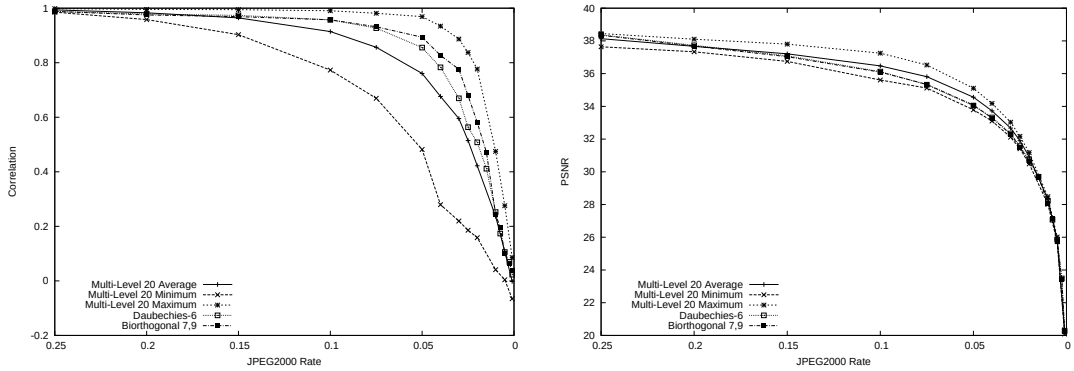
(a) JPEG2000 (b) JPEG

Figure 2.47: Maximum correlation under JPEG2000 and JPEG compression



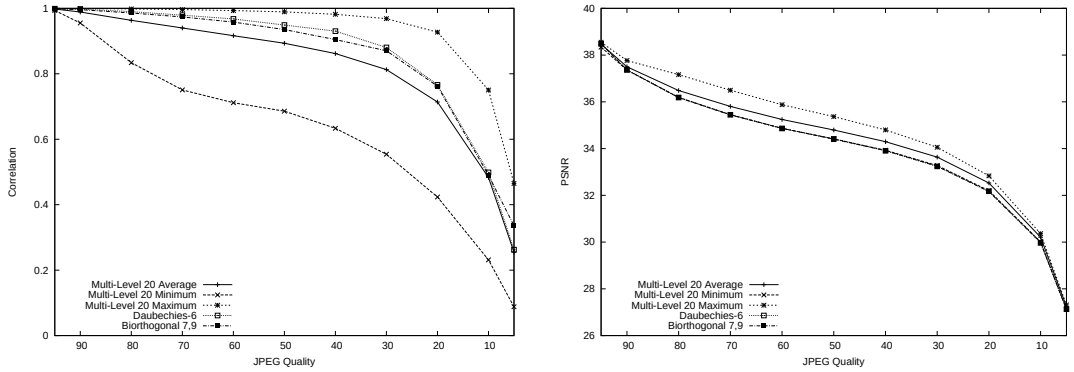
(a) JPEG2000 (b) JPEG

Figure 2.48: Average PSNR under JPEG2000 and JPEG compression



(a) JPEG2000 Correlation (b) JPEG2000 PSNR

Figure 2.49: Comparison of correlation and PSNR under JPEG2000 compression



(a) JPEG Correlation (b) JPEG PSNR

Figure 2.50: Comparison of correlation and PSNR under JPEG compression

systems. The maximum correlation line is above the standard filters which indicates that some parametrization generate better filters than the two standard filters. Even the minimum behavior is still reasonable.

2.7 Example Images

To visualize the influence of the different filter lengths we show a few example images. The watermarked image is on the left and the difference image between the watermarked image and the orignal image is on the right.
Please note that there is considerable variance for different parameter values. The shown images are only examples and should not be taken as necessarily typical for that system.

One Parameter

Figure 2.51 shows the images when only one filter parameter with the value 1.90 is used. You can see a low-frequency block structure with more energy around the edges and textured areas in the image.

Figure 2.51: One Parameter: 1.90

Three Parameters

In the three parameter case, seen in figure 2.52, the watermark pattern shows smaller blocks with a higher frequency pattern. Most differences are in areas with edges or textures.

Figure 2.52: Three Parameters: 1.50 -2.25 2.25

Five Parameters

For five parameters the difference image in figure 2.53 shows some high frequency components, but the contour of the original image is still visible.

Figure 2.53: Five Parameters: 1.00 2.00 -1.75 0.75 -1.25

Nine Parameters

A working filter is generated by the parameters

-2.50 1.50 -0.50 -2.50 -0.50 0.50 0.50 2.50 0.50

Figure 2.54 shows the watermarked and the difference images. The difference image is a high frequency pattern and no obvious correspondence to the edges and textured areas of the original image can be seen.

Figure 2.54: Nine Parameters – Working

A broken filter is generated by the nine parameters:

-2.50 -1.50 -0.50 -2.50 -0.50 0.50 0.50 -0.50 0.50

The corresponding images can be seen in Figure 2.55. There was no watermark embedded into this image. From decomposition and composition with this filter alone we get this horizontal pattern.

Figure 2.55: Nine Parameters – Broken

20 Parameters

Figure 2.56 depicts the results when the following 20 parameters are used:

Level 1	-3.007	2.615	-0.884	-2.151	0.561
Level 2	0.697	-2.523	1.329	0.351	1.235
Level 3	-1.613	0.872	-0.467	-1.258	1.050
Level 4 + higher	-2.756	-1.928	0.949	-2.039	-2.169

We get a block-pattern that puts more energy into regions of high frequency.

Figure 2.56: 20 Parameters – Example 1

Our final example can be seen in figure 2.57 and was generated using the 20 parameters:

Level 1	2.199	-2.428	2.010	-0.506	-1.062
Level 2	1.203	1.559	-2.050	1.343	-0.743
Level 3	0.569	-1.296	1.456	-1.152	2.901
Level 4 + higher	0.814	2.013	1.691	0.692	-2.754

Again the watermark energy is located in regions with edges and strong textures.

Figure 2.57: 20 Parameters – Example 2

2.8 Conclusions

In this chapter we proposed to use wavelet filter parametrization to create a secret transform domain in order to enhance wavelet based watermarking schemes from a security viewpoint.
Experiments reveal that spread-spectrum based watermarking schemes can be made resistant to unauthorized detection by our approach. Specifically we used watermarking algorithms developed by Kim, Wang and Xia and used them with parametrized wavelet filters.
First we examined the 2 parameter system in detail and assessed the security against unauthorized detection and the robustness against JPEG and JPEG2000 compression. All three algorithms show enhanced performance with parametrized wavelet filters.
Then we analyzed the effect of non-stationary decomposition where different filters are used for different decomposition levels. We concluded that the Wang method of selecting significant coefficients is best suited. For the Kim and Xia algorithms non-stationary decomposition is not improving the security as much as expected.
We also investigated the influence of the number of parameters on the robustness against JPEG and JPEG2000 compression. We have seen that more parameters lead to less resistance against compression. The use of 5 parameters to generate a filter seems to be a good compromise between possible key-space and robustness against compression.
Finally we presented a combined system that uses a total of 20 parameters. We use 5 parameters per filter and use four different filters for the non-stationary decomposition. This combination of filter parametrization and non-stationary wavelet decomposition achieves a keyspace of cryptographically reasonable size, with between 2^{99} and 2^{185} possible filters. Also, robustness against JPEG and JPEG2000 compression is on an equal level as compared to the use of standard wavelet filters.
Future work could analyze the influence of the selection of the embedding variation on the system security and determine whether some filter parametrizations are not suitable as embedding keys. The performance of the combined system could also be analyzed further and variations and selection criteria could be developed.

Chapter 3

Wavelet Packet Decomposition

This chapter explores the use of wavelet packet decompositions as a method to increase the security of watermarking systems.
In section 3.1 we introduce wavelet packets and then explain how we use them to increase the security of watermarking systems in section 3.2. Section 3.3 looks at the security properties and section 3.4 at the quality of the systems under compression. Finally in section 3.5 we show example images and tree decompositions and in section 3.6 we make some concluding remarks.

3.1 Theory and Previous Work

In the standard pyramidal wavelet decomposition we recursively apply the decomposition to the approximation subband alone. In figure 1.4 we see how the wavelet decomposition is only further applied to the approximation subband of the previous decomposition. The resulting subband structure is shown in figure 1.5.

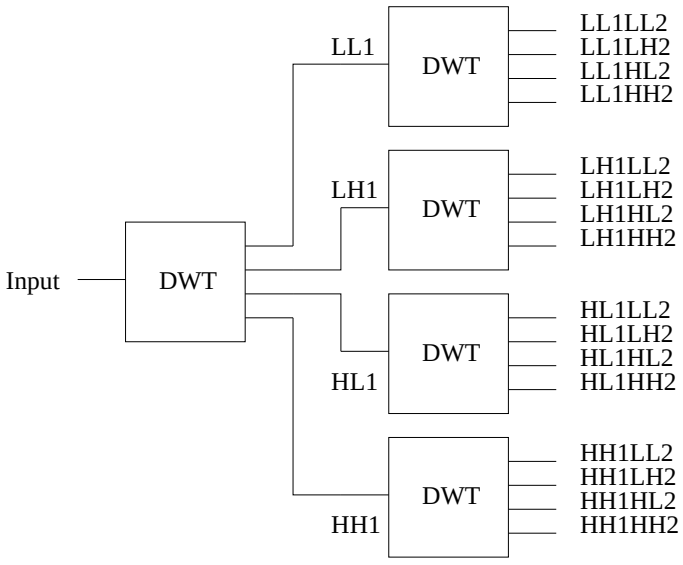


Figure 3.1: System design of two complete decomposition steps

For the wavelet packet decomposition we do not limit the decomposition to the approximation subband and allow further wavelet decomposition of all subbands on all levels. In figure 3.1 we show the system diagram for a complete two level wavelet packet decomposition and in figure 3.2

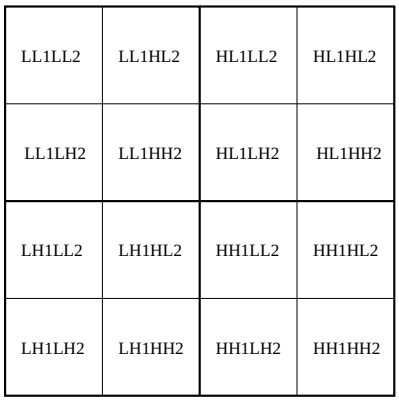


Figure 3.2: Subband structure after two complete decomposition steps

we show the resulting subband structure. We again use the simple decomposition step from 1.1 as basic building block.
The composition step is equal to the pyramidal case. All four subbands on one level are used as input for the inverse transformation and result in the subband on the higher level. This process is repeated until the original image is reproduced.
There are special algorithms to select the best decomposition for a specific input. For an introduction to wavelet packets and the best basis algorithm see [98]. This freedom in selecting the decomposition structure allows a better frequency selection and adoption to the specific image properties and there has been research for many application areas, including image and video compression and implementations on parallel systems. The United States FBI has defined one special wavelet decomposition that it uses for the compression of fingerprint images [7, 41]. Secret wavelet packet decompositions have also been used for multimedia encryption [76, 77, 78].
Wavelet packets have not found too much attention in the watermarking community yet. Wang [97] uses one non-standard decomposition to embed a watermark sequence in the middle frequencies of an image. The algorithm by Tsai [88] uses wavelet packets, but the selection is not specified and no experimental results are provided. One interesting approach is used by Vehel in [54]. The wavelet decomposition structure itself is used as the watermark sequence.
In the following we propose to embed the watermark sequence using a secret wavelet decomposition and to use the decomposition structure as embedding key. The watermark should only be detectable using that specific wavelet decomposition and we also expect that the security against malicious removal attacks is higher compared to the standard pyramidal decomposition.

3.2 Proposed Method

Our system is based on the Wang algorithm proposed in [96]. In the paper the authors already suggest to keep the wavelet decomposition structure secret, but no experimental results are provided. For a detailed description of the Wang algorithm see section 1.3.2 on page 10.
The basic system design is shown in figure 3.3. For the forward wavelet transformation we use a secret wavelet packet tree and then embed the watermark in the generated wavelet coefficients. After embedding the watermark we apply the inverse transformation using the same wavelet packet tree to generate the watermarked image.
The wavelet packet tree is generated by a random process that depends on a secret seed number. In the following we will also call this seed number either simply key or tree number. Two tree numbers that are close together do not necessarily generate similar trees.
We select a tree number and create a random wavelet packet tree using that number. This tree number is kept secret and is later needed to extract the watermark. When it is time to detect the watermark we use the secret tree number to generate the same wavelet packet tree again

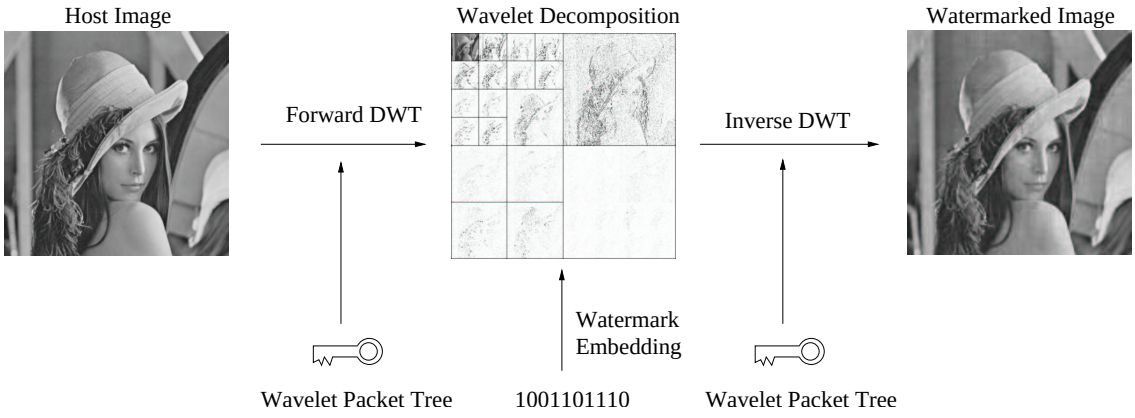


Figure 3.3: Basic system design

and extract the watermark sequence. Then we apply a normalized correlation calculation to the embedded and the extracted sequences and determine the likeliness that a watermark was embedded.
There is a vast number of possible wavelet packet trees. According to [78], for a decomposition with n+1 levels there are

$$f(n) = \sum_{i=0}^{4} \binom{4}{i} \cdot (f(n-1))^i \tag{3.1}$$

possible trees ($f(0) = 1$). For 4 decomposition levels this results in around 2^{65} trees and for 7 levels around 2^{4185} trees are possible.
For all decompositions we use the standard Biorthogonal 7/9 filter. A possible extension is to use the filter parametrization from the previous chapter in combination with the wavelet packet trees. In combination an even larger keyspace could be achieved.
For example tree decompositions and examples of watermarked images see section 3.5.

3.2.1 Tree Decompositions

We have three variables that influence the tree decomposition: the tree number, the number of decomposition levels and the decomposition method. We implemented two methods to randomly construct a tree. The first method randomly decomposes the subbands and the second one puts more focus on decomposing middle frequencies.

Random Tree Decomposition 1

For this method we initialise a random number generator with the tree number as seed and then use a 50 % probability for each subband to decide whether it should be further decomposed or not.
This decomposition strategy gives us the full range of possible decomposition trees, but could also result in trees that are generally not good for watermark embedding. For example, if the generated tree only applies decompositions to the detail subbands on all levels, then we are likely to embed the watermark in a high frequency domain which is more sensitive to image compression.

Random Tree Decomposition 2

We developed this decomposition specifically for our watermarking system. The focus is on building a wavelet tree that has a good resolution of the middle and low frequencies, which are best suited for watermark embedding. No decomposition of the three detail subbands on the first level

(HL_1, LH_1 and HH_1) is performed, only the first approximation subband is further decomposed. More emphasis is put on decomposing in the middle frequencies.
Using this decomposition strategy we basically loose all the trees that are below the three top-level detail subbands. Therefore we have only around 83521 trees for 4 levels, but still around 2^{1046} trees for 7 levels.

3.2.2 Embedding Variations

From the security analysis we learn that common subtrees can happen and can result in higher correlation even for wrong tree numbers. To protect against this we added three embedding variations that add another dependency on the tree number. Then two trees can have a common subtree, but through the embedding variation there is still enough difference between the two tree numbers to make the system secure.
Instead of using the tree number again as seed for the embedding variation we could use another number and use it as additional key element. But to limit the complexity of our analysis we simply reuse the tree number for the embedding variations.

Variation 1 — Tree-dependent Coefficient Skipping

This first variation skips a part of the selected significant coefficients, as proposed by Wang [96]. We use 95% of the coefficients that are selected. The disadvantage of coefficient skipping could be reduced robustness to compression and reduced capacity. We expect that using 95 percent of the coefficient results in very good robustness results and does not limit the capacity too severely.

Variation 2 — Tree-dependent Gaussian Sequence Multiplication

By multiplying with another sequence that was initialised by the tree number we expect to limit the influence of common subtrees. There should be no effect on the robustness or capacity, because we still use the same significant coefficients for watermark embedding.

Variation 3 — Tree-dependent Watermark Shuffling

For the last variation we create a permutation of the watermark sequence before we embed it into the wavelet coefficients. Depending on the tree number we shuffle the elements of the watermark and then embed them into the selected wavelet coefficients. This variation should not have an influence on the robustness or capacity of the watermark, again because we select the same coefficients for embedding.

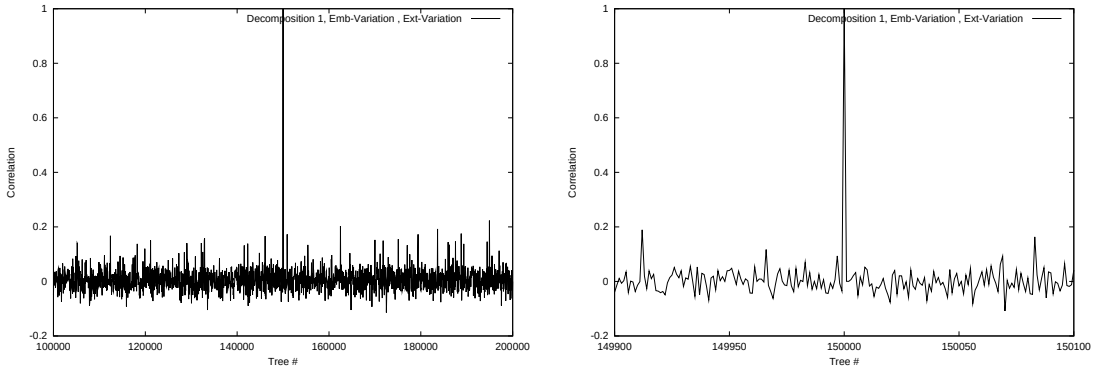
3.3 Security Assessment

For this security assessment we embed a watermark into the Lena image with 40dB PSNR. We use the tree that is generated by the tree number 150000 for embedding and then use two key sets for extraction. The big set starts at 100000 and goes up to 200000 in increments of 50 — giving 2001 measurements. The detail set starts at 149900 and goes up to 150100 in increments of 1 — giving 201 measurements.
We use two different decomposition depths — 4 and 7 levels. Because in the quality assessment we will look at longer watermark sequences, we will also assess the security for watermark lengths 1000, 5000 and 20000.
The most detailed analysis will be presented for 7 decomposition levels with a 1000 element watermark. For this case we will compare the performance of the first and the second tree decomposition methods. For the other combinations we only show the most significant subset of the available results.
Besides showing the effect of using the wrong key to extract the watermark we also look at the effect of using the wrong variation. When we embed the watermark with one of the variations we only want to be able to successfully extract the watermark with that variation.

3.3.1 7 Levels, Watermark Length 1000

Decomposition 1 — No Variation

Figure 3.4 shows the response for decomposition 1 without an embedding variation. There is one clear peak at 150000 and low correlation for all other tree numbers.

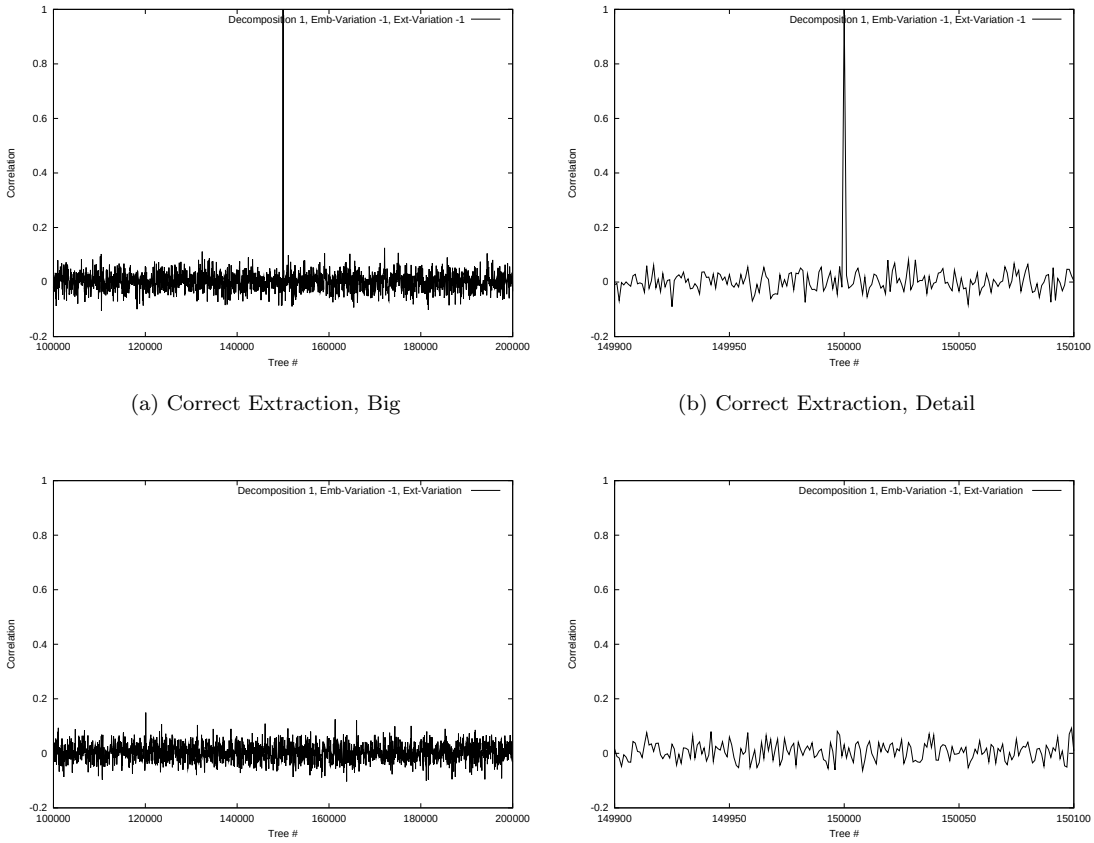


(a) Big (b) Detail

Figure 3.4: Decomposition 1, 7 levels, watermark length 1000, no variation

Decomposition 1 — Variation 1

In figure 3.5 we see the effect of embedding variation one — skipping some coefficients. There is again one clear peak and the correlation of wrong tree numbers was further decreased.
In subfigures (c) and (d) we see what happens when we do not use variation 1 for watermark extraction. There is low correlation for all tree numbers and the watermark is not found.



(a) Correct Extraction, Big

(b) Correct Extraction, Detail

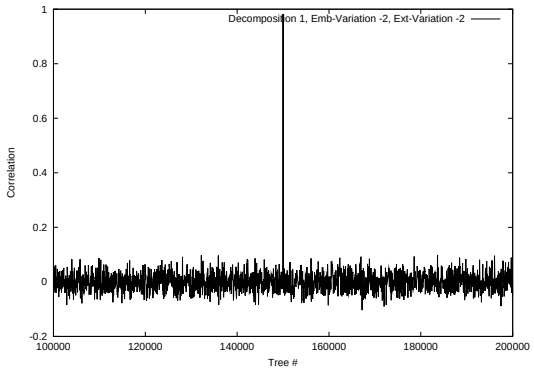
(c) Incorrect Extraction, Big

(d) Incorrect Extraction, Detail

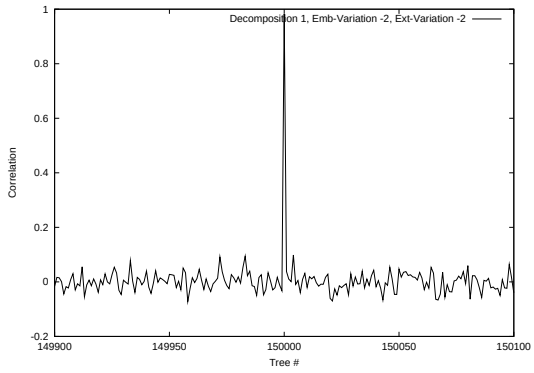
Figure 3.5: Decomposition 1, 7 levels, watermark length 1000, variation 1

Decomposition 1 — Variation 2

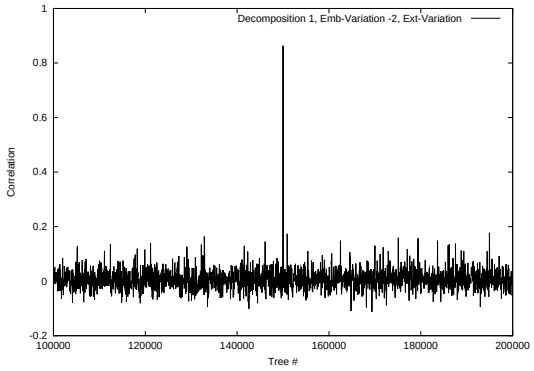
Figure 3.6 shows the measurements for variation 2 — multiplication with another gaussian sequence. 3.6(a) and (b) show the results when the wrong tree number is used. There is one peak and low correlation for all other tree numbers.
But in (c) and (d) we see a problem with variation 2. Even when we do not use variation 2 to extract the watermark, we still get high correlation for the correct tree number. This is because the extracted watermark has high correlation to the embedded watermark even without the additional multiplication with the tree-dependent gaussian sequence.



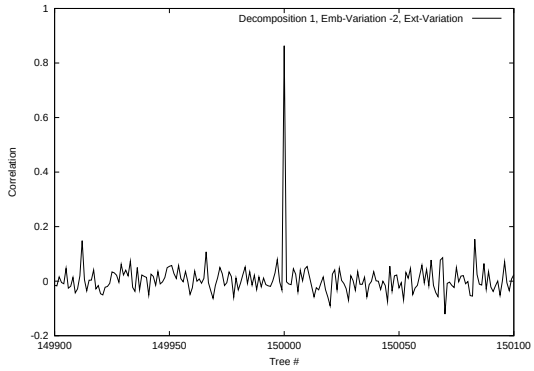
(a) Correct Extraction, Big



(b) Correct Extraction, Detail



(c) Incorrect Extraction, Big

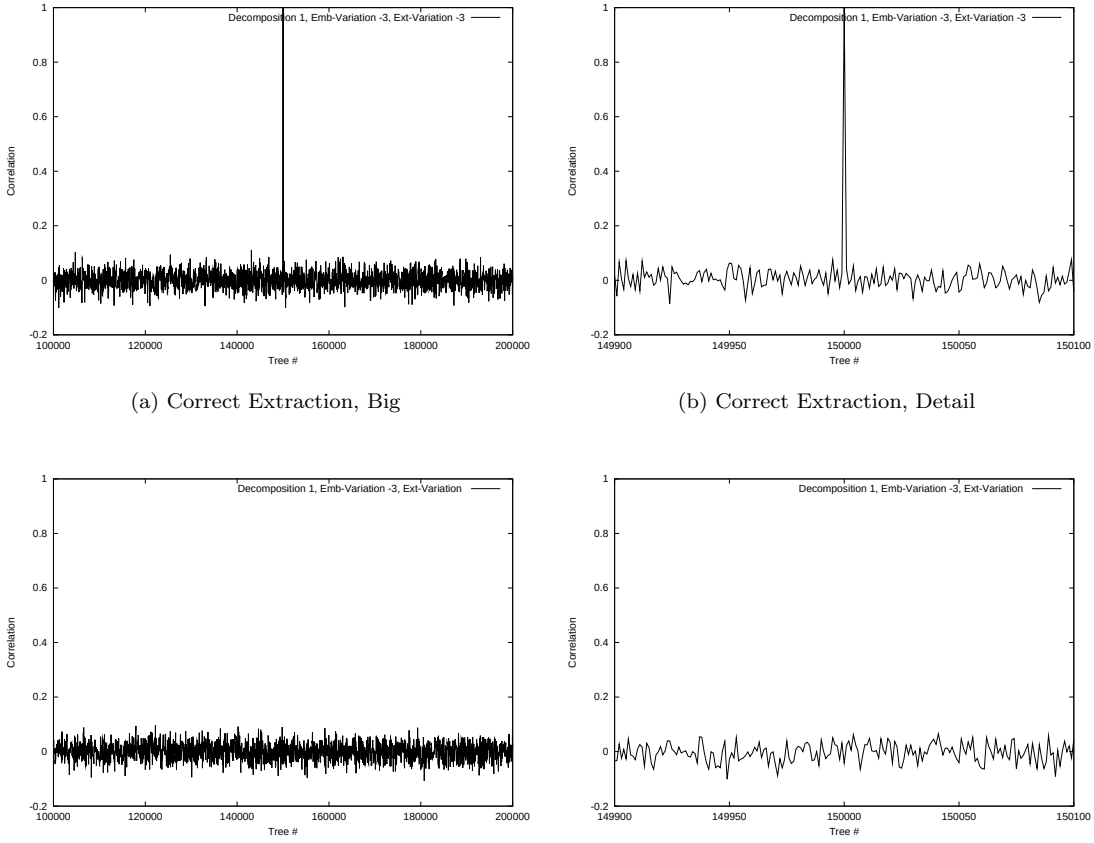


(d) Incorrect Extraction, Detail

Figure 3.6: Decomposition 1, 7 levels, watermark length 1000, variation 2

Decomposition 1 — Variation 3

For variation 3 we shuffle the watermark in a tree-dependent way before we embed it into the image. Figure 3.7 shows the behavior of this system. There is one clear peak with the correct tree number and no correlation if we use the wrong extraction variation.



(a) Correct Extraction, Big

(b) Correct Extraction, Detail

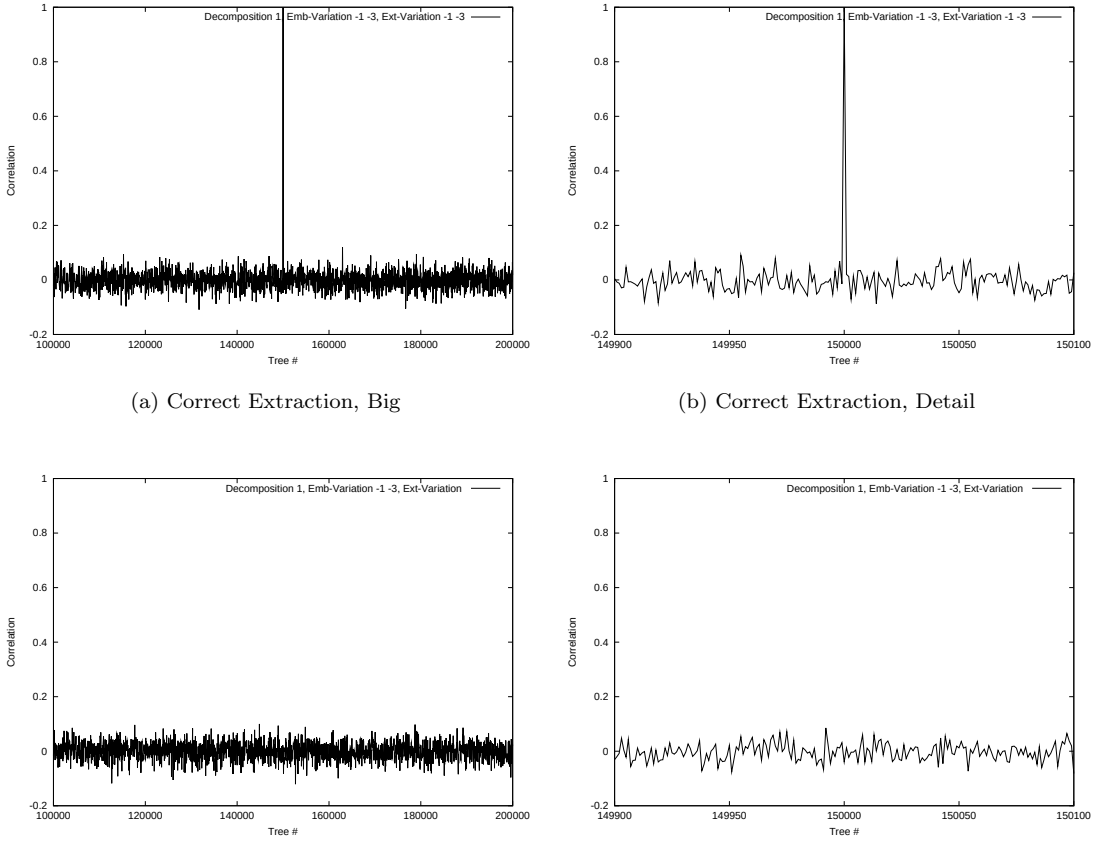
(c) Incorrect Extraction, Big

(d) Incorrect Extraction, Detail

Figure 3.7: Decomposition 1, 7 levels, watermark length 1000, variation 3

Decomposition 1 — Variations 1 and 3

Finally we have a look at combining variations 1 and 3. This means that we shuffle the watermark in a tree-dependent way and then skip some of the wavelet coefficients.
This system also shows very good results. There is only one peak at 150000 and low correlation everywhere else. If we do not use the correct variation for extraction, then we do not detect the watermark at all.



(a) Correct Extraction, Big

(b) Correct Extraction, Detail

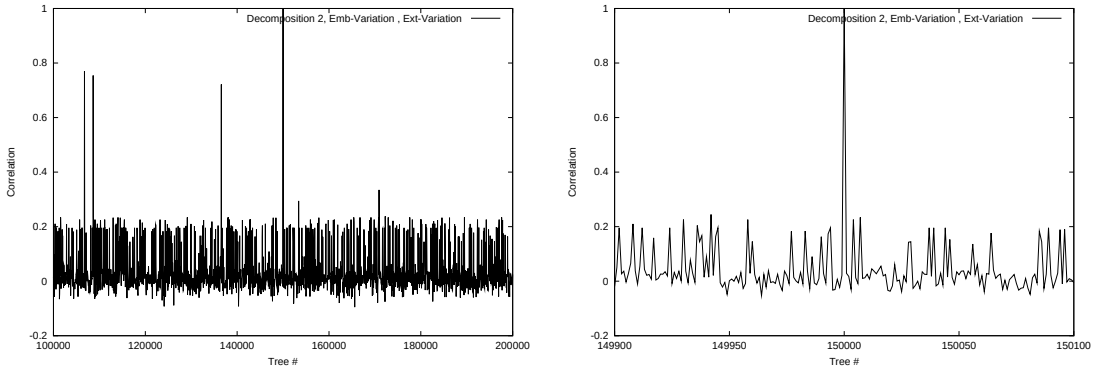
(c) Incorrect Extraction, Big

(d) Incorrect Extraction, Detail

Figure 3.8: Decomposition 1, 7 levels, watermark length 1000, variation 1 & 3

Decomposition 2 — No Variation

The results for decomposition 2 without variation are shown in figure 3.9. We see that in (a) there is one peak at 150000, but that there are also three other tree numbers with more than 60 percent correlation. There are also many other tree numbers with a correlation of around 0.20.
Because for decomposition 2 we do not allow decompositions at the top level and also have a different probability distribution at the lower levels we have more common subtrees than in decomposition 1. This leads to more common sequences in different trees and therefore to higher correlation for the wrong extraction parameters. If two trees are very similar this will lead to the high correlation we see at the three additional peaks in figure (a).
This result was the reason why we introduced the three embedding variations. Different coefficients should be modified or the same coefficients should be modified in a different way, even if common subtrees happen. We added the three tree-number-dependent embedding variations described earlier to add this feature.

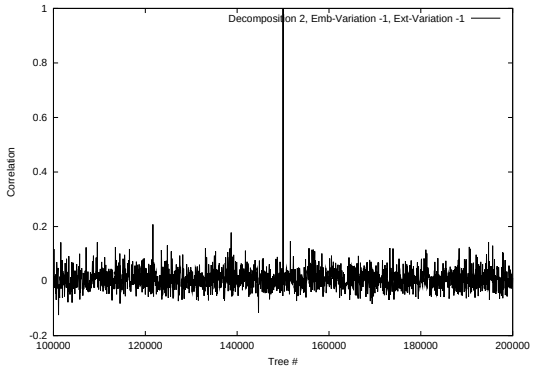


(a) Big (b) Detail

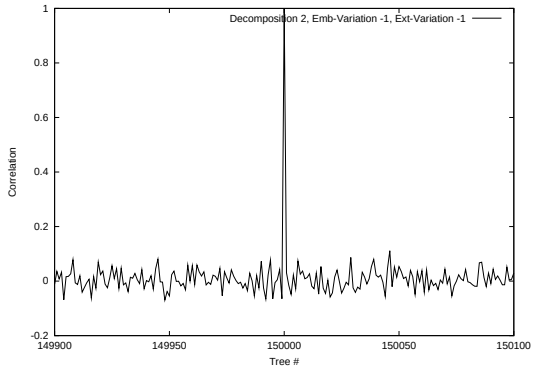
Figure 3.9: Decomposition 2, 7 levels, watermark length 1000, no variation

Decomposition 2 — Variation 1

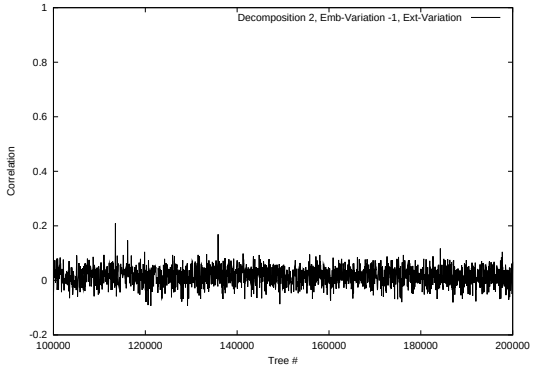
In figure 3.10 we see that there is only one peak for the correct tree number and that the correlation for the wrong trees is reduced. The introduction of the embedding variations makes decomposition 2 a useable system.
Figures (c) and (d) show the results when the wrong extraction variation is used. Again there is very low correlation for all tree numbers and the correct tree number can not be found.



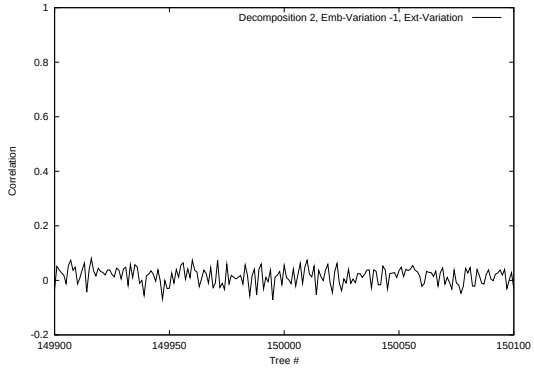
(a) Correct Extraction, Big



(b) Correct Extraction, Detail



(c) Incorrect Extraction, Big

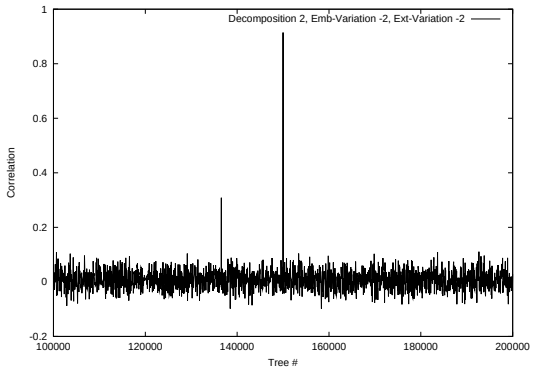


(d) Incorrect Extraction, Detail

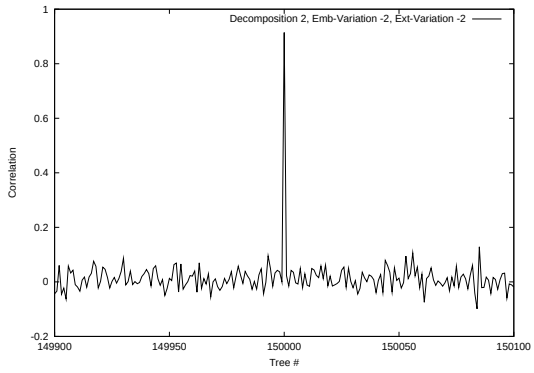
Figure 3.10: Decomposition 2, 7 levels, watermark length 1000, variation 1

Decomposition 2 — Variation 2

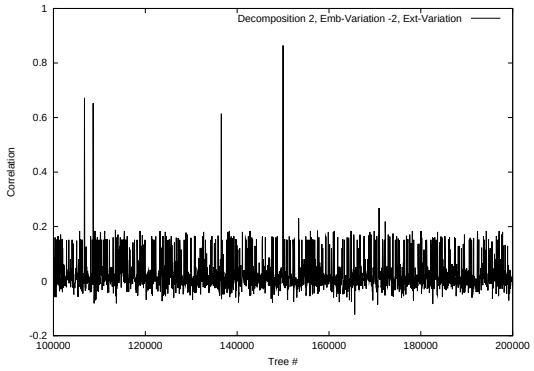
The results for variation 2 are shown in figure 3.11. In figure (a) we see that there is a wrong tree number that has a correlation above 0.20. But what is worse are the results of figures (c) and (d). Even without the correct extraction variation we have high correlation for the correct tree number and also for some other incorrect tree numbers.
Because of these results we decided to not look at variation 2 any further. Variation 2 helps in reducing the correlation of wrong tree numbers, but it is possible to extract the watermark without knowing the variation.



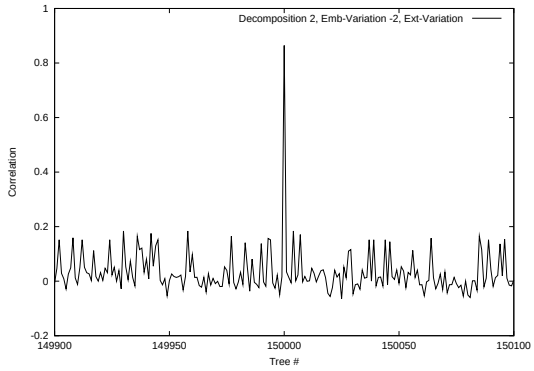
(a) Correct Extraction, Big



(b) Correct Extraction, Detail



(c) Incorrect Extraction, Big

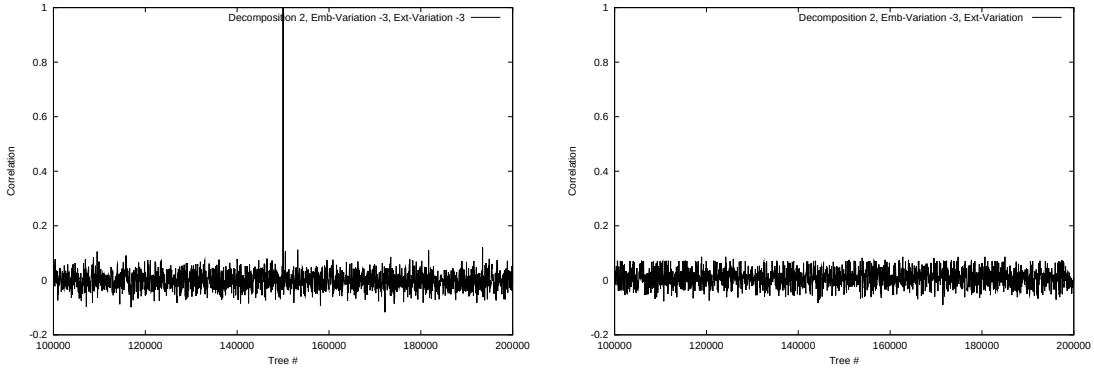


(d) Incorrect Extraction, Detail

Figure 3.11: Decomposition 2, 7 levels, watermark length 1000, variation 2

Decomposition 2 — Variation 3

Variation 3 shows the expected result — one clear peak for tree number 150000 and no high correlation for incorrect tree numbers. Without the correct variation there is no high correlation anywhere.



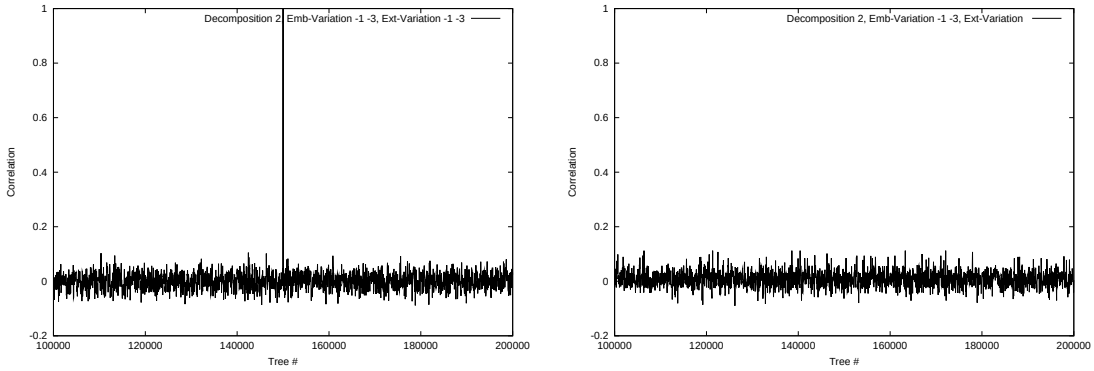
(a) Correct Extraction, Big

(b) Incorrect Extraction, Big

Figure 3.12: Decomposition 2, 7 levels, watermark length 1000, variation 3

Decomposition 2 — Variations 1 and 3

The combination of variations 1 and 3 shows the expected behavior.



(a) Correct Extraction, Big

(b) Incorrect Extraction, Big

Figure 3.13: Decomposition 2, 7 levels, watermark length 1000, variation 1 & 3

3.3.2 4 Levels, Watermark Length 1000

Decomposition 1

If we only use 4 decomposition levels the likeliness of similar subtrees for different tree numbers is higher. If the watermark is short then this can result in very high correlation.
Figure 3.14 shows the results for decomposition 1 without a variation. There are two other tree numbers that resulted in a nearly 1.00 correlation and many other incorrect tree numbers that have very high correlation.
But again the solution is to use one of the embedding variations. Figure 3.15 show decomposition 1 with variations 1 and 3. Now there is only one clear peak and low correlation for the incorrect tree numbers. If we do not use the correct variation for extraction we get no high correlation for all tree numbers.

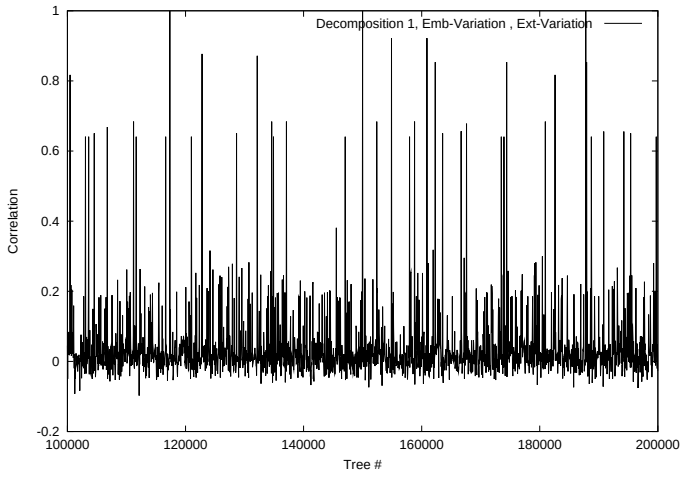
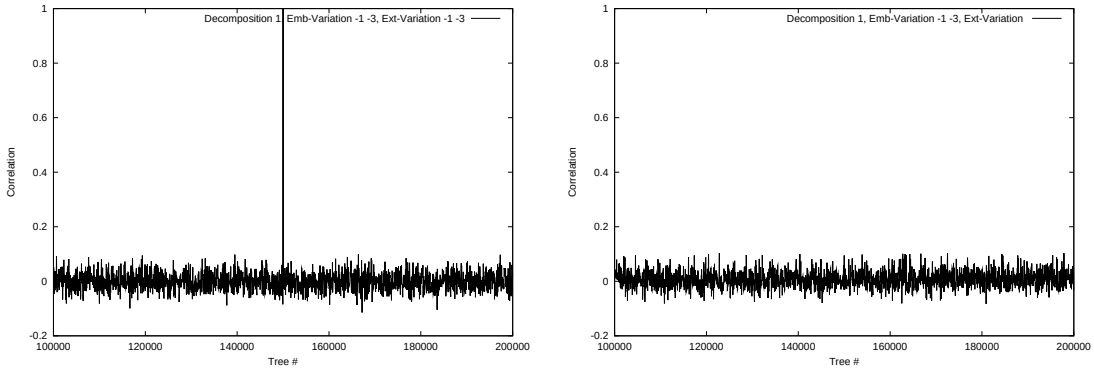


Figure 3.14: Decomposition 1, 4 levels, watermark length 1000, no variation



(a) Correct Extraction

(b) Incorrect Extraction

Figure 3.15: Decomposition 1, 4 levels, watermark length 1000, variation 1 & 3

Decomposition 2

For decomposition 2 we have similar results. Many sub-trees are similar and therefore for many of the tree numbers we get high correlation.
In figure 3.16 we see the result when no embedding variation is used. There is only one clear peak, but for many incorrect tree numbers there is a correlation of around 0.20. The reason why the correlation is close to 0.20 for so many trees is not clear at the moment and may need further study in the future.
The results with variations 1 and 3 are shown in figure 3.17. As expected we have only one clear peak for the correct tree number. In figure (b) no variation is used for extraction and therefore we have no peak in the correlation results.

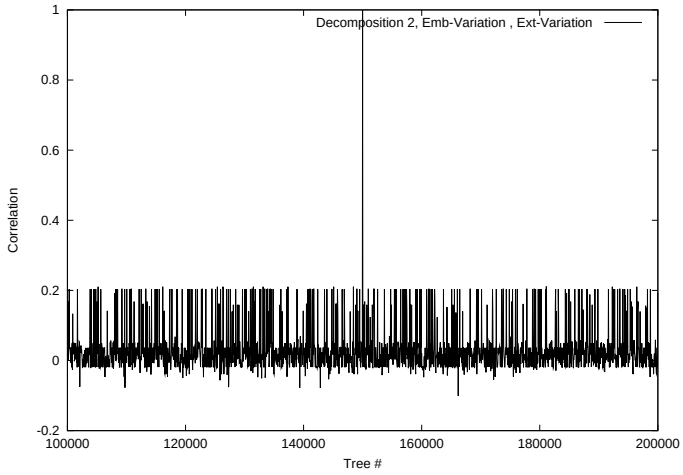
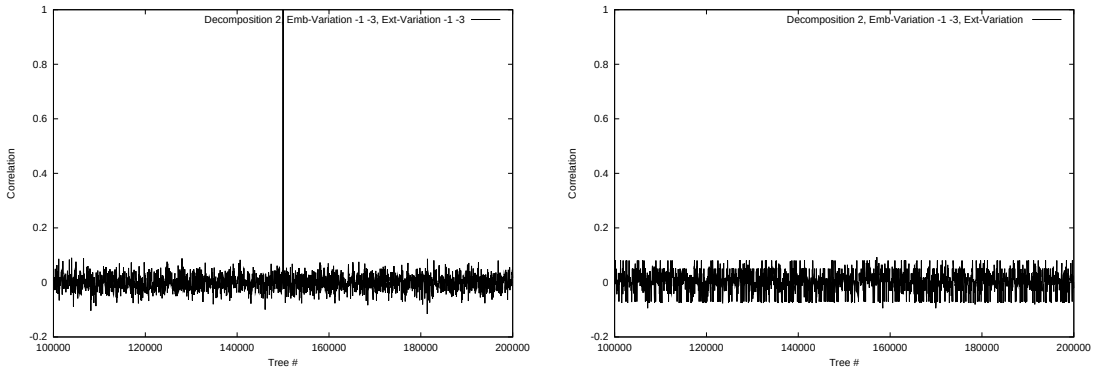


Figure 3.16: Decomposition 2, 4 levels, watermark length 1000, no variation



(a) Correct Extraction

(b) Incorrect Extraction

Figure 3.17: Decomposition 2, 4 levels, watermark length 1000, variation 1 & 3

3.3.3 Watermark Length 5000 — Decomposition 2

7 Levels

Figure 3.18 shows the security measurement for decomposition 2 with a 7 level decomposition and a 5000 element watermark. There is only one clear peak and only three other tree decompositions have higher correlation.
In figure 3.19 we see that using variations 1 and 3 results in a system with one clear peak and no correlation when used without the correct variation.

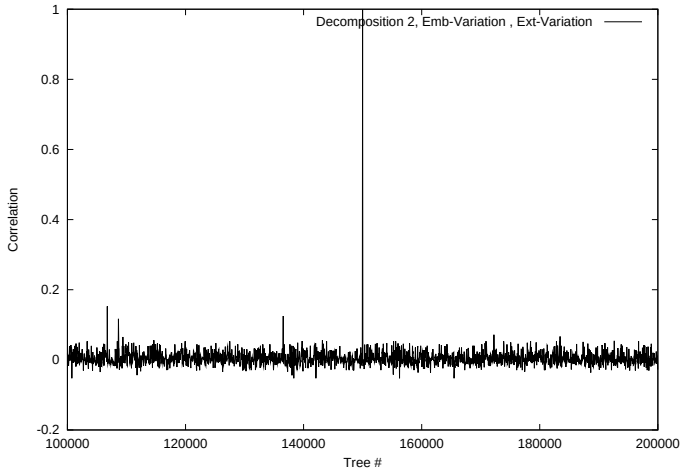
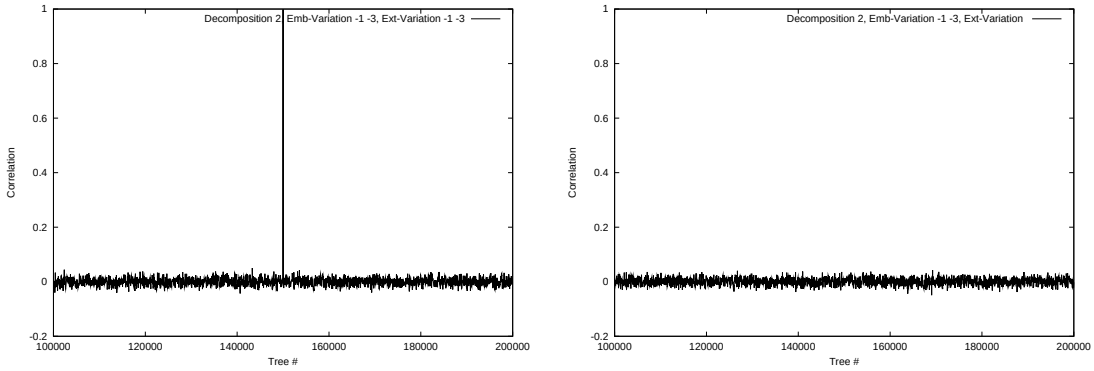


Figure 3.18: Decomposition 2, 7 levels, watermark length 5000, no variation



(a) Correct Extraction

(b) Incorrect Extraction

Figure 3.19: Decomposition 2, 7 levels, watermark length 5000, variation 1 & 3

4 Levels

The results for a 4 level decomposition are shown in figures 3.20 and 3.21. With the longer watermark sequence the influence of common subtrees seems to be lessened. There is only one clear peak, even without embedding variation. The results with variations 1 and 3, skipping coefficients and shuffling the watermark sequence, are shown in figure 3.21. The correlation for the wrong tree numbers is further reduced and without the correct variation the extraction is not possible.

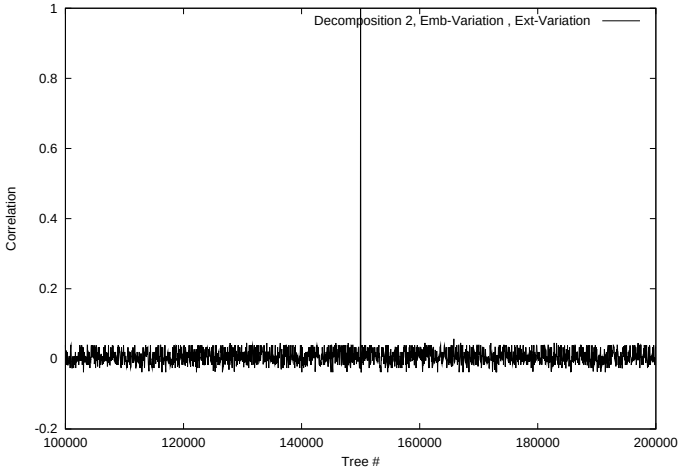
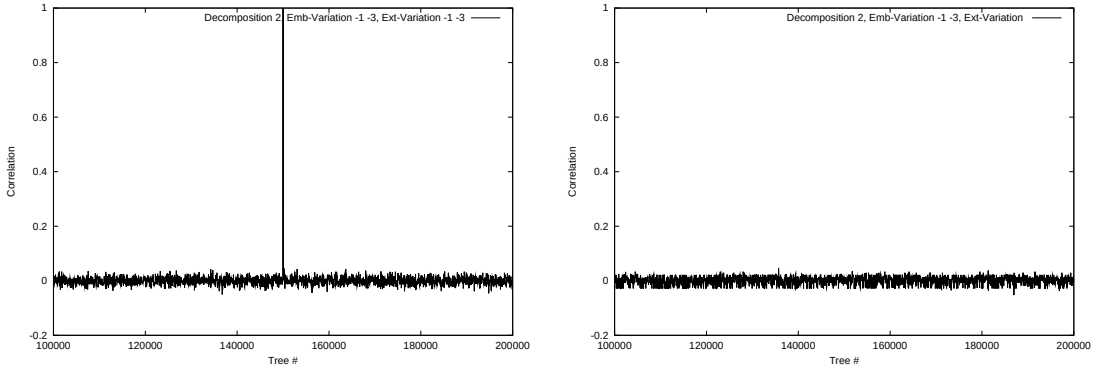


Figure 3.20: Decomposition 2, 4 levels, watermark length 5000, no variation



(a) Correct Extraction

(b) Incorrect Extraction

Figure 3.21: Decomposition 2, 4 levels, watermark length 5000, variation 1 & 3

3.3.4 Watermark Length 20000 — Decomposition 2

7 Levels

In figure 3.22 we see that beside the correct peak at tree number 150000 there are 4 other, but much smaller, peaks. As can be seen in figure 3.23 using variations 1 and 3 removes those additional peaks. The embedding variation reduces the correlation for the wrong tree numbers and the extraction is only successful with the correct variation.

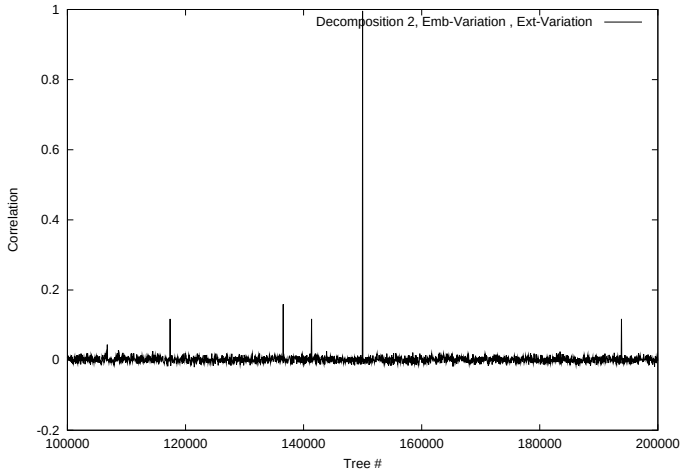
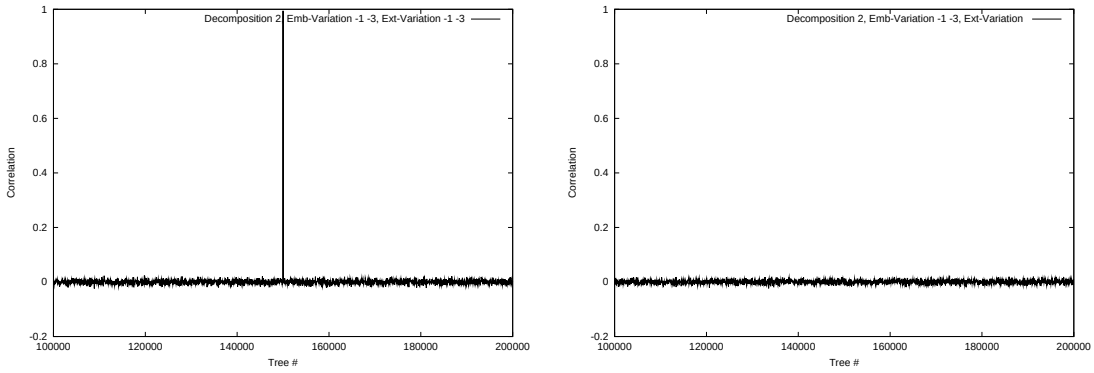


Figure 3.22: Decomposition 2, 7 levels, watermark length 20000, no variation



(a) Correct Extraction

(b) Incorrect Extraction

Figure 3.23: Decomposition 2, 7 levels, watermark length 20000, variation 1 & 3

4 Levels

Figures 3.24 and 3.25 show the results for 4 levels. We have one peak at 150000 and no high correlation otherwise. By using variations 1 and 3 we lower the average correlation a little bit and ensure that the watermark can only be extracted with the correct variation.

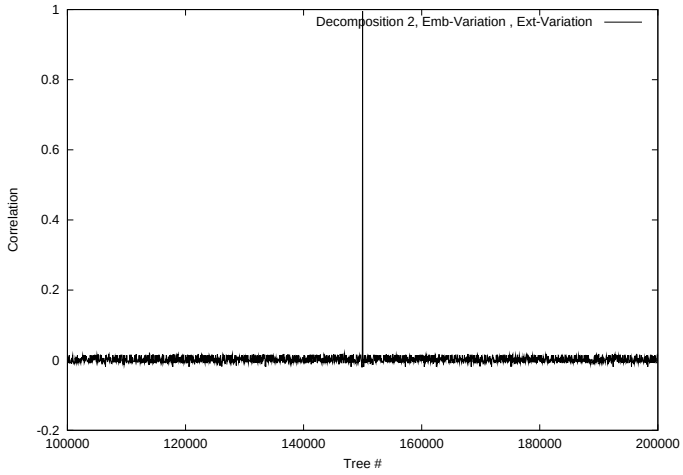
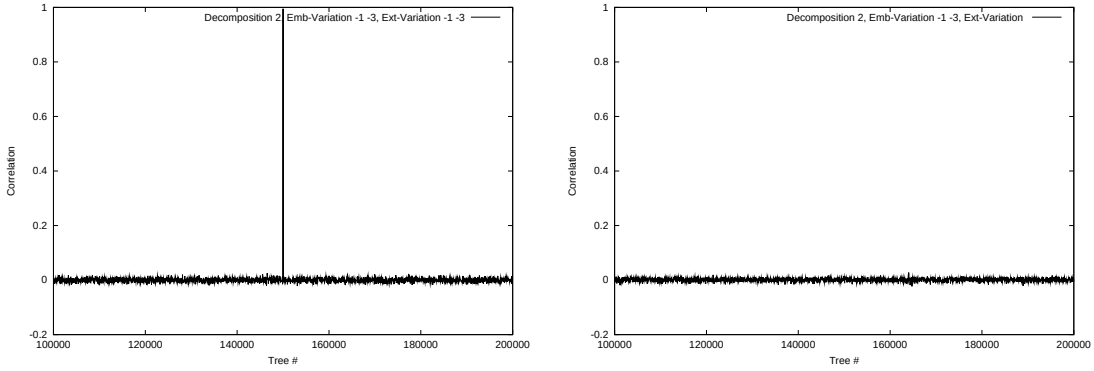


Figure 3.24: Decomposition 2, 4 levels, watermark length 20000, no variation



(a) Correct Extraction

(b) Incorrect Extraction

Figure 3.25: Decomposition 2, 4 levels, watermark length 20000, variation 1 & 3

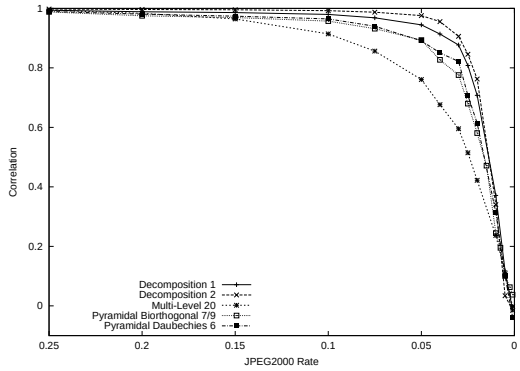
3.4 Quality Assessment

For the quality assessment we embed a watermark with 40dB PSNR and then compress the watermarked image with JPEG and JPEG2000. We then try to detect the watermark in the compressed image and measure the correlation. As a measure of the distortion we use the PSNR of the compressed image.
To get a good variation of different trees we use tree numbers from 100000 to 200000 with increments of 400. From all those 251 measurements we calculate the average, maximum and minimum correlation and PSNR.
We will compare tree decomposition 1 and 2. We expect that tree decomposition 2 will have better results for higher compression rates.
With more subband decompositions we expect that it will be possible to embed longer watermark sequences compared to the pyramidal decomposition. To see whether this is true we analyze the image quality with watermark lengths 1000, 5000 and 20000.
And we will again use either 4 or 7 decomposition levels to measure the influence of the tree depth on the watermark quality under compression.
The two standard images “Lena” and “Barbara” will be used to analyze the influence of the host image. See appendix A for the original images and for the effects of JPEG and JPEG2000 compression.
We compare the results of the wavelet packet methods with the standard Wang watermarking system using the Daubechies 6 and the Biorthogonal 7/9 filters and the parametrized system Multi-Level 20 that uses 20 parameters, 5 parameters per filter and 4 different filters for four decomposition levels. For the parametrized system we took 251 random parametrizations to get a good variation of the possible filters.
We will show the most detailed results for 7 decomposition levels and a watermark of length 1000. For the other systems we will not show the detailed results and only show the behavior in comparison with other systems.

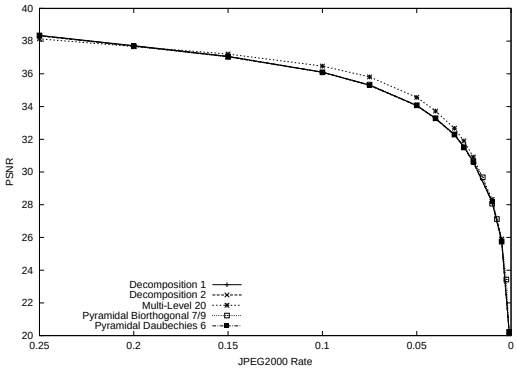
3.4.1 7 Levels, Watermark Length 1000

Lena

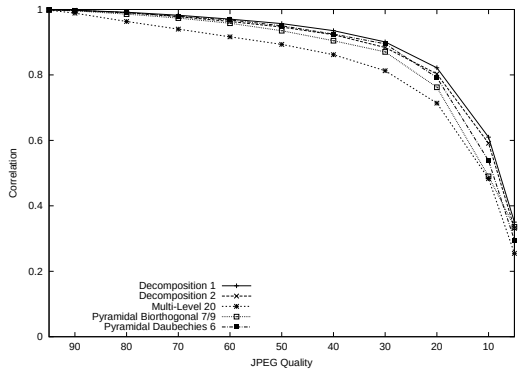
Figure 3.26 shows the compression behavior of the two different wavelet packet decompositions in comparison with the Multi-Level 20 parametrized filter system and the standard system with the Biorthogonal 7/9 and the Daubechies 6 filters.
In figure (a) we see the correlation behavior under JPEG2000 compression. The performance of decomposition 2 is superior to all other systems. The wavelet packet system with decomposition 1 is also better than the two standard systems and the parametrized system shows the worst performance in this comparison.
Figure 3.26(c) shows the JPEG compression results. In this case the difference is smaller than under JPEG2000 compression. The decomposition 1 system performs slightly better than decomposition 2 and both wavelet packet systems are above the standard decompositions.
The results for the PSNR performance are shown in diagrams (b) and (d). All systems behave very similarly, just the parametrized system shows slightly better results for some compression rates.



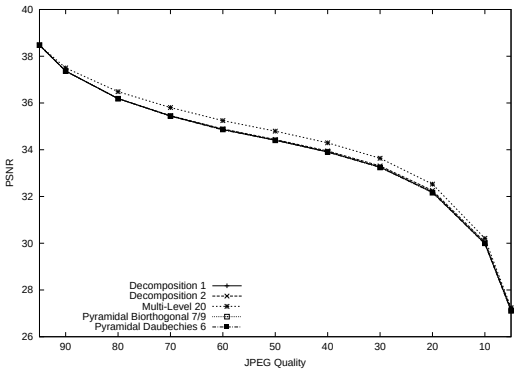
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



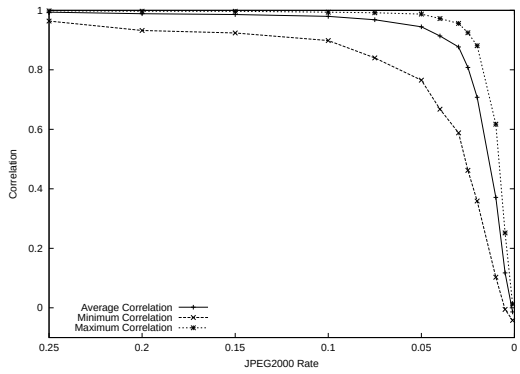
(c) JPEG Correlation



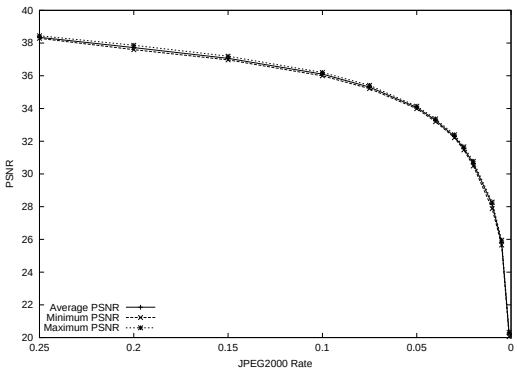
(d) JPEG PSNR

Figure 3.26: Lena: 7 levels, watermark length 1000, comparison of methods

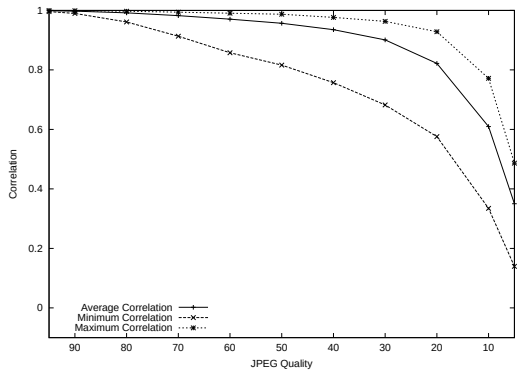
Figure 3.27 gives a close look at the results for decomposition 1. This diagram shows the average, minimum and maximum behavior. The minimum correlation behavior is still very good and the average is closer to the maximum. For the PSNR behavior there is little difference between the minimum and the maximum.



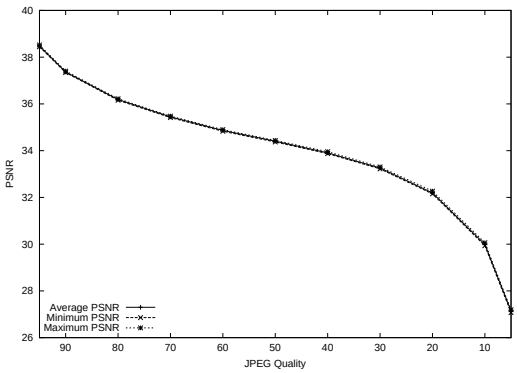
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



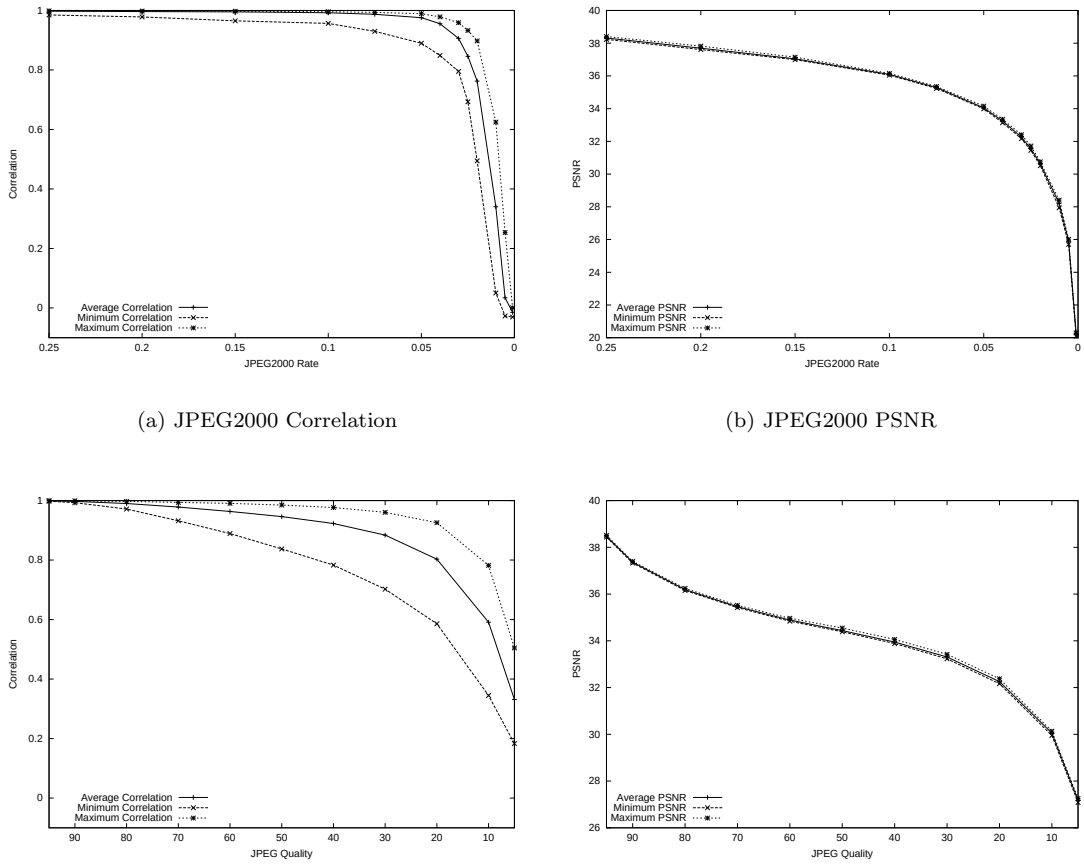
(c) JPEG Correlation



(d) JPEG PSNR

Figure 3.27: Lena: decomposition 1, 7 levels, watermark length 1000, no variation

In comparison figure 3.28 shows the detailed results for decomposition 2. The minimum correlation under JPEG2000 compression is higher for decomposition 2. The results for JPEG compression show little difference to the results for decomposition 1. Again the PSNR behavior is nearly the same for all decompositions.



(a) JPEG2000 Correlation

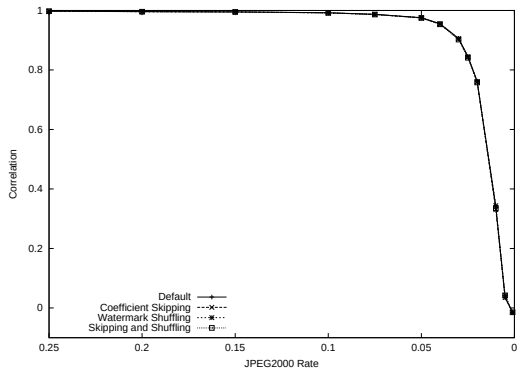
(b) JPEG2000 PSNR

(c) JPEG Correlation

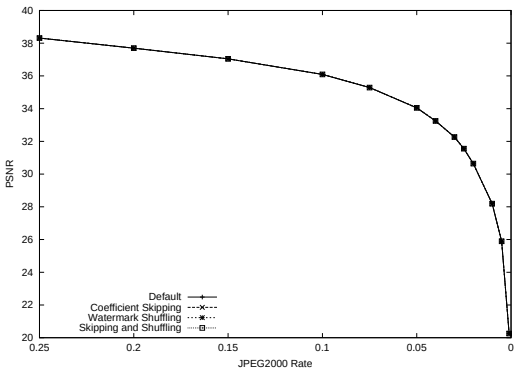
(d) JPEG PSNR

Figure 3.28: Lena: decomposition 2, 7 levels, watermark length 1000, no variation

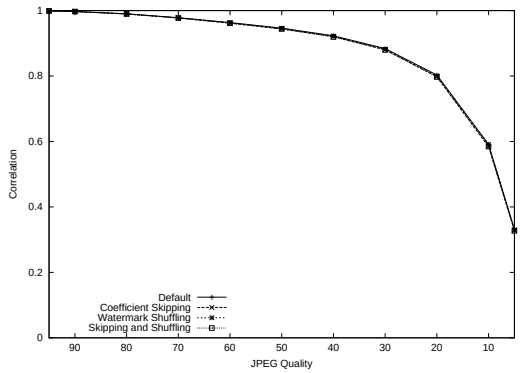
Finally figure 3.29 compares the performance of the different embedding variations for decomposition 2. As you can see in all 4 diagrams there is nearly no influence of the embedding variation on the compression behavior. For decomposition 1 there is also no difference between the embedding variations, but we do not show the results here.



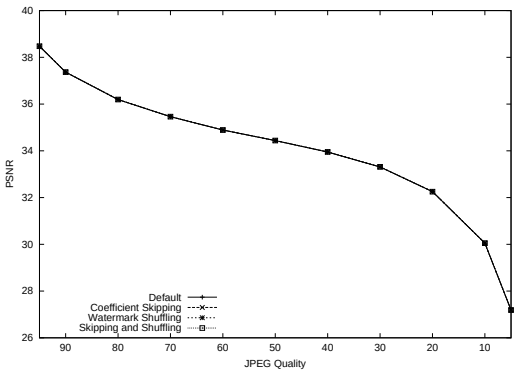
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

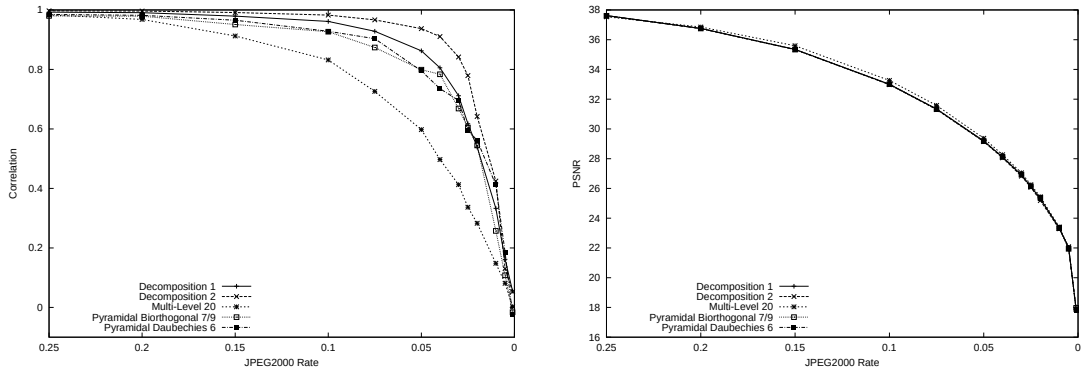


(d) JPEG PSNR

Figure 3.29: Lena: decomposition 2, 7 levels, watermark length 1000, comparison of variations

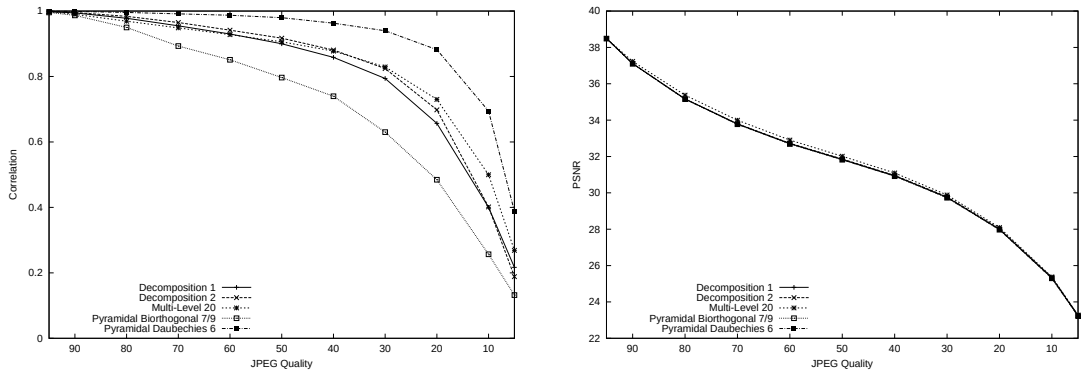
Barbara

Figure 3.30 compares the different watermarking methods when "Barbara" is used as host image. In (a) there is a larger advantage for decomposition 2 this time, otherwise the results are comparable to the results for "Lena".
Figure (c) shows the behavior under JPEG compression. The two standard systems show very different results for this image. Using the Biorthogonal 7/9 filter produces the worst results, while using the Daubechies 6 filter for embedding the watermark produces the best results under JPEG compression. The three proposed systems are in the middle of those two systems, with the parametrized system being slightly better than the wavelet packet systems.
Again the results for PSNR behavior do not show a significant difference between the systems.



(a) JPEG2000 Correlation

(b) JPEG2000 PSNR

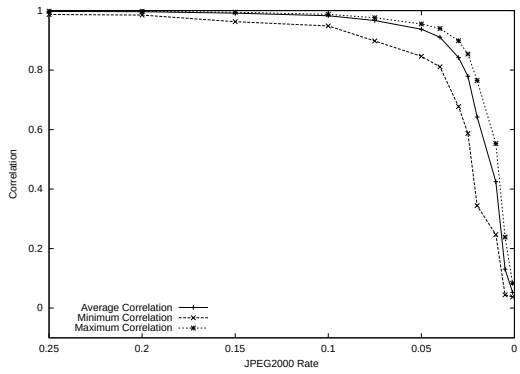


(c) JPEG Correlation

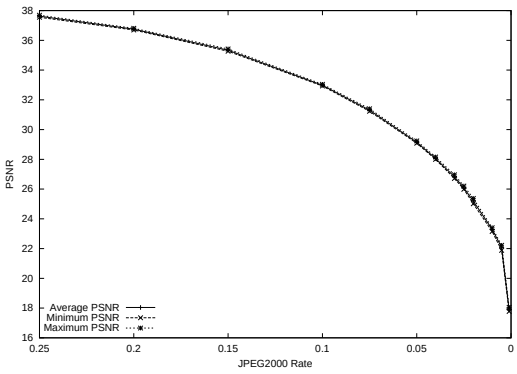
(d) JPEG PSNR

Figure 3.30: Barbara: 7 levels, watermark length 1000, comparison of methods

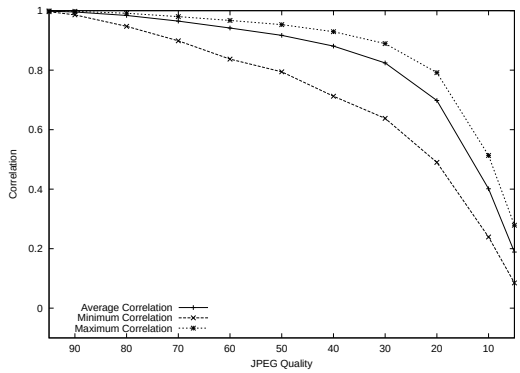
Figure 3.31 shows the detailed behavior of decomposition 2. The average result is closer to the maximum, which means that there are only a few tree decompositions that show the minimum behavior. But even the worst behavior is still very reasonable.



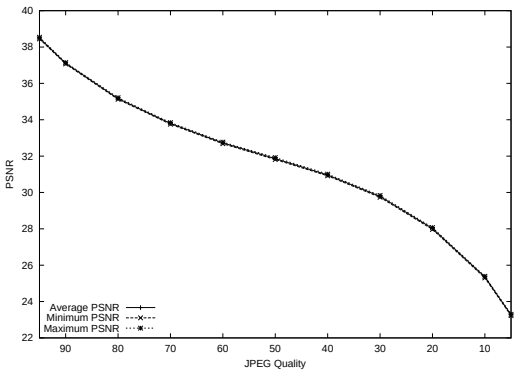
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

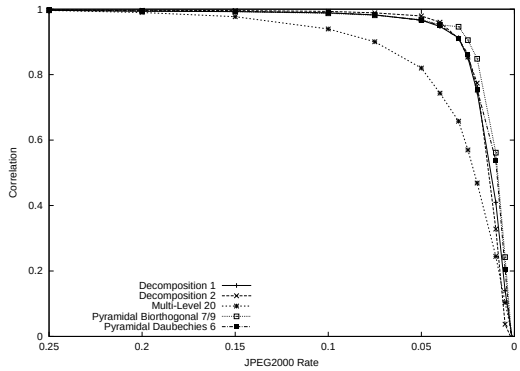


(d) JPEG PSNR

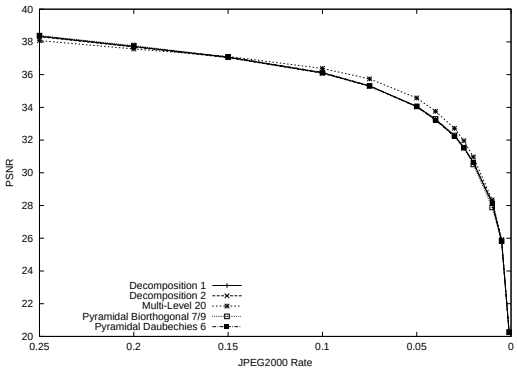
Figure 3.31: Barbara: decomposition 2, 7 levels, watermark length 1000, no variation

3.4.2 4 Levels, Watermark Length 1000

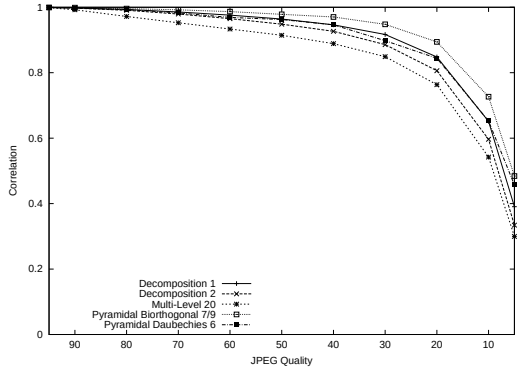
The behavior is not very different, if only 4 decomposition levels are used. The comparison of the different systems is shown in figure 3.32. Under JPEG2000 compression all the systems behave nearly identical, only the parametrized system shows lower correlation results. For JPEG compression there is a larger difference between the systems. The standard filters work best, followed by the two tree decompositions. Again the parametrized system is a little bit below the other systems. The PSNR behavior shows a little advantage for the parametrized system, but is otherwise very close together.



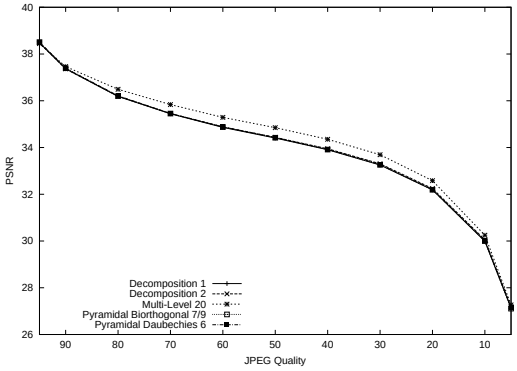
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



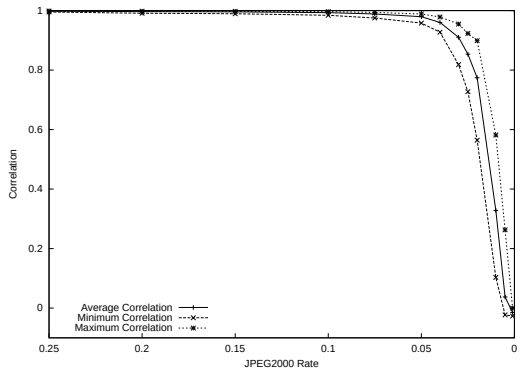
(c) JPEG Correlation



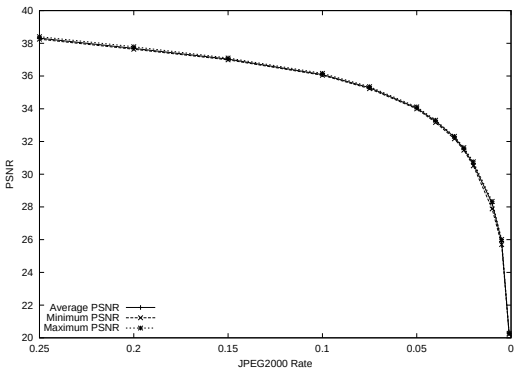
(d) JPEG PSNR

Figure 3.32: Lena: 4 levels, watermark length 1000, comparison of methods

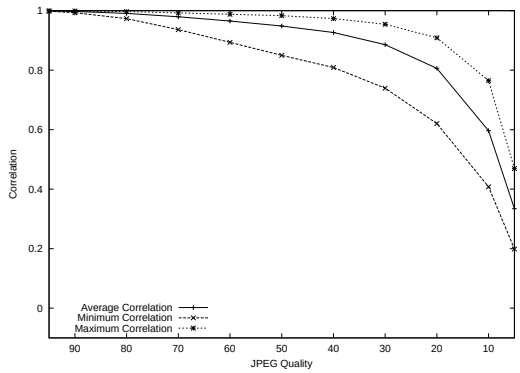
The minimum, maximum and average of decomposition strategy 2 is shown in figure 3.33. The variation between the different tree decompositions is very low and even the minimum behavior is still good.



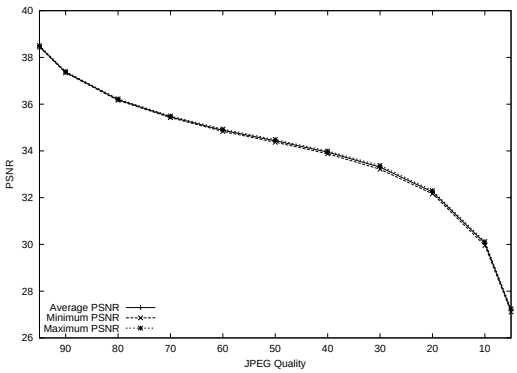
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

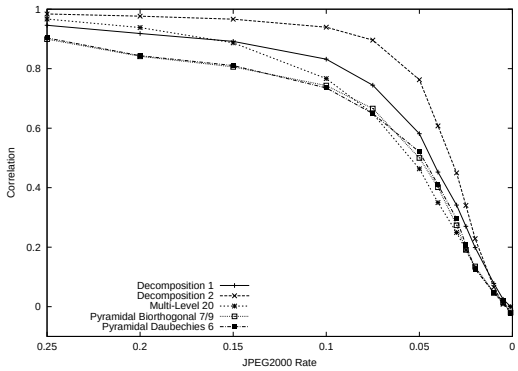


(d) JPEG PSNR

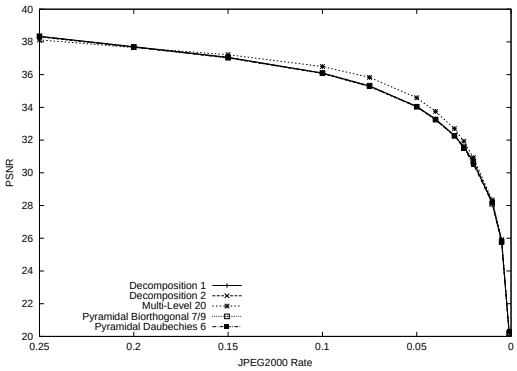
Figure 3.33: Lena: decomposition 2, 4 levels, watermark length 1000, no variation

3.4.3 7 Levels, Watermark Length 5000

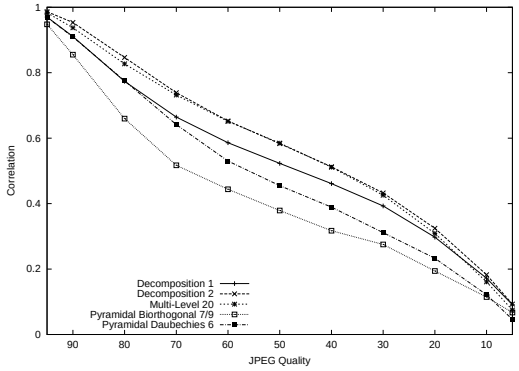
Figure 3.34 shows the results when a 5000 element watermark is embedded into a 7 level tree. In (a) we clearly see the advantage of decomposition 2. At a JPEG2000 compression rate of 0.05 the average correlation with decomposition 2 is 0.76 while for decomposition 1 the correlation is only 0.58, nearly 20 percent lower.
The results for JPEG compression are shown in figure (c). This time decomposition 2 is better than decomposition 1 even for JPEG compression. Also the parametrized system shows better results than the decomposition 1 wavelet packet system.



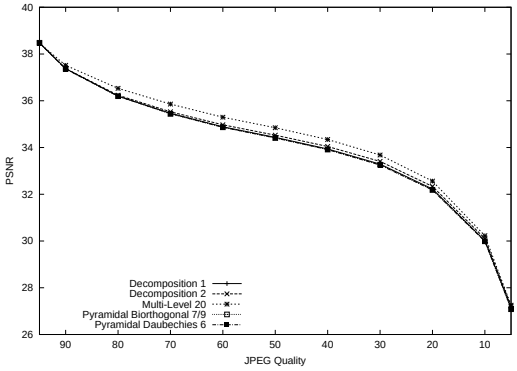
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



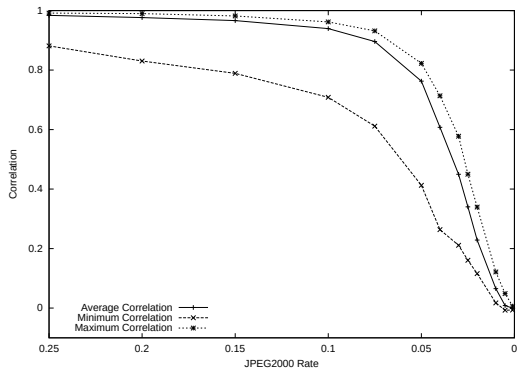
(c) JPEG Correlation



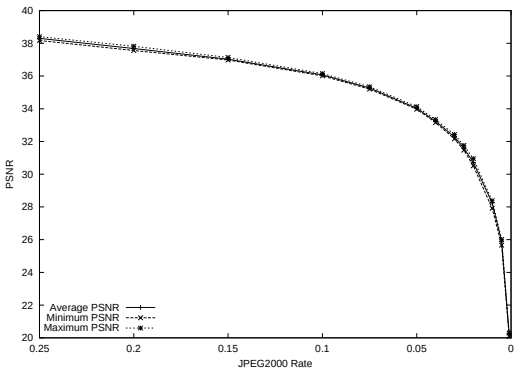
(d) JPEG PSNR

Figure 3.34: Lena: 7 levels, watermark length 5000, comparison of methods

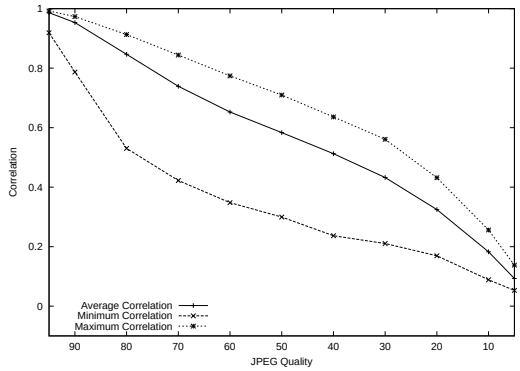
The detailed results for decomposition 2 are shown in figure 3.35. We can see that the average result is very good and close to the best performing results. But the minimum behavior is dropping significantly lower with 5000 watermark elements.



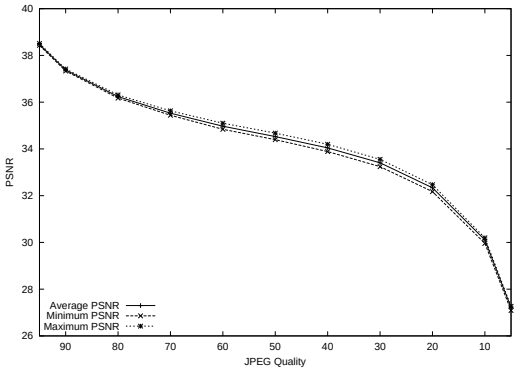
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



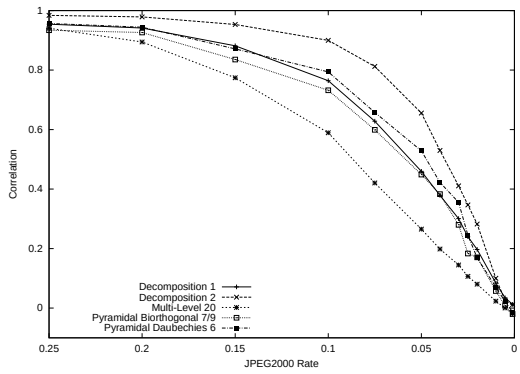
(c) JPEG Correlation



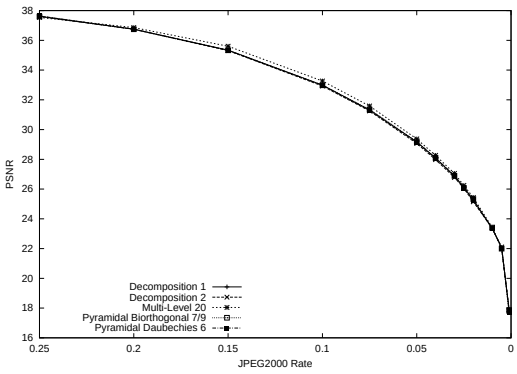
(d) JPEG PSNR

Figure 3.35: Lena: decomposition 2, 7 levels, watermark length 5000, no variation

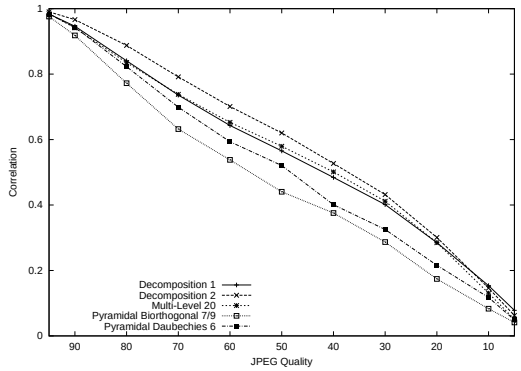
The results for the cover image "Barbara" are shown in figure 3.36 and show comparable behavior. For both JPEG2000 and JPEG compression decomposition 2 shows the best results. For JPEG compression all three proposed systems show results that are better than the two standard systems.



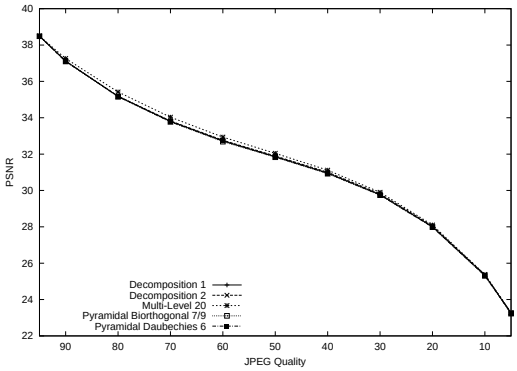
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

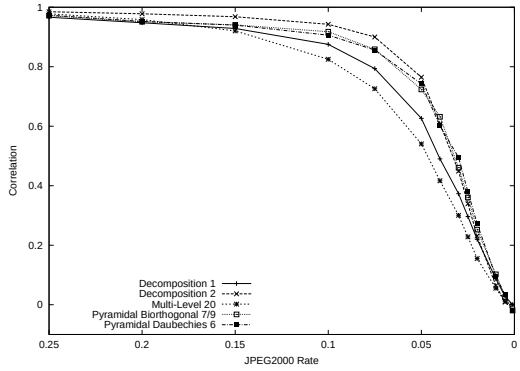


(d) JPEG PSNR

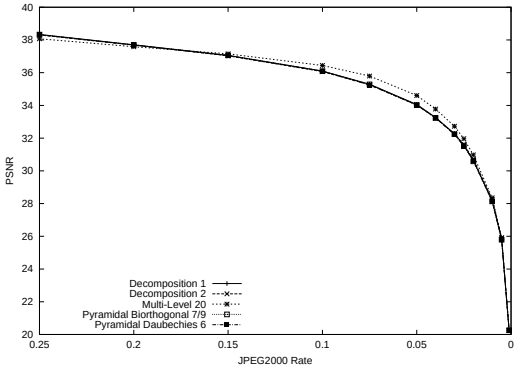
Figure 3.36: Barbara: 7 levels, watermark length 5000, comparison of methods

3.4.4 4 Levels, Watermark Length 5000

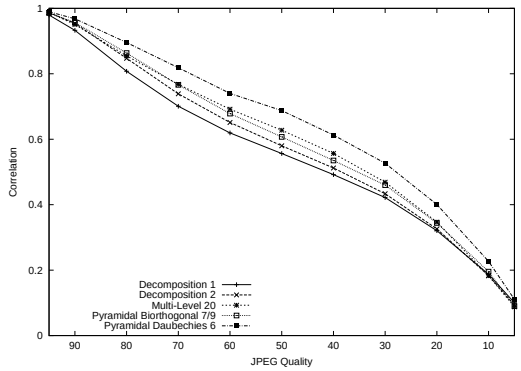
For JPEG2000 decomposition 2 is again the best system. Decomposition 1 and the Multi-Level 20 system are below the two standard systems. For JPEG compression both wavelet packet systems show the worst performance and are below the standard systems and the parametrized system. The reason why the performance of the wavelet packet systems show this behavior is unclear at the moment.



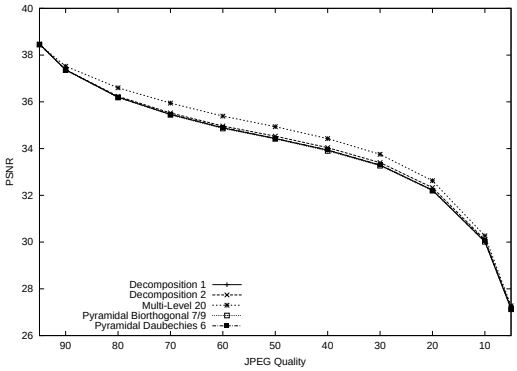
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

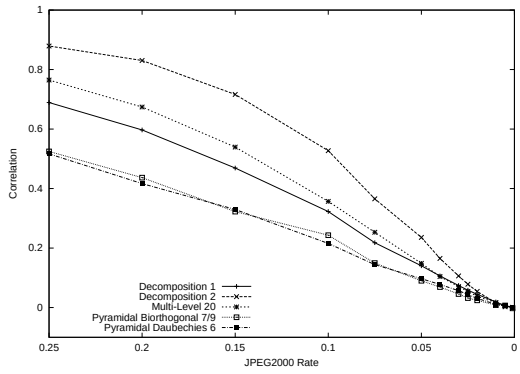


(d) JPEG PSNR

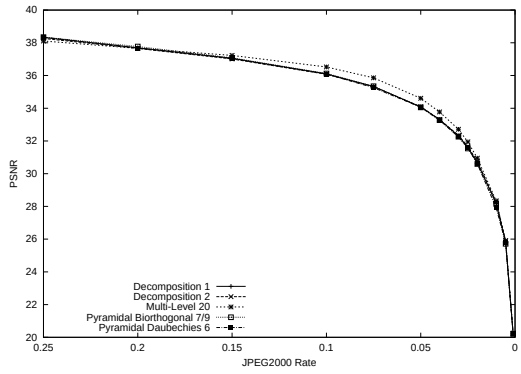
Figure 3.37: Lena: 4 levels, watermark length 5000, comparison of methods

3.4.5 7 Levels, Watermark Length 20000

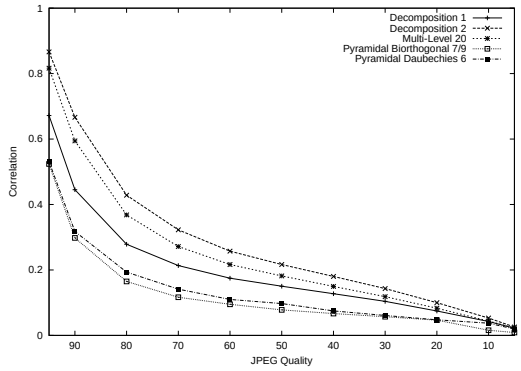
The results for this system are shown in figure 3.38. Here we see a clear advantage for decomposition 2.
At a JPEG2000 compression rate of 0.15 decomposition 2 has an average correlation of 0.71 and the Multi-Level 20 system has a correlation of 0.54. Decomposition 1 has a correlation of only 0.47 at this compression rate. Decomposition 2 has more than 20 percent higher correlation than decomposition 1. All three proposed systems have higher correlation than the two standard systems. Also under JPEG compression decomposition 2 and the parametrized filter system show the best performance and even decomposition 1 has a correlation above the two standard systems.



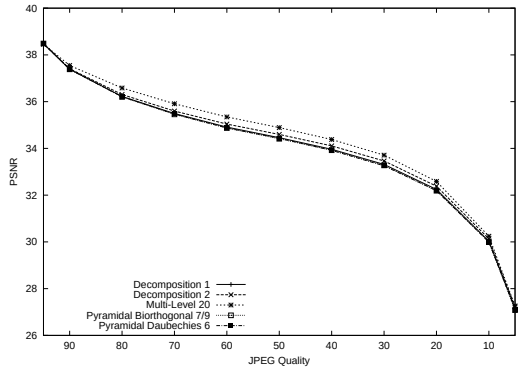
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



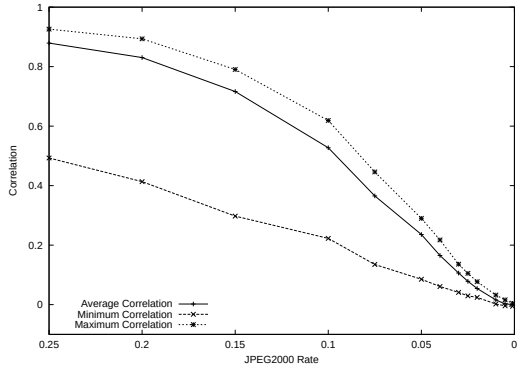
(c) JPEG Correlation



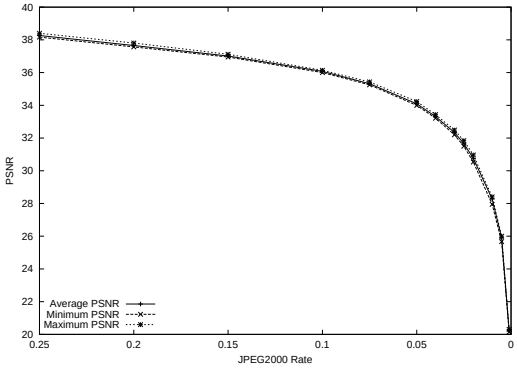
(d) JPEG PSNR

Figure 3.38: Lena: 7 levels, watermark length 20000, comparison of methods

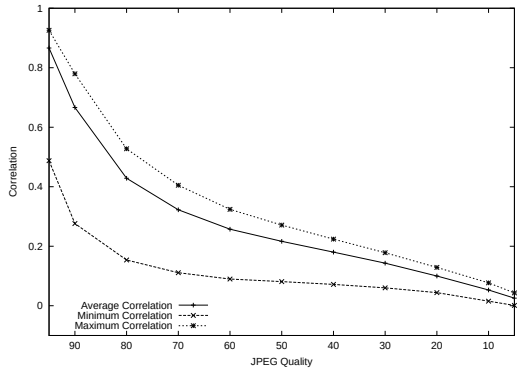
The detailed results are shown in figure 3.39.
Again the average is a lot closer to the maximum, which means that there are only a few tree decompositions that show bad behavior.
The line of the worst behaving system closely matches the standard decomposition systems with the Biorthogonal 7/9 filter. Our interpretation of this is that for some of the trees the coefficients selected for embedding closely match the coefficients that are used for the standard pyramidal decomposition and therefore show comparable results.



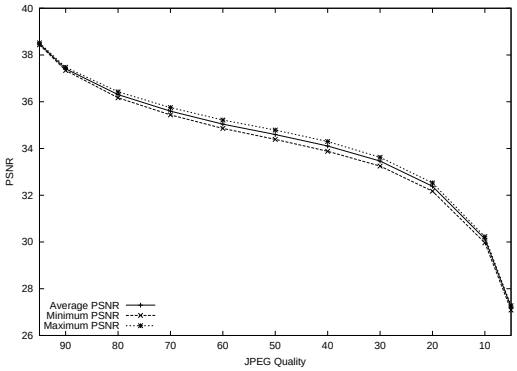
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



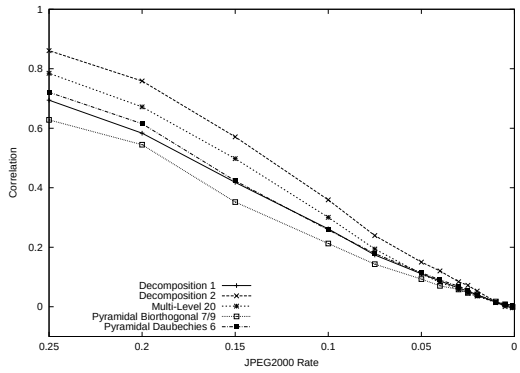
(c) JPEG Correlation



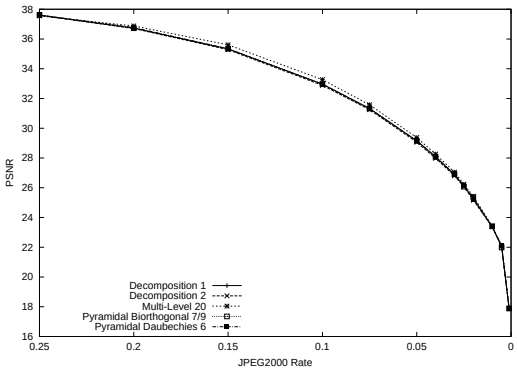
(d) JPEG PSNR

Figure 3.39: Lena: decomposition 2, 7 levels, watermark length 20000, no variation

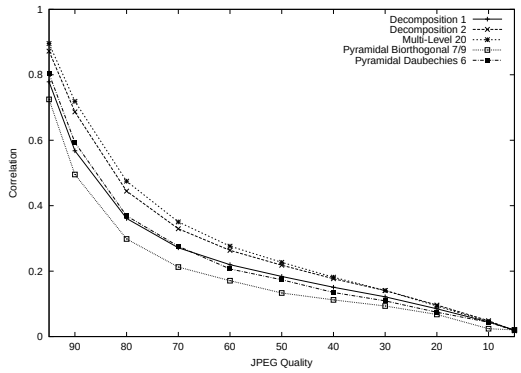
Figure 3.40 shows the results, when using "Barbara" as host image.
Under JPEG2000 compression decomposition 2 and the parametrized filter system are the clear winners again.
For JPEG compression the parametrized system shows slightly better performance than the wavelet packet system with decomposition 2. Both system are again better than decomposition 1 and the two standard systems.



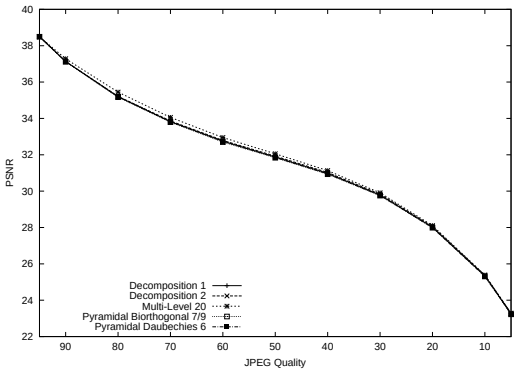
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

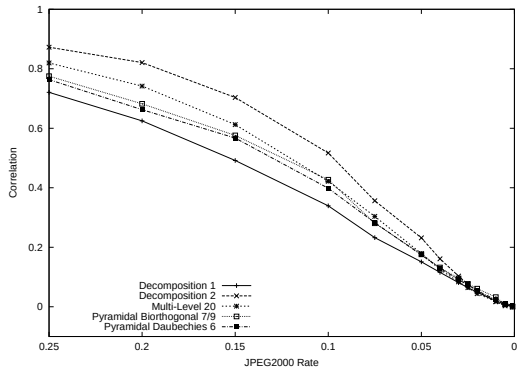


(d) JPEG PSNR

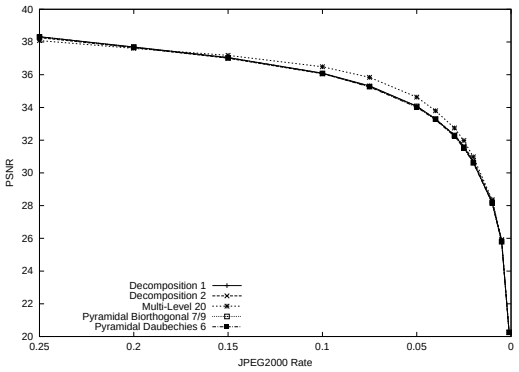
Figure 3.40: Barbara: 7 levels, watermark length 20000, comparison of methods

3.4.6 4 Levels, Watermark Length 20000

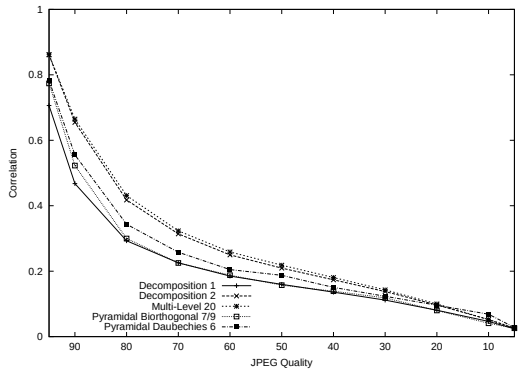
The results when we use a watermark of length 20000 and only 4 decomposition levels are shown in figure 3.41. For JPEG2000 compression decomposition 2 is the clear winner and is above all other systems. For low compression ratios the parametrized system also shows an advantage over the standard systems. Decomposition 1 has the worst results for this parameter setting. Under JPEG compression we have similar results. Decomposition 2 and the parametrized Multi-Level 20 system show the best performance and decomposition 1 is a bit below the two standard systems.



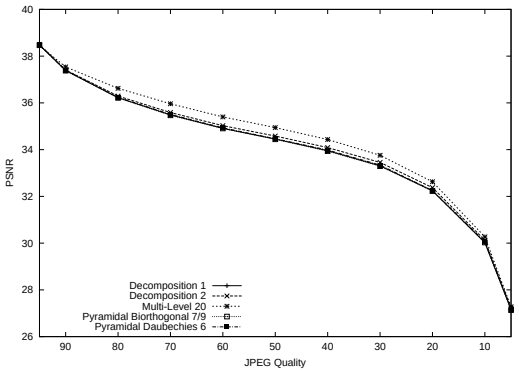
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



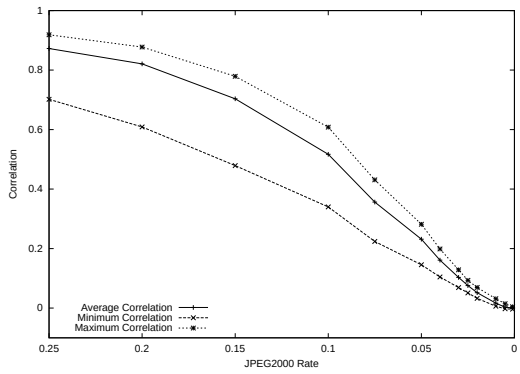
(c) JPEG Correlation



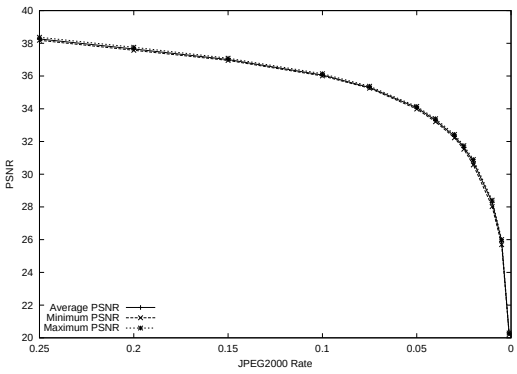
(d) JPEG PSNR

Figure 3.41: Lena: 4 levels, watermark length 20000, comparison of methods

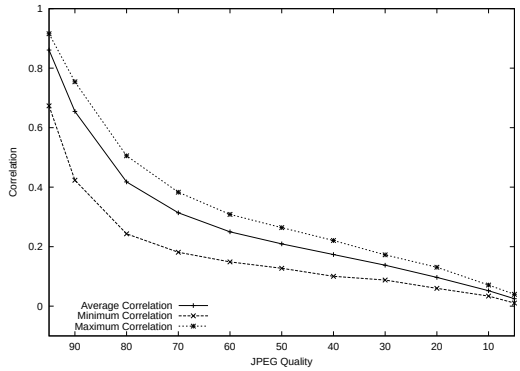
Figure 3.42 shows the detailed results for decomposition 2. The average behavior is again closer to the maximum and even the worst-case correlation results are still good. The results for the PSNR show nearly no influence of the selected tree decomposition.



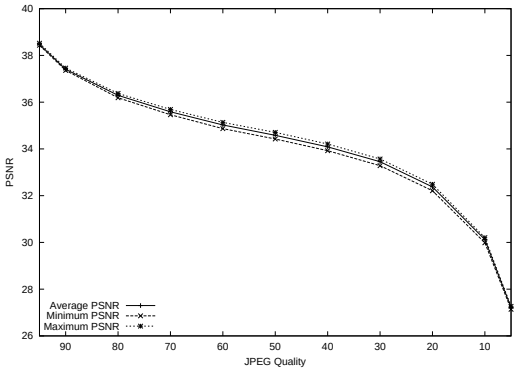
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

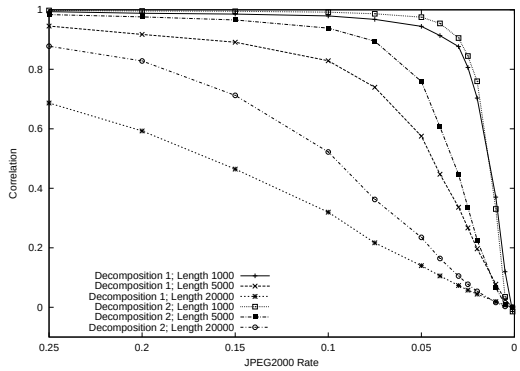


(d) JPEG PSNR

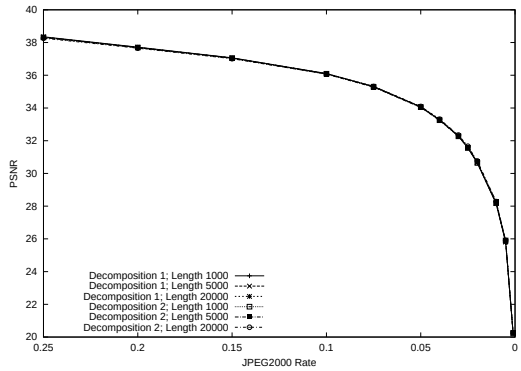
Figure 3.42: Lena: decomposition 2, 4 levels, watermark length 20000, no variation

3.4.7 Length Comparison

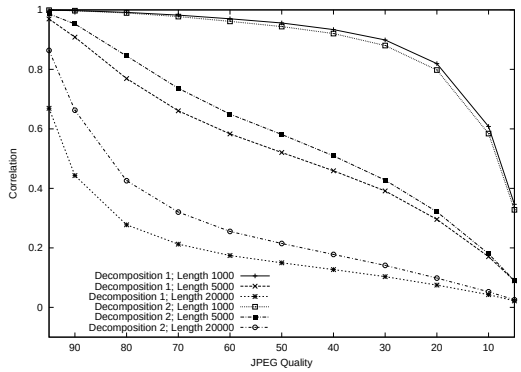
Figure 3.43 shows the performance of decompositions 1 and 2 with 7 decomposition levels for watermarks of length 1000, 5000 and 20000.
In (a) we see that under JPEG2000 compression decomposition 2 is the better system for all watermark lengths. The advantage of decomposition 2 gets bigger for longer watermarks. Figure 3.43(c) shows the results under JPEG compression. For a watermark of length 1000 decomposition 1 has a slight advantage, but for lengths 5000 and 20000 decomposition 2 is clearly the better system.
Again the PSNR shows little difference, because the watermark is fixed to have 40dB PSNR when it is embedded.



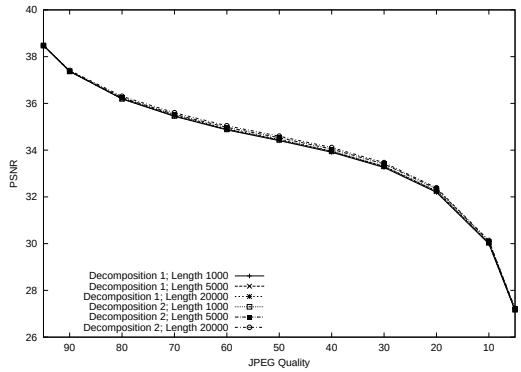
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



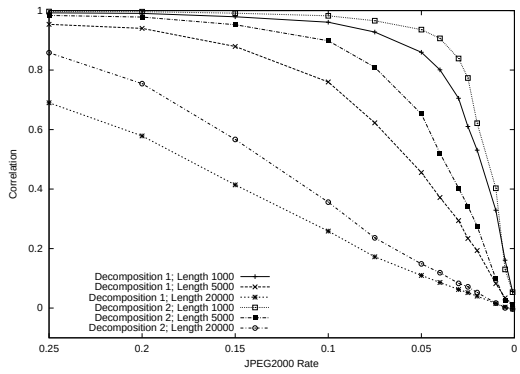
(c) JPEG Correlation



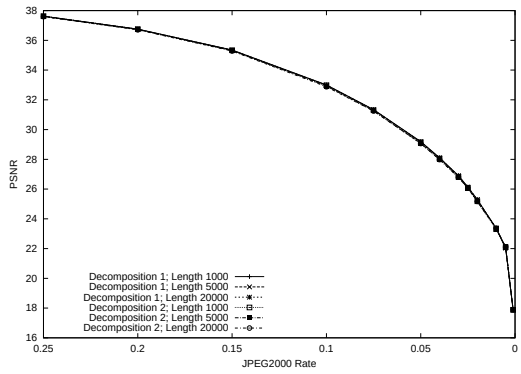
(d) JPEG PSNR

Figure 3.43: Lena: 7 levels, length comparison

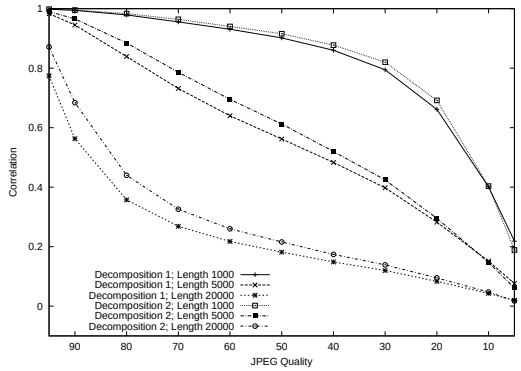
The results for "Barbara" are shown in figure 3.44.
For this image decomposition 2 is always the better performing system under both JPEG and JPEG2000 compression and for all watermark lengths.



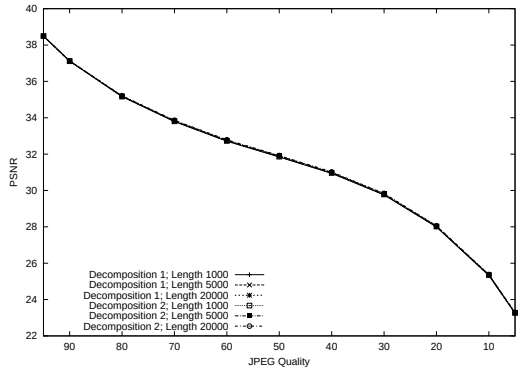
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation

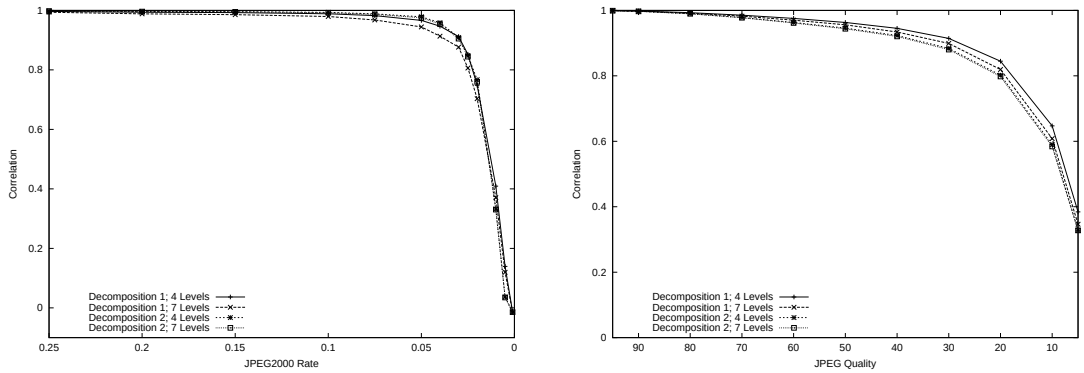


(d) JPEG PSNR

Figure 3.44: Barbara: 7 levels, length comparison

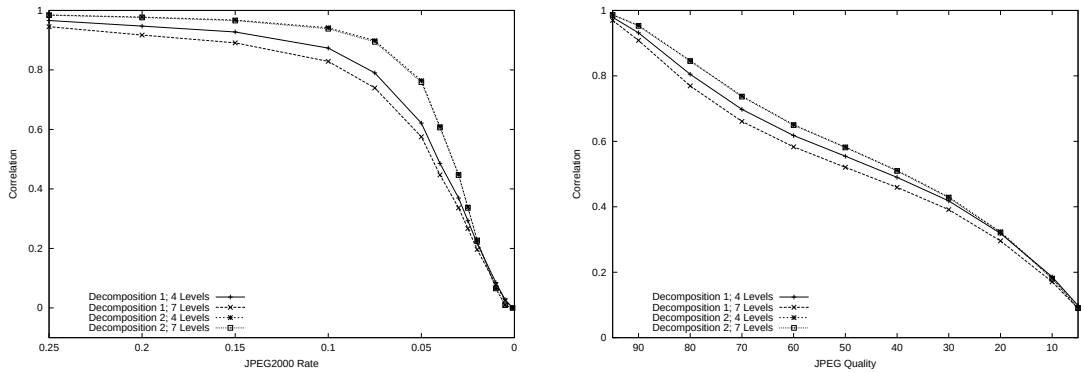
3.4.8 Levels Comparison

Figures 3.45, 3.46 and 3.47 compare the performance of decomposition 1 and 2 for both 4 and 7 levels with a watermark length of 1000, 5000 and 20000, respectively.
In all three figures and for both JPEG and JPEG2000 compression there is little difference between 4 and 7 levels for decomposition 2. But for decomposition 1 the trees with only 4 levels generate slightly better results.



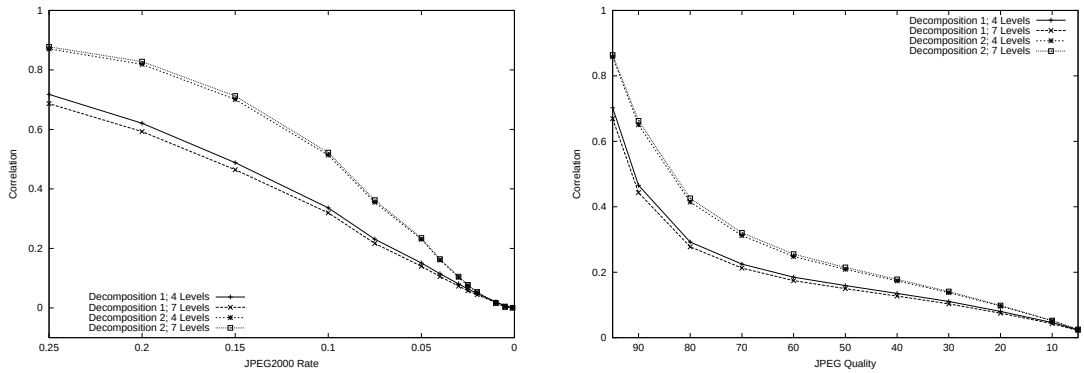
(a) JPEG2000 Correlation (b) JPEG Correlation

Figure 3.45: Lena: watermark length 1000, levels comparison



(a) JPEG2000 Correlation (b) JPEG Correlation

Figure 3.46: Lena: watermark length 5000, levels comparison



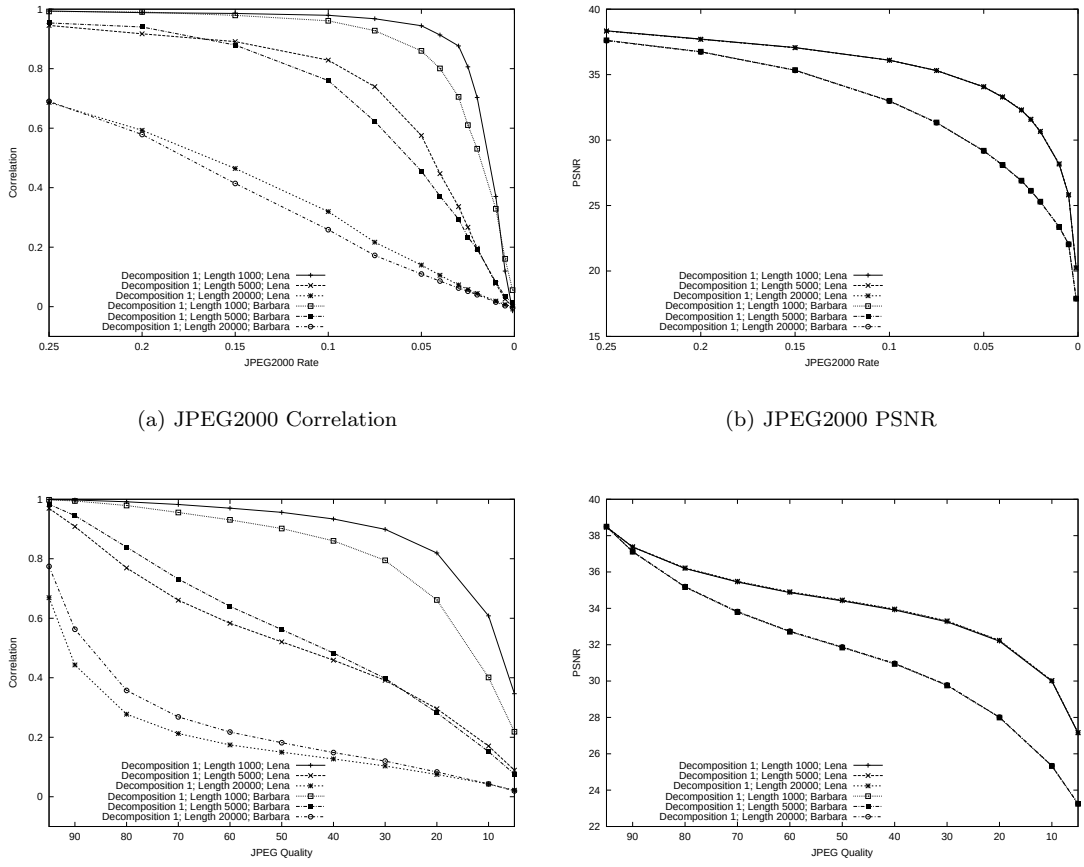
(a) JPEG2000 Correlation

(b) JPEG Correlation

Figure 3.47: Lena: watermark length 20000, levels comparison

3.4.9 Picture Comparison

The last comparison we are presenting is the performance difference between the "Lena" and the "Barbara" images.
Figure 3.48 shows the results when decomposition 1 is used. For JPEG2000 compression the correlation of Lena is higher than the correlation for Barbara. But for JPEG compression and watermarks of lengths 5000 and 20000 Barbara actually has the better results. The PSNR is better for Lena for high compression rates. This may be caused by the many textures in Barbara that take severe damage under strong compression.



(a) JPEG2000 Correlation

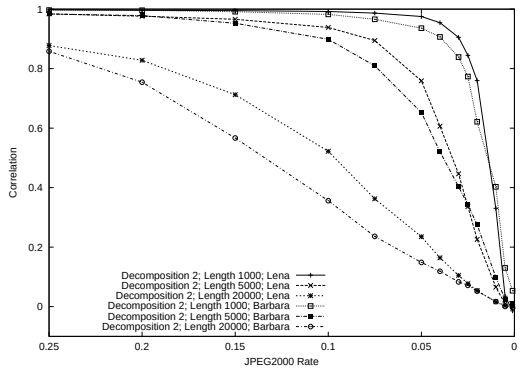
(b) JPEG2000 PSNR

(c) JPEG Correlation

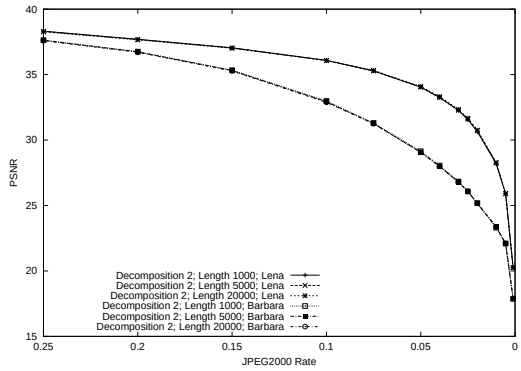
(d) JPEG PSNR

Figure 3.48: Decomposition 1, picture comparison

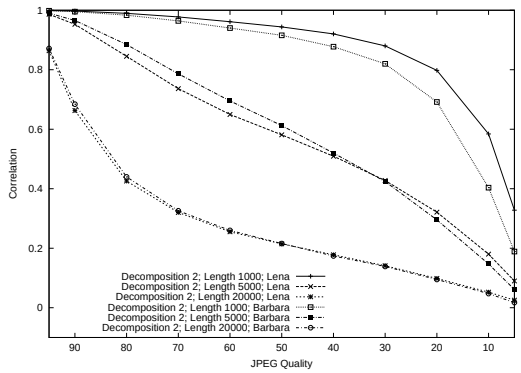
For decomposition 2 the results are shown in figure 3.49. The results are again better for the Lena image.
From these results we see that the selection of the cover image has the biggest impact on the PSNR behavior of the compressed image. The PSNR under compression depends on the image properties, like how many textured areas exist. The correlation shows less dependency on the host image. Both images show good results under compression, although Lena shows advantages for high compression ratios.



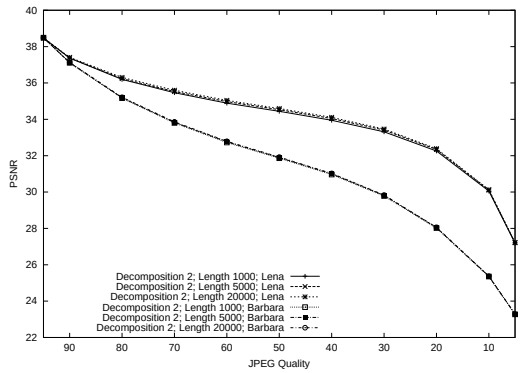
(a) JPEG2000 Correlation



(b) JPEG2000 PSNR



(c) JPEG Correlation



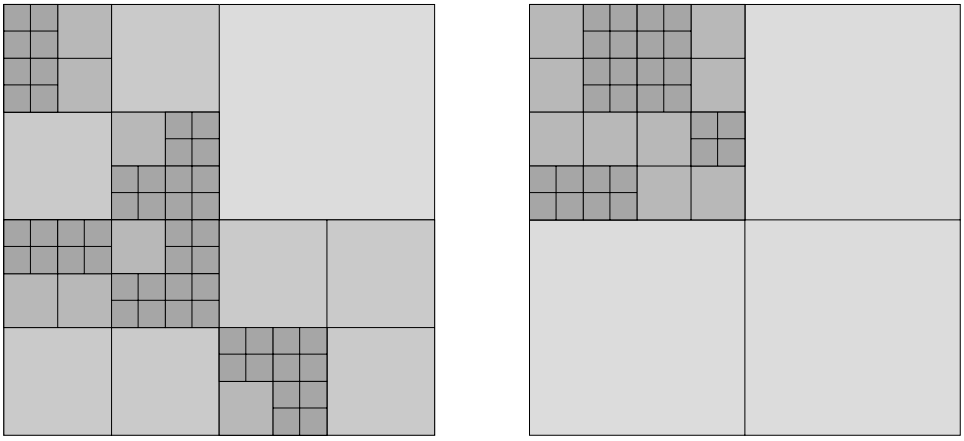
(d) JPEG PSNR

Figure 3.49: Decomposition 2, picture comparison

3.5 Example Decompositions and Images

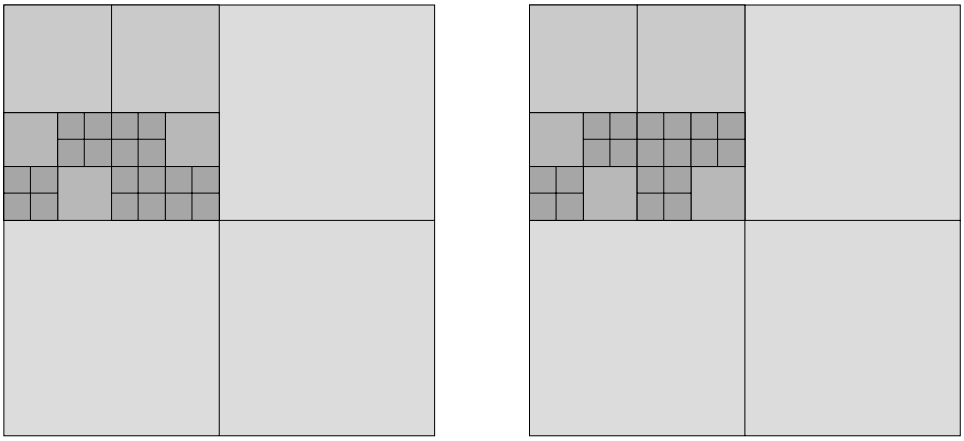
3.5.1 Tree Decompositions

The tree decompositions are independent of the cover image and the signature that is used. We show the trees for decomposition methods 1 and 2 with 4 and 7 levels for the tree numbers 150000 and 200000. Note that we use tree number 150000 in our security assessments in section 3.3.
In figure 3.51 it is interesting to note that there is little difference between tree numbers 150000 and 200000. Only one decomposition of a subband is different between the trees. This is exactly the situation that can result in higher correlation, because the common subtrees can lead to common embedding sequences.



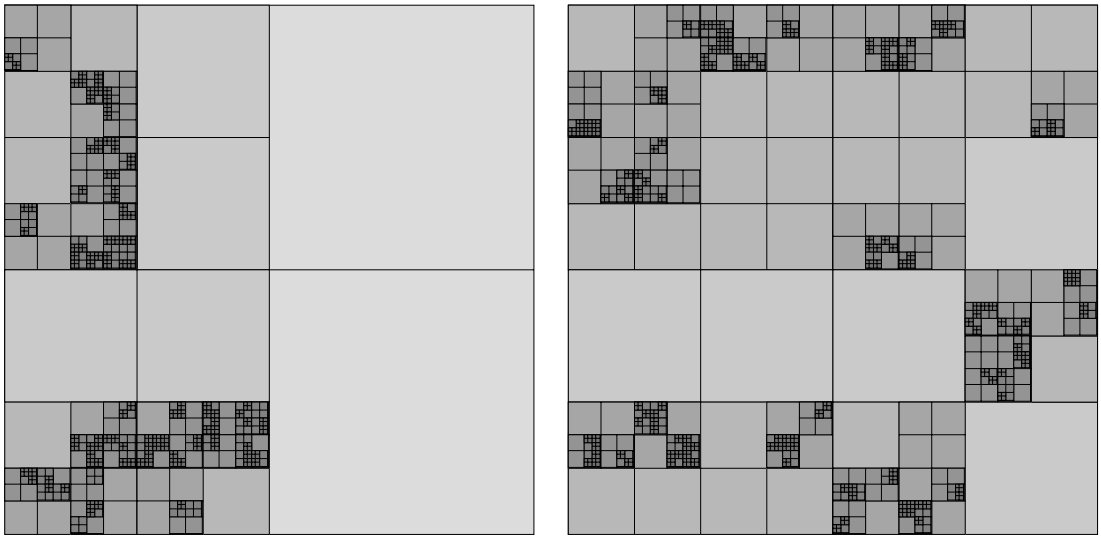
(a) Tree 150000 (b) Tree 200000

Figure 3.50: Decomposition 1, 4 levels



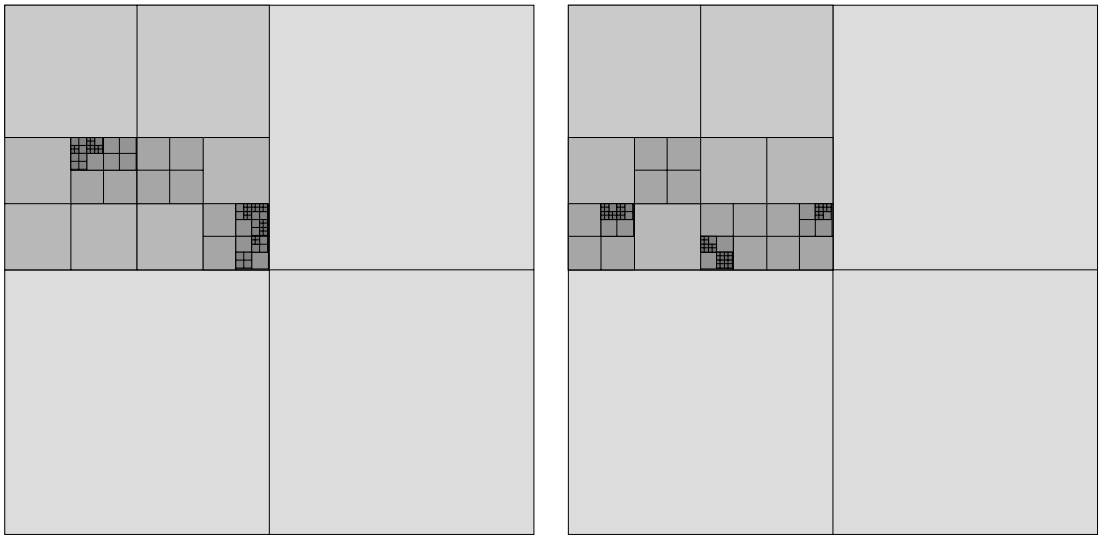
(a) Tree 150000 (b) Tree 200000

Figure 3.51: Decomposition 2, 4 levels



(a) Tree 150000 (b) Tree 200000

Figure 3.52: Decomposition 1, 7 levels



(a) Tree 150000 (b) Tree 200000

Figure 3.53: Decomposition 2, 7 levels

3.5.2 7 Levels, Watermark Length 1000

The following images are included as examples of the effects of the proposed watermarking system. On the left we have the watermarked image. The watermark is always embedded with 40dB PSNR. On the right we see the differences between the original cover image and the watermarked image.

Figure 3.54: Lena: decomposition 1, tree 150000, no variation

Figure 3.55: Lena: decomposition 1, tree 150000, coefficient skipping

Figure 3.56: Lena: decomposition 1, tree 150000, watermark shuffling

Figure 3.57: Lena: decomposition 1, tree 150000, skipping and shuffling

Figure 3.58: Lena: decomposition 2, tree 150000, no variation

Figure 3.59: Lena: decomposition 2, tree 150000, coefficient skipping

Figure 3.60: Lena: decomposition 2, tree 150000, watermark shuffling

Figure 3.61: Lena: decomposition 2, tree 150000, skipping and shuffling

Figure 3.62: Lena: decomposition 2, tree 200000, coefficient skipping

Figure 3.63: Barbara: decomposition 2, tree 150000, no variation

3.5.3 4 Levels, Watermark Length 1000

Figure 3.64: Lena: decomposition 1, tree 150000, coefficient skipping

Figure 3.65: Lena: decomposition 2, tree 150000, coefficient skipping

3.5.4 7 Levels, Watermark Length 5000

Figure 3.66: Lena: decomposition 1, tree 150000, coefficient skipping

Figure 3.67: Lena: decomposition 2, tree 150000, coefficient skipping

3.5.5 4 Levels, Watermark Length 5000

Figure 3.68: Lena: decomposition 1, tree 150000, coefficient skipping

Figure 3.69: Lena: decomposition 2, tree 150000, coefficient skipping

3.5.6 7 Levels, Watermark Length 20000

Figure 3.70: Lena: decomposition 1, tree 150000, coefficient skipping

Figure 3.71: Lena: decomposition 2, tree 150000, coefficient skipping

3.5.7 4 Levels, Watermark Length 20000

Figure 3.72: Lena: decomposition 1, tree 150000, coefficient skipping

Figure 3.73: Lena: decomposition 2, tree 150000, coefficient skipping

3.6 Conclusions

In this chapter we presented how the wavelet packet decomposition can be used to enhance the security of wavelet-based watermarking systems.
We use the Wang coefficient selection method and propose two methods for generating random trees. The first method uses a 50% probability of decomposition for all subbands. The second method does not decompose the detail subbands on the first level and puts more emphasis on decomposing in the low and middle frequency range. The seed for the random number generator is used as key and is kept secret. For 7 decomposition levels we have 2^{4185} or 2^{1046} possible tree decompositions for the first and the second decomposition method, respectively.
We also introduce three methods to protect against common subtrees that can result in higher correlation even for the wrong tree number. These variations modify the way the significant coefficients are modified to embed the watermark sequence. The first method is to skip some of the significant coefficients. In our experiments we skip 5% of the coefficients. Multiplying with another gaussian sequence is the second analyzed method to protect against common subtrees. Finally, we also experiment with applying a random permutation of the watermark sequence before we embed it in the wavelet coefficients.
The results of the security assessment show that because of common subtrees it is necessary to use one of the embedding variations. Coefficient skipping and watermark shuffling both result in improved security. The multiplication with another gaussian sequence reduces the effect of common subtrees, but the correlation remains high even without knowledge of the correct variation.
The quality assessment under both JPEG and JPEG2000 compression results in good behavior for the proposed method. For a watermark of length 1000 the tree decompositions show results slightly better than the standard systems using the Biorthogonal 7/9 or the Daubechies 6 filters. With longer watermarks of length 5000 and 20000 the advantage of the wavelet packet systems gets larger and tree decomposition 2 has higher robustness to compression than the other methods. Overall we think that a random tree decomposition that focuses on the low and middle frequency range and uses either coefficient skipping or watermark shuffling results in a robust and secure watermarking system.
Future work will focus on a combination of parametrized non-stationary wavelet filters with random wavelet packet decompositions. This will result in an even larger keyspace and allow for better adoption to image properties. Work on the random tree decomposition method should try to prevent common subtrees and investigate the influence of the embedding tree number on the system security and on the robustness under compression.

Chapter 4

Attacks on Watermark Coefficients

4.1 Introduction

In this chapter we describe malicious attacks on an embedded watermark.
We embed a watermark with the standard Wang method and with the two methods we proposed — parametrized wavelet filters and wavelet packet decomposition. The attacker only knows that we use the Wang method to select the coefficients, but does not know which filters and which decomposition tree we use.
For a realistic attack we only have the watermarked image. By applying the wavelet coefficient selection on the already watermarked image we are likely to select different coefficients from the ones that were used for embedding. The attack therefore will have to modify more coefficients and hope that the correct coefficients are attacked. As a boundary condition the quality of the image should not be damaged too much, otherwise the value of the image is lost to the attacker.
To see how effective the different attacks are at removing the watermark information from the coefficients we also implemented the attacks with the original image supplied. This is of course no realistic attack scenario, because the attacker would simply take the original image and not attack the watermarked image. But when the original image is available we can use it to select the exact wavelet coefficients that were used for embedding. The attack therefore will be applied to the coefficients that contain the watermark and we will see how effective the attack method is at removing the watermark from the coefficients.
Previously, in section 1.4, we already introduced the basic terminology and existing attacks. Now, in section 4.2 we will describe the attack methods that we implemented in more detail and then in section 4.3 we will present the results of attacking the different watermarking methods.

4.2 Description of Attacks

We embed a watermark of length 1000 and use a 7 level decomposition. The "Lena" image is used and all watermarks are measured to result in 40dB PSNR. As reference we use the standard Wang algorithm with a pyramidal decomposition using the Daubechies 6 and the Biorthogonal 7/9 filters.
For the parametrized filters we use 2, 3 and 5 parameters to generate the wavelet filters. We also test the combined system that uses five parameters per filter and four different filters for the first four decomposition levels, giving a total of 20 parameters. For each system we select 40 parameter combinations at random and embed the watermark with each combination. Then we calculate the average, minimum and maximum correlation and PSNR of the different combinations.
For the wavelet packet method we use tree decomposition 2 without an embedding variation. We

use tree numbers from 100000 to 200000 with a step size of 2500 resulting in 41 tested trees. Again we calculate the average, minimum and maximum of all tested trees.
For all attacks we use a 7 level pyramidal decomposition with either the Daubechies 6 or the Biorthogonal 7/9 filters. We attack between 100 and 80000 coefficients, but we are only presenting the range up to 20000 coefficients, because no major change happens after that.

4.2.1 Setting Coefficients to Zero

This first attack is the most effective and also the most damaging one. We simply set all the coefficients we select to zero and thereby remove all watermark information that might be contained in that coefficient.
Because the coefficients that are used for embedding the watermark are very significant for the image we will also damage the image quality severly and probably make the image worthless with this attack.

4.2.2 Fixed Quantization of Coefficients

This attack applies a fixed quantization step size to the selected coefficients. We tried step sizes of 1, 10, 20, 50 and 100. A step size of 100 is most effective at removing the watermark information and still preserving the image quality.

4.2.3 10% Scaling of Coefficients

This time we apply different attacks to the even and the odd selected wavelet coefficients. The odd coefficients are multiplied with 0.90 and then rounded to the nearest integer. The even coefficients are multiplied with 1.10 and then rounded to the nearest integer.
This way 50% of the coefficients are reduced by 10% and the other half is increased by 10%. We hope that this will remove the correlation with the watermark sequence while still preserving a good image quality.

4.2.4 Barni-Lewis Perceptual Quantization of Coefficients

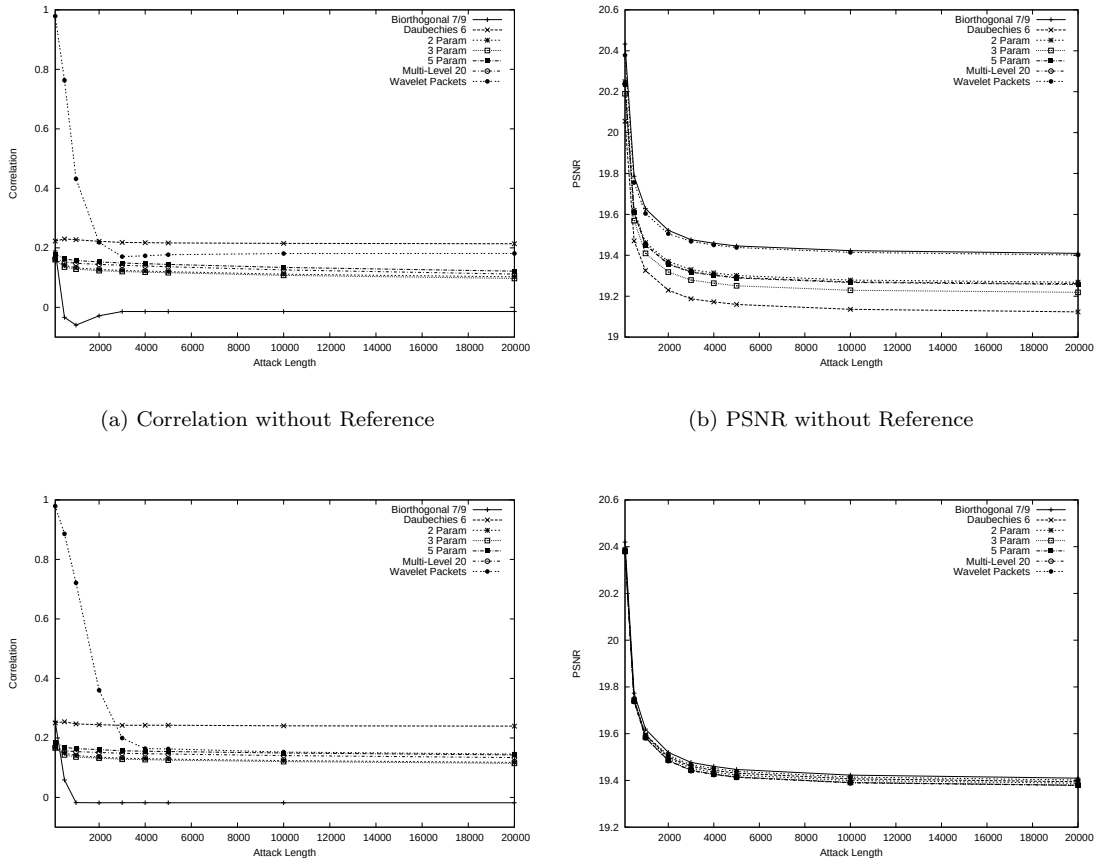
For the last attack we use a perceptual quantization model proposed by Lewis [55] and modified by Barni [4]. More information about watermarking with perceptual constraints can be found in [73, 72, 74, 5].
The quantization step size is determined for every coefficient depending on the characteristics of it and the environment it is in. This way we get a large quantization step size for perceputally insignificant coefficients and are more likely to remove the watermark and at the same time minimize the perceptual damage we do to the image.

4.3 Attacks on Watermarks

4.3.1 Setting Coefficients to Zero

Attack with Biorthogonal 7/9 filter: In figure 4.1 we show an overview of the average behavior of all the different systems when we use the Biorthogonal 7/9 filter for the attack. (a) and (b) do not have the original image as reference, (c) and (d) also have the original image as reference. We see that the correlation is very low after attacking only a few coefficients.
The correlation for the Biorthogonal 7/9 embedding filter is close to zero after modifying 1000 coefficients. This can be explained by the fact that when the decomposition filter that is used for embedding the watermark and the filter that is used for the attack are the same, then the attacked coefficients are very likely the same as the coefficients used for embedding. In (a) we do not have the reference image and therefore after 3000 attacked coefficients there is no additional change in

correlation. In (c) we have the original image and after attacking 1000 coefficients the watermark is completely removed.
The standard pyramidal system that uses the Daubechies 6 filter for embedding has a correlation of slightly above 0.20. This can be explained by the different coefficients that are used for embedding because of the different decomposition filter.
The average correlation for the parametrized systems is between 0.15 and 0.20. This can also be explained by the different decomposition filter.
In the beginning the wavelet packet system shows the highest correlation on average. After attacking 1000 coefficients we still have a correlation of over 0.40 on average.
The image quality is very severely affected by this attack. As we can see in figures 4.1(b) and (d) the PSNR is between 20.4 and 19.0dB. This is hardly an usable image anymore. The larger difference between the systems in figure (b) can be explained by the missing reference image. Because the coefficients are selected depending on the watermarked images we select different coefficients for the different systems. Therefore the average PSNR varies more without the reference image. In (d) we use the original image for all systems and therefore attack the same coefficients. This explains the smaller difference in PSNR values.
In figure 4.2 we compare the two standard systems with the average, minimum and maximum of the combined and of the wavelet packet system.
Figure 4.2(a) shows the combined Multi-Level 20 system. The minimum is just sligthly above



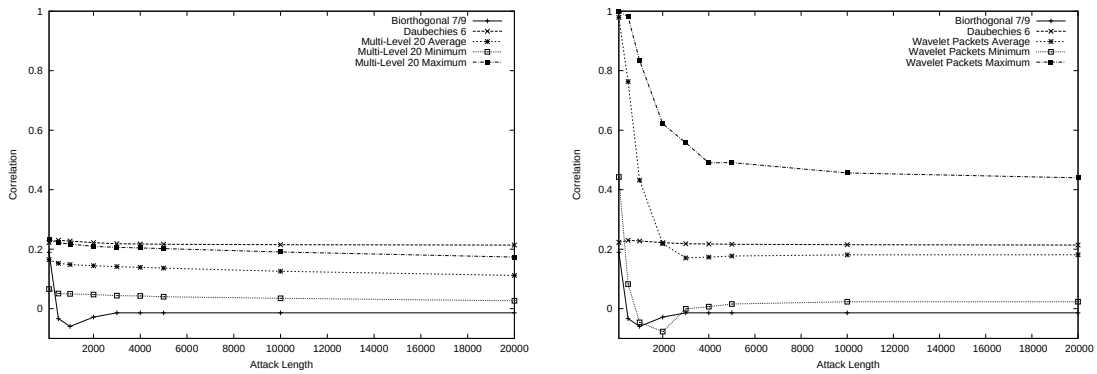
(a) Correlation without Reference

(b) PSNR without Reference

(c) Correlation with Reference

(d) PSNR with Reference

Figure 4.1: Attack with Biorthogonal 7/9; Zeroing coefficients



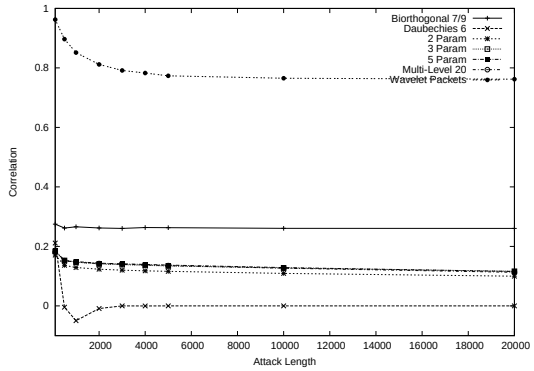
(a) Multi-Level 20

(b) Wavelet Packets

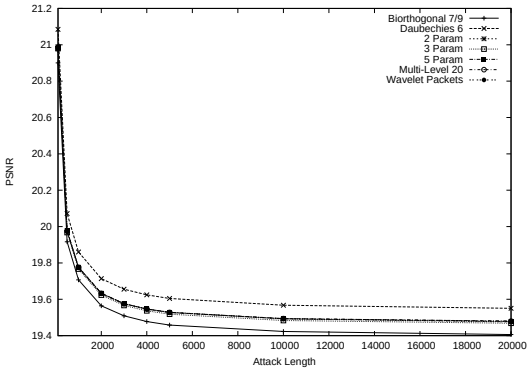
Figure 4.2: Attack with Biorthogonal 7/9; Zeroing coefficients; Correlation without reference

zero, the maximum performs about the same as the Daubechies 6 standard system, which is at around 0.20. On average we have a correlation between 0.15 and 0.20.
For the wavelet packet system we show the detailed results in figure 4.2(b). The minimum performs very similarly to the Biorthogonal 7/9 system. This can be explained by the fact that the wavelet packet system uses the Biorthogonal 7/9 filter for the decomposition. Therefore if the wavelet packet tree is in a way similar to the pyramidal decomposition the attacks are more effective. The best performing wavelet packet system has a correlation of above 0.40 even after 20000 coefficients have been attacked. On average the wavelet packet decomposition works well for only a few attacked coefficients. When more than 2000 coefficients are attacked it performs close to 0.20.

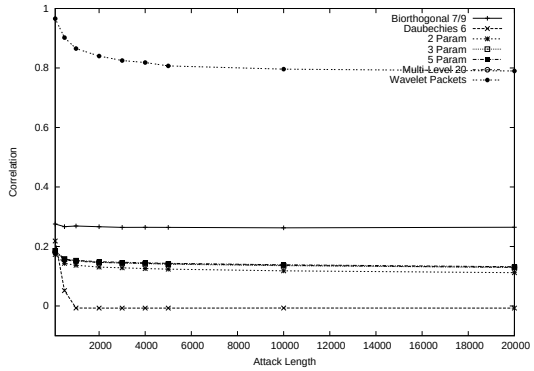
Attack with Daubechies 6 filter: Figures 4.3 and 4.4 show the results when the Daubechies 6 filter is used for the attack.
The results for the parametrized systems are comparable to the previous results.
But the wavelet packet system shows very good performance with an average correlation above 0.75 even after 20000 coefficients have been modified. And even the minimum correlation is above 0.30 for all attack lengths.
This can be explained by the fact that the wavelet packet decomposition uses the Biorthogonal 7/9 filter. So this time we combine the positive effects of using a different filter domain and also using a different decomposition structure. This made the wavelet packet system very robust to the attack with the Daubechies 6 filter.
This attack is most effective at removing the watermark at the cost of reducing the image quality to around 20dB PSNR. We see an advantage in the robustness of both the parametrized filters and the wavelet packet system. On average the combined parametrized system performs around 15 percent above zero for both attack filters. The wavelet packet system shows better results when the embedding filter is different from the attack filter. The average for an attack with the Biorthogonal 7/9 filter is around 0.20, for an attack with the Daubechies 6 filter above 0.75.



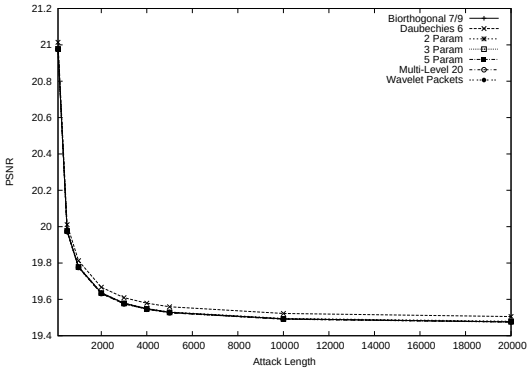
(a) Correlation without Reference



(b) PSNR without Reference

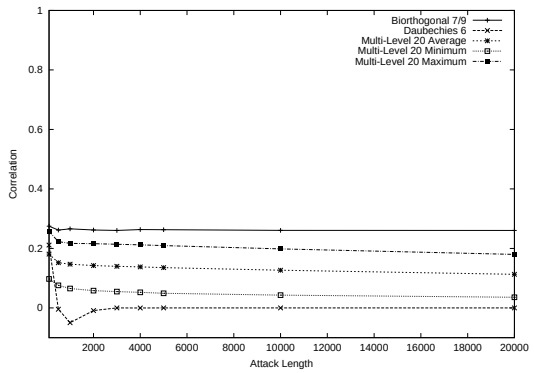


(c) Correlation with Reference

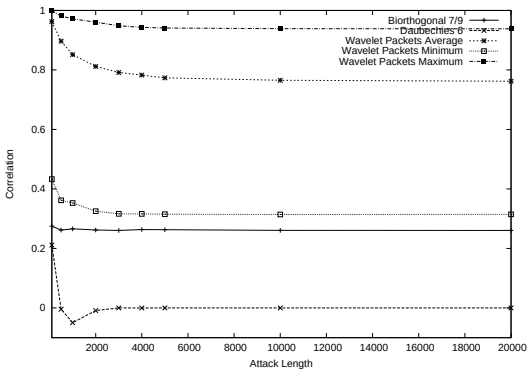


(d) PSNR with Reference

Figure 4.3: Attack with Daubechies 6; Zeroing coefficients



(a) Multi-Level 20



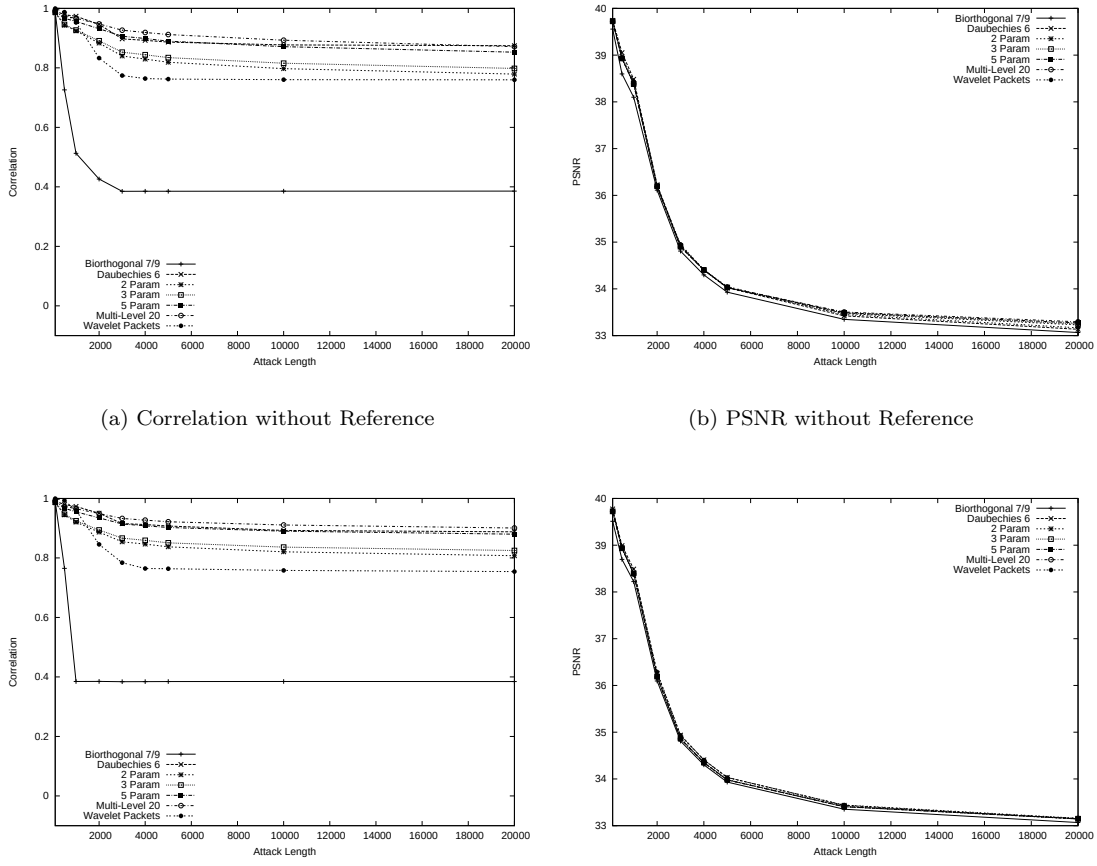
(b) Wavelet Packets

Figure 4.4: Attack with Daubechies 6; Zeroing coefficients; Correlation without reference

4.3.2 Fixed Quantization of Coefficients

The second attack we want to describe in more detail is quantization with a fixed step size of 100.

Attack with Biorthogonal 7/9 filter: Figure 4.5 compares the correlation and PSNR of the different systems, with and without the reference image, when we use the Biorthogonal 7/9 filter for the attack. In (a) we see that the standard system that also uses the Biorthogonal 7/9 filter has a correlation of around 0.40 after 3000 coefficients have been attacked. In (c) we see the result when the reference image is available. After attacking the 1000 coefficients that contain the watermark the correlation has dropped to 0.40.



(a) Correlation without Reference

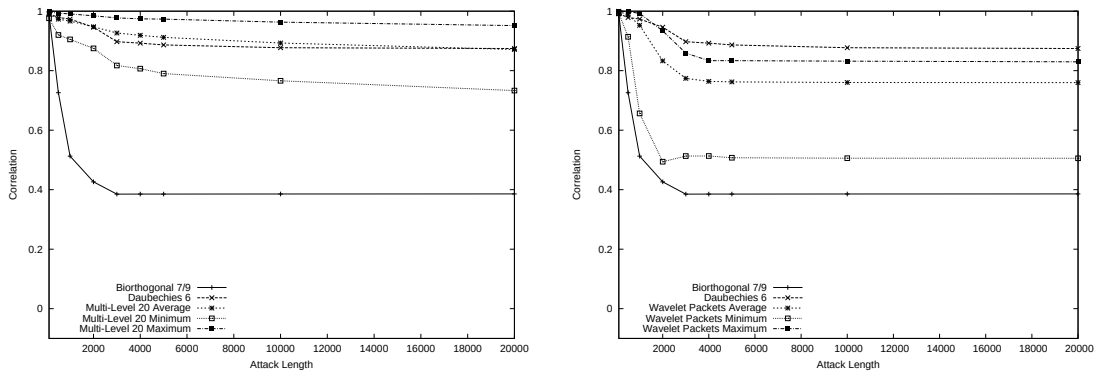
(b) PSNR without Reference

(c) Correlation with Reference

(d) PSNR with Reference

Figure 4.5: Attack with Biorthogonal 7/9; Quantization step size 100

The parametrized systems and the wavelet packet system all perform more than 0.30 better.
The average combined system performs best with a correlation of nearly 90 percent even after 20000 coefficients have been attacked. The worst performer under this attack is the wavelet packet system which still has a correlation of more than 0.75 after 20000 coefficients have been attacked.
The advantage of this attack is that the PSNR is not degraded as badly. After attacking 20000 coefficients we still have a PSNR above 33dB. Because we only need to attack 2000 or 3000 coefficients to significantly reduce the correlation for the standard system we can have an attacked image quality of 36 or 35dB respectively.



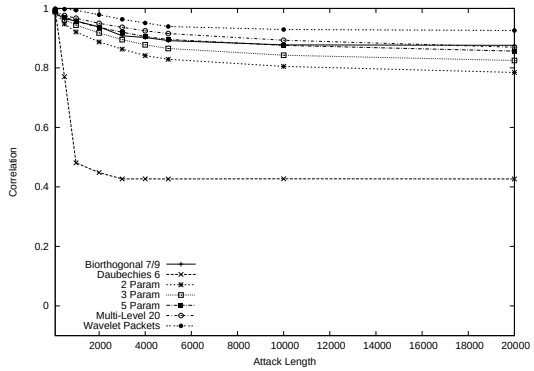
(a) Multi-Level 20

(b) Wavelet Packets

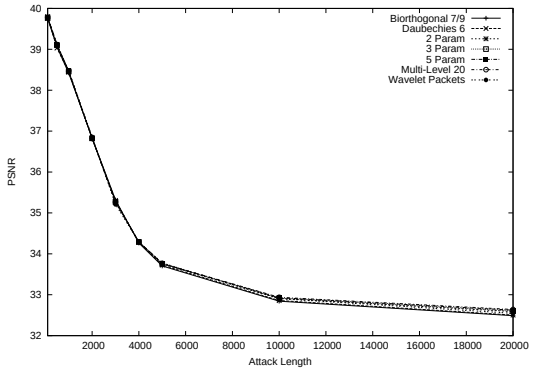
Figure 4.6: Attack with Biorthogonal 7/9; Quantization step size 100; Correlation without reference

The average, minimum and maximum are shown in figure 4.6. The low minimum for the wavelet packet system can again be explained by the possibility of a similarity between the pyramidal decomposition used for the attack and the tree that is used for embedding. Because both decompositions use the Biorthogonal 7/9 filter the minimum can get very close to the behavior of the standard system.

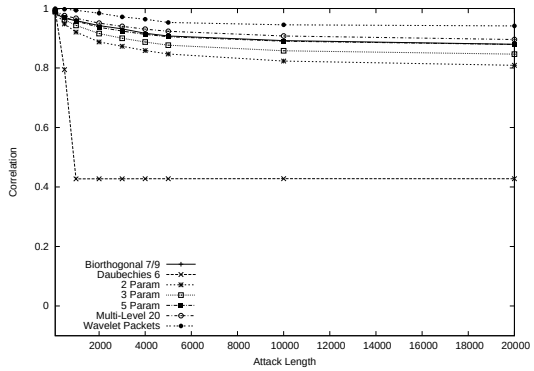
Attack with Daubechies 6 filter: Figure 4.7 shows the result when the Daubechies 6 filter is used for the attack decomposition. The results for the parametrized systems are nearly unchanged. The performance of the wavelet packet system is better, because now the decomposition filters do not match.
This can also be seen in figure 4.8. The minimum wavelet packet system correlation is now more than 40 percent above the correlation of the standard Daubechies 6 system for 20000 attacked coefficients.
With this attack we see a clear advantage of our proposed systems. With the standard Biorthogonal 7/9 and Daubechies 6 embedding methods the correlation drops below 40 percent while the image quality is still reasonable. This means that an un-watermarked image can be obtained without a severe quality reduction.
With both the parametrized filters and the wavlet packet decomposition we get significantly higher correlation and could still proof ownership of the images.



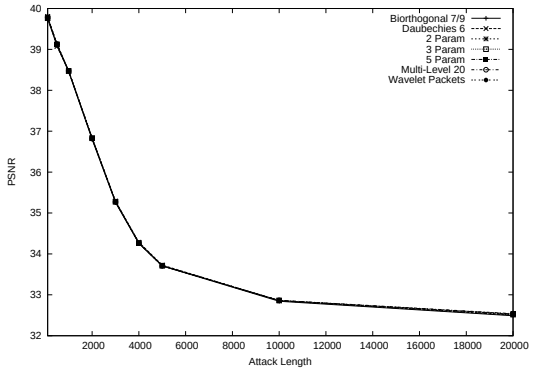
(a) Correlation without Reference



(b) PSNR without Reference

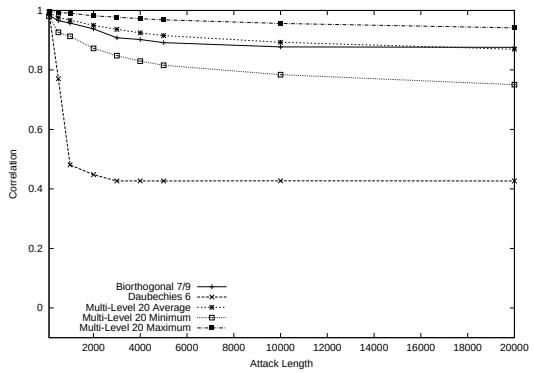


(c) Correlation with Reference

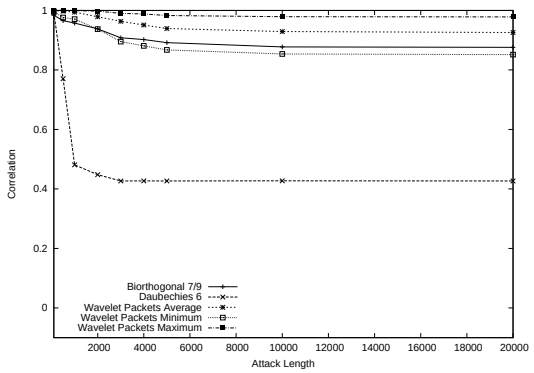


(d) PSNR with Reference

Figure 4.7: Attack with Daubechies 6; Quantization step size 100



(a) Multi-Level 20

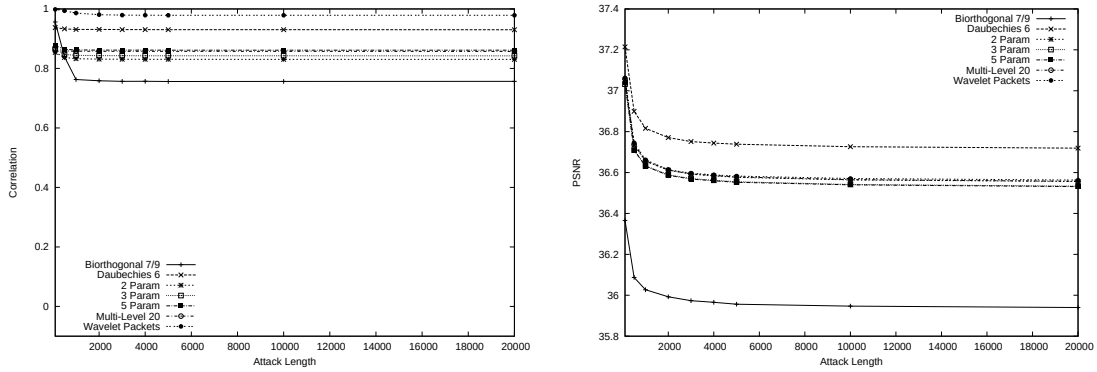


(b) Wavelet Packets

Figure 4.8: Attack with Daubechies 6; Quantization step size 100; Correlation without reference

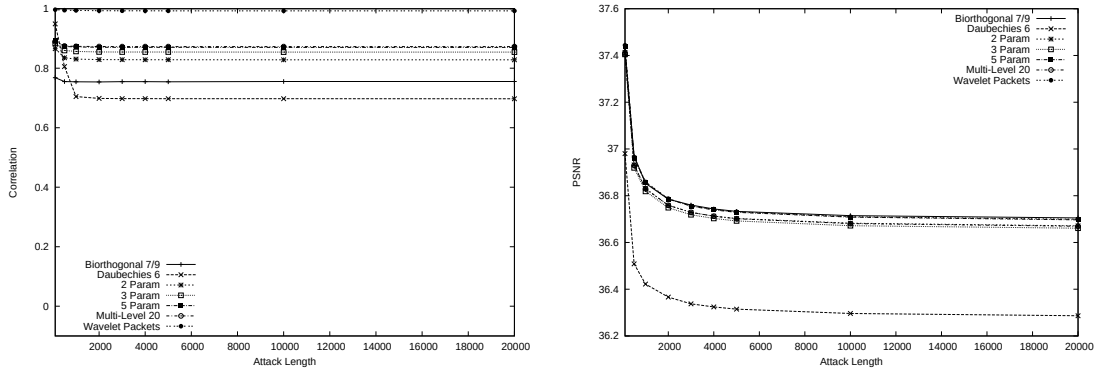
4.3.3 10% Scaling of Coefficients

The results for this attack are shown in diagrams 4.9 and 4.10.
With this attack we have a smaller advantage for the proposed systems. The correlation of the parametrized systems is between 5 and 15 percent above the corresponding standard filter correlation. The attack shows very little effect on the wavelet packet system, leaving its correlation above 0.95 even after 20000 coefficients have been attacked. The attack only results in mild degradation of the image quality. Even after 20000 modified coefficients we still have a PSNR around 36dB.
The disadvantage of this attack is that the correlation does not drop below 0.75 for any system, so there is no clear advantage for the proposed systems.



(a) Correlation (b) PSNR

Figure 4.9: Attack with Biorthogonal 7/9; 10% Scaling; Without reference

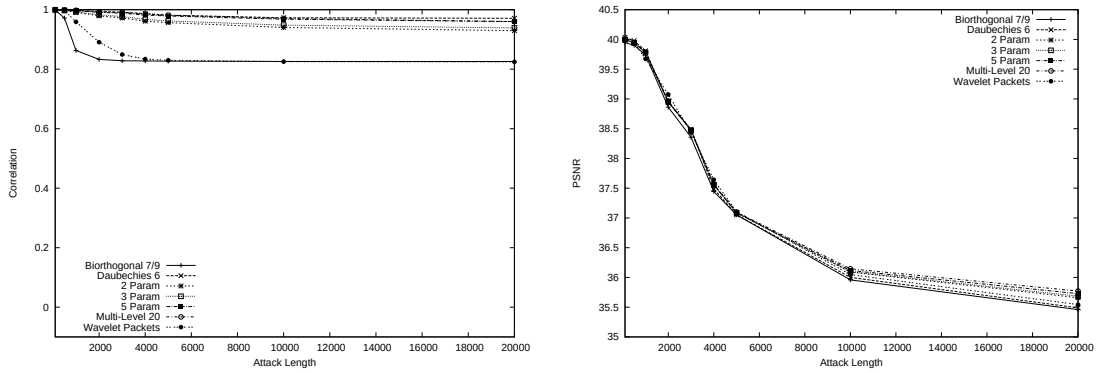


(a) Correlation (b) PSNR

Figure 4.10: Attack with Daubechies 6; 10% Scaling; Without reference

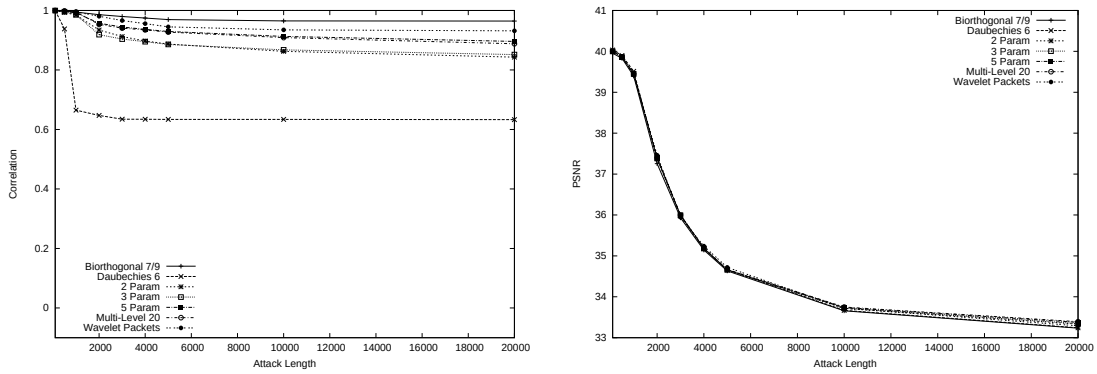
4.3.4 Barni-Lewis Perceptual Quantization of Coefficients

For this attack the filter that is used for the decomposition has a stronger influence. Figure 4.11 shows the results when the Biorthogonal 7/9 filter is used for the attack. The parametrized systems have around 0.10 higher correlation, but all systems have a correlation of above 0.80. Therefore there is no clear advantage for the proposed systems.
The results for the Daubechies 6 filter are shown in figure 4.12. This time the proposed systems have at least 20 percent higher correlation after the attacks. The advantage of the proposed systems is larger this time, but again the correlation never drops below 0.60.
The image quality is not affected too badly by this attack. For the Biorthogonal 7/9 attack-filter the PSNR is always above 35dB, for the Daubechies 6 filter it is always above 33dB — even after 20000 attacked coefficients.



(a) Correlation (b) PSNR

Figure 4.11: Attack with Biorthogonal 7/9; Perceptual Quantization; Without reference



(a) Correlation (b) PSNR

Figure 4.12: Attack with Daubechies 6; Perceptual Quantization; Without reference

4.4 Example Images

In this section we want to visualize the effects of the different attacks by showing a few example images. For all attacks we use the Lena image as input and embed a watermark of length 1000 with 40dB PSNR using the standard Biorthogonal 7/9 filter. Then we attack the image with the specified method. We always use the Biorthogonal 7/9 filter for the attack decomposition. All figures show the image after the attack on the left and the difference image between the original and the attacked image on the right.
The most severe effects are achieved by zeroing the significant coefficients. Figure 4.13, 4.14 and 4.15 show the results after 100, 1000 and 20000 coefficients have been set to zero, respectively.
In figure 4.16 we see what happens after quantizing the significant coefficients with a step size of 100. Most changes happen along the edges and the attacked image is still of very good quality.
Scaling 20000 coefficients by $\pm 10\%$ produces the results shown in figure 4.17.
Finally, figure 4.18 contains the images for perceptually quantizing 20000 coefficients. Most differences are again around edges or in strongly textured regions.

Figure 4.13: Lena after zeroing 100 coefficients; PSNR = 20.39 dB

Figure 4.14: Lena after zeroing 1000 coefficients; PSNR = 19.63 dB

Figure 4.15: Lena after zeroing 20000 coefficients; PSNR = 19.41 dB

Figure 4.16: Lena after quantizing 20000 coefficients with step size 100; PSNR = 33.08 dB

Figure 4.17: Lena after scaling 20000 coefficients ±10%; PSNR = 35.94 dB

Figure 4.18: Lena after perceptually quantizing 20000 coefficients; PSNR = 35.45 dB

4.5 Conclusions

In this chapter we presented four different methods for malicious removal attacks and analyzed whether our proposed systems are more resistant to those attacks.
The most effective attack is quantization with a step size of 100. Considering the impact of this attack we see a clear advantage of our proposed systems. With the standard Biorthogonal 7/9 and Daubechies 6 embedding methods the correlation drops below 40 percent while the quality of the attacked image is still reasonable. This means that an un-watermarked image can be obtained without a severe quality reduction.
Using secret parametrized filters or random wavelet packet decompositions we maintain significantly higher correlation under attack and could still proof ownership of the attacked images. In order to remove the watermark we need to reduce the image quality below an acceptable level. This clearly demonstrates that embedding the watermark in a key-dependent wavelet transform domain provides superior resilience against unauthorized removal attacks.
Future work will focus on other custom attacks on the wavelet coefficients and investigate the influence of the embedding variations on the attack robustness.

Appendix A

Example Images

In this chapter we include some additional images and figures.
Section A.1 shows the two original images “Lena” and “Barbara” that are used in the experiments.
Sections A.2 and A.3 show the effects of JPEG and JPEG2000 compression on the original images without watermarks. This should help in evaluating the robustness assessments and visualize the artifacts that are created by strong compression.
Sections A.4 and A.5 show wavelet decompositions of the original images using different filters.

A.1 Original Images

Figure A.1: Uncompressed Lena; 512 x 512 pixels; grayscale, 8 bits per pixel

Figure A.2: Uncompressed Barbara; 512 x 512 pixels; grayscale, 8 bits per pixel

A.2 Compressed Lena

JPEG2000 Compression

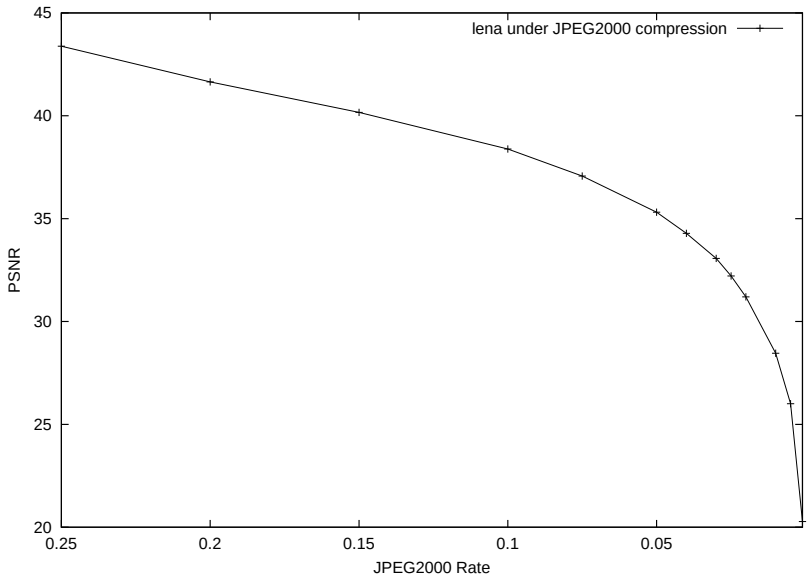


Figure A.3: Lena: PSNR under JPEG2000 Compression

JPEG Compression

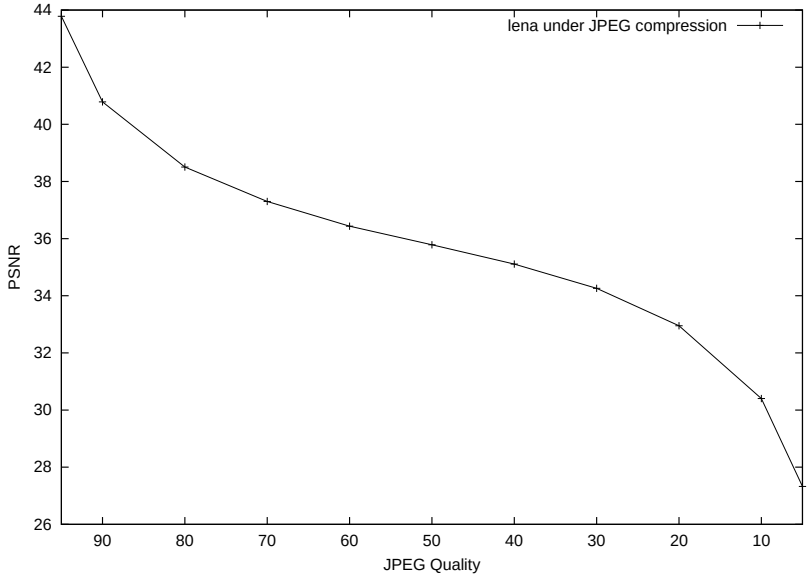


Figure A.4: Lena: PSNR under JPEG Compression

(a) JPEG2000 Rate 0.15; Compression 6.6:1; PSNR = 40.16 dB

(b) JPEG2000 Rate 0.075; Compression 13.3:1 ; PSNR = 37.07 dB

(c) JPEG2000 Rate 0.02; Compression 50:1 ; PSNR = 31.20 dB

(d) JPEG2000 Rate 0.001; Compression 1000:1 ; PSNR = 20.27 dB

Figure A.5: Lena under JPEG2000 compression

(a) JPEG Quality 90; Compression 4.4:1 ; PSNR = 40.78 dB

(b) JPEG Quality 70; Compression 8.9:1 ; PSNR = 37.30 dB

(c) JPEG Quality 10; Compression 32.5:1 ; PSNR = 30.40 dB

(d) JPEG Quality 5; Compression 45.7:1 ; PSNR = 27.32 dB

Figure A.6: Lena under JPEG compression

A.3 Compressed Barbara

JPEG2000 Compression

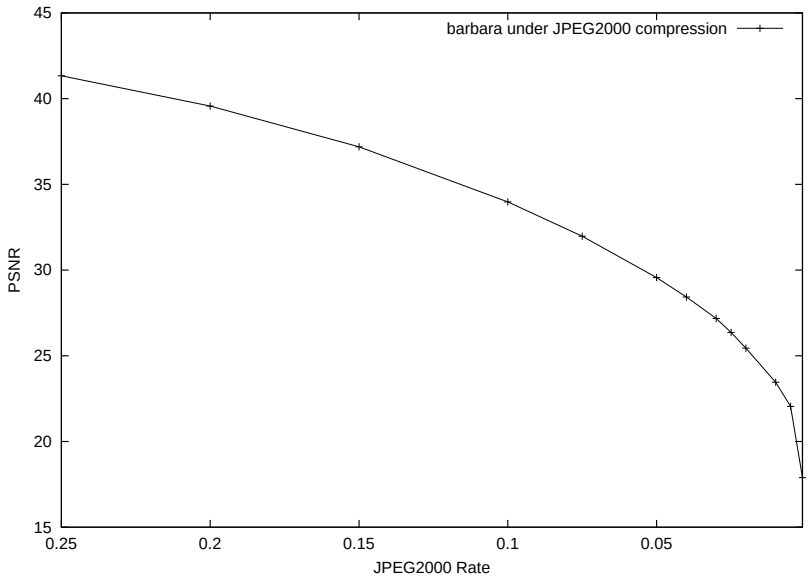


Figure A.7: Barbara: PSNR under JPEG2000 Compression

JPEG Compression

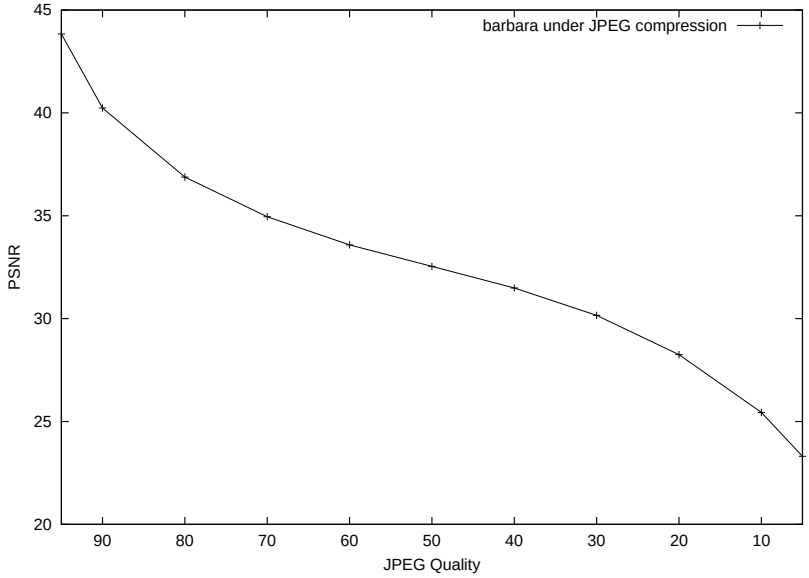
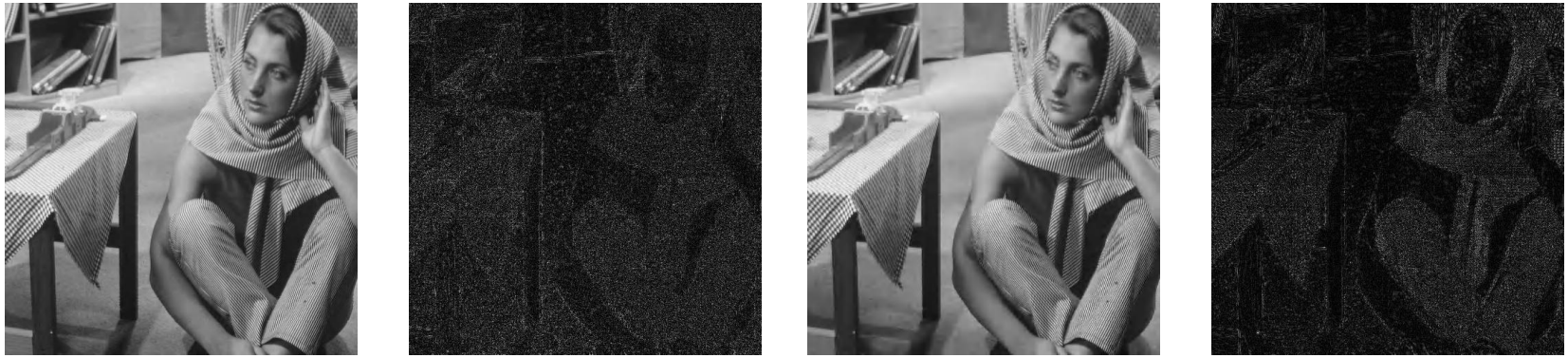


Figure A.8: Barbara: PSNR under JPEG Compression



(a) JPEG2000 Rate 0.15; Compression 6.6:1; PSNR = 37.19 dB

(b) JPEG2000 Rate 0.075; Compression 13.3:1 ; PSNR = 31.98 dB

(c) JPEG2000 Rate 0.02; Compression 50:1 ; PSNR = 25.44 dB

(d) JPEG2000 Rate 0.001; Compression 1000:1 ; PSNR = 17.89 dB

Figure A.9: Barbara under JPEG2000 compression

(a) JPEG Quality 90; Compression 3.5:1 ; PSNR = 40.24 dB

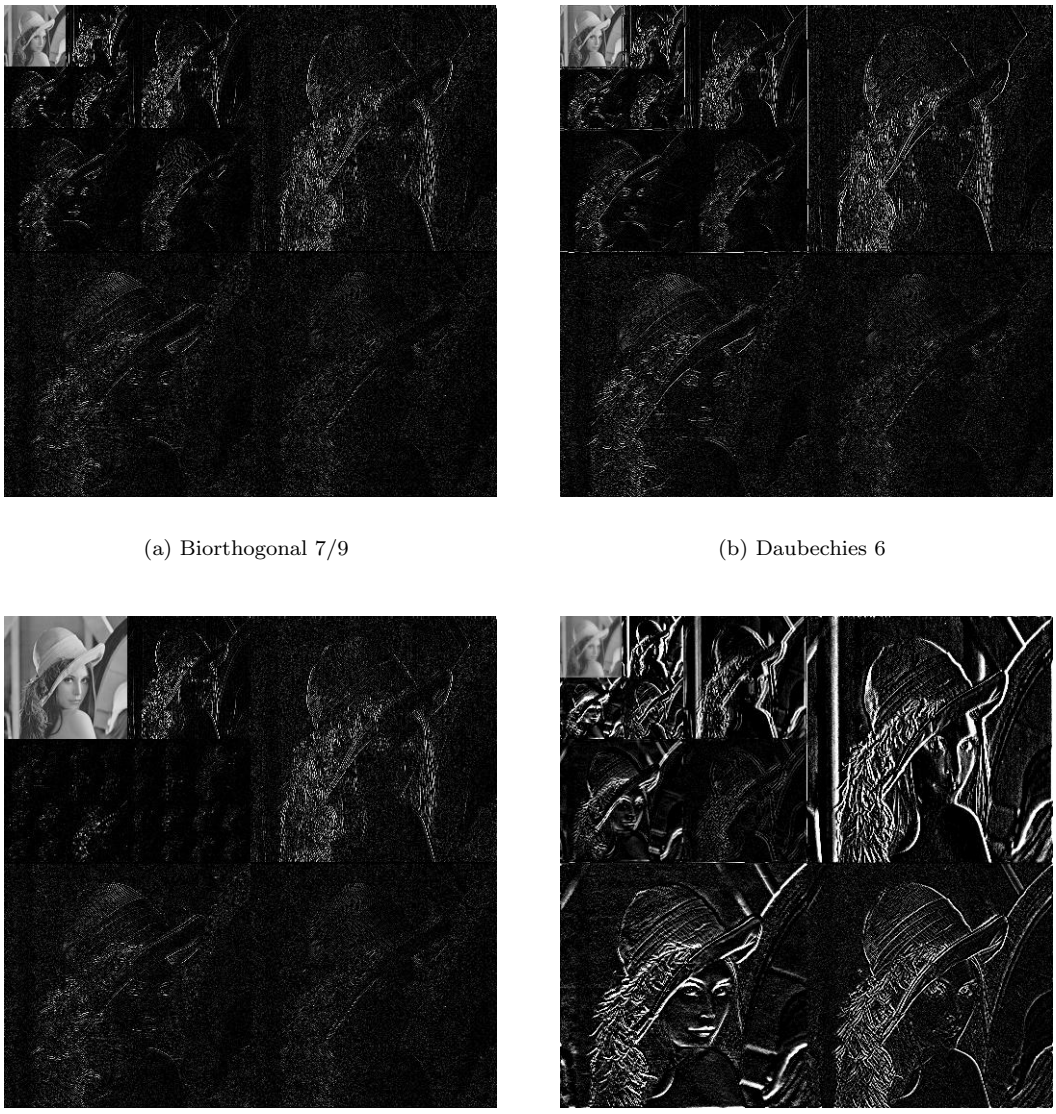
(b) JPEG Quality 70; Compression 6.4:1 ; PSNR = 34.96 dB

(c) JPEG Quality 10; Compression 24.2:1 ; PSNR = 25.44 dB

(d) JPEG Quality 5; Compression 40.4:1 ; PSNR = 23.31 dB

Figure A.10: Barbara under JPEG compression

A.4 Decomposed Lena



(a) Biorthogonal 7/9

(b) Daubechies 6

(c) Tree 150000

(d) Parametrized

Figure A.11: Decomposed Lena

A.5 Decomposed Barbara

(a) Biorthogonal 7/9

(b) Daubechies 6

(c) Tree 150000

(d) Parametrized

Figure A.12: Decomposed Barbara

Appendix B

Development Environment

For the measurements in chapter 2 we used the programs developed by Peter Meerwald.
`http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/`
For chapters 3 and 4 we developed programs in C++ using the Ganesh++ library.
`http://www.ganesh.org/`
Other tools used include:

- `Perl` scripts for the automation of the tests were used. `http://www.perl.com/`
- `The Gimp` for manual image manipulations. `http://www.gimp.org/`
- `convert`, from the ImageMagick packet, for automatic image manipulations. `http://www.imagemagick.org/`
- `XFig` for vector graphics. `http://epb.lbl.gov/xfig/`
- `JasPer` for JPEG2000 compression. `http://www.ece.uvic.ca/~mdadams/jasper/`
- `cjpeg` and `djpeg` from the Independent JPEG Group. `http://www.ijg.org/`
- This thesis document was typeset using LaTeX2e and written with `Emacs` or `XEmacs`. `http://www.ctan.org/`, `http://www.gnu.org/software/emacs/emacs.html`, `http://www.xemacs.org/`
- All work was performed on `Linux` workstations, usually with a `Debian` distribution. `http://www.kernel.org/`, `http://www.debian.org/`

And all the other `GNU` goodies (`http://www.gnu.org/`) I am so used to. Thanks to all the developers!

Bibliography

[1] A.N. Akansu and R.A. Haddad. *Multiresolution signal decomposition*. Academic Press, Boston, San Diego, 1992.

[2] Faisal Alturki and Russell M. Mersereau. Secure fragile digital watermarking technique for image authentication. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '01*, Thessaloniki, Greece, October 2001.

[3] Peter Amon. Signal processing attacks on watermarks. online presentation, January 1999.

[4] Mauro Barni, Franco Bartolini, and Alessandro Piva. Improved wavelet-based watermarking through pixel-wise masking. *IEEE Transactions on Image Processing*, 10(5):783–791, May 2001.

[5] M. Bertran, Jean-François Delaigle, and B. Macq. Some improvements to HVS models for fingerprinting in perceptual decompressors. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '01*, Thessaloniki, Greece, October 2001.

[6] Dan Boneh and James Shaw. Collusion secure fingerprinting for digital data. In Don Coppersmith, editor, *Proceedings of the 15th annual International Cryptology Conference*, volume 963 Series: Lecture Notes in Computer Science, pages 452–465, Santa Barbara, CA, USA, August 1995. Springer-Verlag.

[7] Jonathan N. Bradley, C. M. Brislawn, and T. Hopper. The FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression. In *SPIE Proceedings, Visual Information Processing II*, volume 1961, pages 293–304, Orlando, FL, USA, April 1993.

[8] K.R. Castleman. *Digital Image Processing*. Prentice Hall, 1996.

[9] R. Chandramouli and Nasir Memon. Analysis of LSB based image steganography techniques. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '01*, Thessaloniki, Greece, October 2001.

[10] C.K. Chui. *Wavelets: A Mathematical Tool for Signal Analysis*. SIAM, 1997.

[11] I. J. Cox and M. L. Miller. Electronic watermarking: the first 50 years. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing, MMSP '01*, Cannes, France, October 2001.

[12] Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal G. Shamoon. Secure spread spectrum watermarking for multimedia. Technical report, NEC Research Institure, Princeton, USA, October 1995.

[13] Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal G. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '96*, pages 243–246, Lausanne, Switzerland, September 1996. IEEE Press.

[14] Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal G. Shamoon. Secure spread spectrum watermarking for multimedia. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '97*, volume 6, pages 1673–1687, Santa Barbara, California, USA, October 1997.

[15] Ingemar J. Cox and Jean-Paul Linnartz. Some general methods of tampering with watermarks. *IEEE Journal on Selected Areas of Communications*, 16(4):587–602, May 1998.

[16] Ingemar J. Cox, Matthew L. Miller, and Jeffrey A. Bloom. Watermarking applications and their properties. In *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, 2000.

[17] Ingemar J. Cox, Matthew L. Miller, and Jeffrey A. Bloom. *Digital Watermarking*. Morgan Kaufmann, 2002.

[18] Scott A. Craver, John P. McGregor, Min Wu, Bede Liu, Adam Stubblefield, Ben Swartzlander, Dan S. Wallach, Drew Dean, and Edward W. Felten. Reading between the lines: lessons from the SDMI challenge. In *Proceedings of the 4th Information Hiding Workshop '01*, Portland, OR, USA, April 2001.

[19] Scott A. Craver, Nasir Memon, Boon-Lock Yeo, and Minerva M. Yeung. Can invisible watermarks resolve rightful ownerships? Technical Report 20509, IBM Research Report, July 1996.

[20] Scott A. Craver, Nasir Memon, Boon-Lock Yeo, and Minerva M. Yeung. On the invertibility of invisible watermarking techniques. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '97*, volume 1, page 540, Santa Barbara, California, USA, October 1997.

[21] Scott A. Craver, Boon-Lock Yeo, and Minerva M. Yeung. Technical trials and legal tribulations. *Communications of the ACM*, 41(7):45–54, July 1998.

[22] Ingrid Daubechies. *Ten Lectures on Wavelets*. Number 61 in CBMS-NSF Series in Applied Mathematics. SIAM Press, Philadelphia, PA, USA, 1992.

[23] Jana Dittmann. Combining digital watermarks and collusion secure fingerprints for customer copy monitoring. In *Proceedings of IEE Electronics & Communications, Secure Images and Image Authentication*, pages 1–6, London, UK, 2000.

[24] Jana Dittmann, editor. *Digitale Wasserzeichen: Grundlagen, Verfahren, Anwendungsgebiete*. Springer Verlag, 2000.

[25] Jana Dittmann, Alexander Behr, Mark Stabenau, Peter Schmitt, Jörg Schwenk, and Johannes Ueberberg. Combining digital watermarks and collusion secure fingerprints for digital images. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of the 11th SPIE Annual Symposium, Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, volume 3657, San Jose, CA, USA, January 1999.

[26] J.-L. Dugelay, C. Rey, G. Doërr, and G. K. Csurka. Dewatermarking based on self-similarities. Technical Report 02-060, January 2002.

[27] Jean-Luc Dugelay and Fabien A. P. Petitcolas. Possible counter-attacks against random geometric distortions. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971, San Jose, CA, USA, January 2000.

[28] Jean-Luc Dugelay and Stephane Roche. Fractal transform based large digital watermark embedding and robust full blind extraction. In *Proceedings of the IEEE International Conference on Multimedia & Computing Systems, ICMCS '99*, volume 2, pages 1003–1004, Florence, Italy, June 1999.

[29] Joachim J. Eggers, R. Bäuml, and Bernd Girod. Digital watermarking facing attacks by amplitude scaling and additive white noise. In *4th International ITG Conference on Source and Channel Coding*, Berlin, Germany, 30, 2002.

[30] Ahmet M. Eskicioglu and Edward J. Delp. An overview of multimedia content protection in consumer electronics. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971, San Jose, CA, USA, January 2000.

[31] Ahmet M. Eskicioglu and Edward J. Delp. An overview of multimedia content protection in consumer electronics devices. *Signal Processing: Image Communication*, 16(7):681–699, 2001.

[32] V. Fotopoulos and A. N. Skodras. JPEG2000 parameters against watermarking. In *Proceedings of the 14th International Conference on Digital Signal Processing, DSP '02*, pages 712–716, Santorini, Greece, July 2002.

[33] Elke Franz, Anja Jerichow, Steffen Möller, Andreas Pfitzmann, and Ingo Stierand. Computer based steganography: How it works and why therefore any restrictions on cryptography are nonsense, at best. In Ross Anderson, editor, *Information hiding: first international workshop*, volume 1174 of *Lecture Notes in Computer Science*, Cambridge, UK, 1996. Springer Verlag, Berlin, Germany.

[34] Jiri Fridrich. Key-dependent random image transforms and their applications in image watermarking. In *Proceedings of the 1999 International Conference on Imaging Science, Systems, and Technology, CISST '99*, pages 237–243, Las Vegas, NV, USA, June 1999.

[35] Jiri Fridrich, Arnold C. Baldoza, and Richard J. Simard. Robust digital watermarking based on key-dependent basis functions. In David Aucsmith, editor, *Information hiding: second international workshop*, volume 1525 of *Lecture notes in computer science*, pages 143–157, Portland, OR, USA, April 1998. Springer Verlag, Berlin, Germany.

[36] Jiri Fridrich, Rui Du, and Meng Long. Steganalysis of LSB encoding in color images. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '00*, New York, NY, USA, July 2000.

[37] R.C. Gonzalez and R.E. Woods. *Digital Image Processing – Second Edition*. Prentice-Hall, 2002.

[38] J.C. Goswami and A.K. Chan. *Fundamentals of Wavelets: Theory, Algorithms, and Applications*. Wiley, 1999.

[39] F. Hartenstein. Parametrization of discrete finite biorthogonal wavelets with linear phase. In *Proceedings of the 1997 International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, April 1997.

[40] Frank Hartung, Jonathan K. Su, and Bernd Girod. Spread spectrum watermarking: Malicious attacks and counter-attacks. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of the 11th SPIE Annual Symposium, Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, volume 3657, San Jose, CA, USA, January 1999.

[41] T. Hopper. Compression of gray-scale fingerprint images. In H.H. Szu, editor, *Wavelet Applications*, volume 2242 of *SPIE Proceedings*, pages 180–187, 1994.

[42] Junichi Ishimaru. Examples of watermark applications. In *Pacific Rim Workshop on Digital Steganography, STEG '02*, Kyushu Institute of Technology, Tobata, Kitakyushu, Japan, July 2002.

[43] Neil F. Johnson, Zoran Duric, and Sushil Jajodia. *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures*. Kluwer Academic Publishers, 2000.

[44] Neil F. Johnson and Sushil Jajodia. Steganalysis: The investigation of hidden information. In *IEEE Information Technology Conference*, pages 113–116, September 1998.

[45] Ton Kalker, Jean-Paul Linnartz, Geert Depovere, and Maurice Maes. On the reliability of detecting electronic watermarks in digital images. In *Proceedings of the 9th European Signal Processing Conference, EUSIPCO '98*, pages 13–16, Island of Rhodes, Greece, September 1998.

[46] Stefan Katzenbeisser and Fabien A. P. Petitcolas. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, December 1999.

[47] Auguste Kerckhoff. La cryptographie militaire. *Journal des sciences militaires*, 9:5–38, January 1883.

[48] Jong Ryul Kim and Young Shik Moon. A robust wavelet-based digital watermark using level-adaptive thresholding. In *Proceedings of the 6th IEEE International Conference on Image Processing, ICIP '99*, page 202, Kobe, Japan, October 1999.

[49] Edward Kimber. Spread spectrum systems. online presentation, 1999.

[50] Mei Kobayashi, editor. *Wavelets and their Applications: Case Studies*. SIAM, 1998.

[51] Deepa Kundur. Improved digital watermarking through diversity and attack characterization. In *Proceedings of the ACM Workshop on Multimedia Security '99*, pages 53–58, Orlando, FL, USA, October 1999.

[52] Martin Kutter, Sviatoslav Voloshynovskiy, and Alexander Herrigel. Watermark copy attack. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971, San Jose, CA, USA, January 2000.

[53] Yeuan-Kuen Lee and Ling-Hwei Chen. An adaptive image steganographic model based on minimum-error LSB replacement. In *Proceedings of the Ninth National Conference on Information Security*, pages 8–15, Taichung, Taiwan, 1999.

[54] Jacques Levy-Vehel and Anne Manoury. Wavelet packet based digital watermarking. In *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000.

[55] A.S. Lewis and G. Knowles. Image compression using the 2-D wavelet transform. *IEEE Trans. on Image Process.*, 1(2):244–250, April 1992.

[56] Eugene T. Lin and Edward J. Delp. A review of fragile image watermarks. In *Proceedings of the ACM Workshop on Multimedia Security '99*, pages 25–29, Orlando, FL, USA, October 1999.

[57] Jean-Paul Linnartz, Ton Kalker, Geert Depovere, and Rob Beuker. A reliability model for the detection of electronic watermarks in digital images. In *Proceedings of the 5th Symposium on Communications and Vehicular Technology*, pages 202–209, Enschede, The Netherlands, 1997.

[58] Jean-Paul Linnartz and Marten van Dijk. Analysis of the sensitivity attack against electronic watermarks in images. In David Aucsmith, editor, *Information hiding: second international workshop*, volume 1525 of *Lecture notes in computer science*, pages 258–272, Portland, OR, USA, April 1998. Springer Verlag, Berlin, Germany.

[59] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1997.

[60] Peter Meerwald. Digital image watermarking in the wavelet transform domain. Master's thesis, Department of Scientific Computing, University of Salzburg, Austria, January 2001.

[61] A.J. Menezes, P.v. Oorschot, and S.A. Vanston. *Handbook of Applied Cryptography*. CRC Press, October 1996.

[62] Hirofumi Muratani. A collusion-secure fingerprinting code reduced by chinese remaindering and its random error resilience. In *Proceedings of the 4th Information Hiding Workshop '01*, Portland, OR, USA, April 2001.

[63] T.Q. Nguyen. A tutorial on filter banks and wavelets. In *27th International Conference on Digital Signal Processing*, June 1995.

[64] Athanasios Nikolaidis, Sofia Tsekeridou, Anastasios Tefas, and Vassilios Solachidis. A survey on watermarking application scenarios and related attacks. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '01*, Thessaloniki, Greece, October 2001.

[65] Saibal Kumar Pal, P. K. Saxena, and S. K. Muttoo. Smart steganographic applications. In *Pacific Rim Workshop on Digital Steganography, STEG '02*, Kyushu Institute of Technology, Tobata, Kitakyushu, Japan, July 2002.

[66] Fabien A. P. Petitcolas and Ross J. Anderson. Weaknesses of copyright marking systems. In *Multimedia and Security Workshop at the 6th ACM International Multimedia Conference*, pages 55–61, Bristol, England, 1998.

[67] Fabien A. P. Petitcolas and Ross J. Anderson. Evaluation of copyright marking systems. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems '99*, volume 1, pages 574–579, Florence, Italy, June 1999.

[68] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In David Aucsmith, editor, *Information hiding: second international workshop*, volume 1525 of *Lecture notes in computer science*, Portland, OR, USA, April 1998. Springer Verlag, Berlin, Germany.

[69] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding - a survey. *Proceedings of the IEEE*, 87(7):1062–1078, July 1999.

[70] Birgit Pfitzmann. Information hiding terminology - results of an informal plenary meeting and additional proposals. In Ross Anderson, editor, *Information hiding: first international workshop*, volume 1174 of *Lecture Notes in Computer Science*, pages 347–350, Cambridge, UK, May 1996. Springer Verlag, Berlin, Germany.

[71] Dwayne Phillips. Steganography: Hiding information in plain sight. *The C/C++ Users Journal*, pages 49–59, November 1998.

[72] Christine I. Podilchuk and Wenjun Zeng. Digital image watermarking using visual models. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *Proceedings of the 2nd SPIE Human Vision and Electronic Imaging Conference*, volume 3016, pages 100–111, June 1997.

[73] Christine I. Podilchuk and Wenjun Zeng. Perceptual watermarking of still images. In *IEEE Workkshop on Multimedia Signal Processing*, Princeton, New Jersey, USA, June 1997.

[74] Christine I. Podilchuk and Wenjun Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications, special issue on Copyright and Privacy Protection*, 16(4):525–539, May 1998.

[75] David Pollen. Parametrization of compactly supported wavelets. Technical report, Aware Inc., USA, 1989.

[76] A. Pommer and A. Uhl. Wavelet packet methods for multimedia compression and encryption. In *Proceedings of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 1–4, Victoria, Canada, August 2001. IEEE Signal Processing Society.

[77] A. Pommer and A. Uhl. Selective encryption of wavelet packet subband structures for obscured transmission of visual data. In *Proceedings of the 3rd IEEE Benelux Signal Processing Symposium (SPS 2002)*, pages 25–28, Leuven, Belgium, March 2002. IEEE Benelux Signal Processing Chapter.

[78] A. Pommer and A. Uhl. Selective encryption of wavelet packet subband structures for secure transmission of visual data. In J. Dittmann, J. Fridrich, and P. Wohlmacher, editors, *Multimedia and Security Workshop, ACM Multimedia*, pages 67–70, Juan-les-Pins, France, December 2002.

[79] L. Prasad and S.S. Iyengar. *Wavelet Analysis with Applications to Image Processing*. CRC Press, 1997.

[80] J. Schneid and S. Pittner. On the parametrization of the coefficients of dilation equations for compactly supported wavelets. *Computing*, 51:165–173, May 1993.

[81] B. Schneier. *Applied cryptography (2nd edition): protocols, algorithms and source code in C*. Wiley Publishers, 1996.

[82] Sandy Shaw. Overview of watermarks, fingerprints, and digital signatures. Technical report, University of Edinburgh, UK, 1999.

[83] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers, 1996.

[84] Harold Stone. Analysis of attacks on image watermarks with randomized coefficients. Technical report, NEC Research Institute, NJ, USA, May 1996.

[85] Jonathan K. Su. Data embedding and digital watermarking, 1999.

[86] D. Taubman and M.W. Marcellin. *JPEG2000 — Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[87] David B. H. Tay. A class of lifting based integer wavelet transform. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '01*, Thessaloniki, Greece, October 2001.

[88] Min-Jen Tsai, Kuang-Yoo Yu, and Yi-Zhang Chen. Wavelet packet and adaptive spatial transformation of watemark for digital image authentication. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '00*, Vancouver, Canada, September 2000.

[89] A. Uhl. Image compression using non-stationary and inhomogeneous multiresolution analyses. *Image and Vision Computing*, 14(5):365–371, 1996.

[90] S.E. Umbaugh. *Computer Vision and Image Processing*. Prentice-Hall, 1999.

[91] Sviatoslav Voloshynovskiy, Shelby Pereira, and Thierry Pun. Watermark attacks. In *DFG V3D2 Watermarking Workshop 1999*, Erlangen, Germany, October 1999.

[92] Sviatoslav Voloshynovskiy, Shelby Pereira, Thierry Pun, Jonathan K. Su, and Joachim J. Eggers. Attacks and benchmarking. *IEEE Communication Magazin*, 2001.

[93] Steve Walton. Image authentication for a slippery new age. *Dr. Dobb's Journal*, (229):18–26, April 1995.

[94] Houng-Jyh Wang, Yi-Liang Bao, C.-C. Jay Kuo, and Homer Chen. Multi-threshold wavelet codec (MTWC). Technical report, Derpartment of Electrical Engineering, University of Southern California, Los Angeles, CA, USA, Geneva, Switzerland, March 1998.

[95] Houng-Jyh Wang and C.-C. Jay Kuo. High fidelity image compression with multithreshold wavelet coding (MTWC). In *SPIE's Annual meeting - Application of Digital Image Processing XX*, San Diego, CA, USA, August 1997.

[96] Houng-Jyh Wang and C.-C. Jay Kuo. Watermark design for embedded wavelet image codec. In *Proceedings of the SPIE's 43rd Annual Meeting, Applications of Digital Image Processing*, volume 3460, pages 388–398, San Diego, CA, USA, July 1998.

[97] Y. Wang, J. F. Doherty, and R. E. Van Dyck. A wavelet-based watermarking algorithm for copyright protection of digital images. *IEEE Transactions on Image Processing*, 11(2):77–88, February 2002.

[98] M.V. Wickerhauser. *Adapted wavelet analysis from theory to software*. A.K. Peters, Wellesley, Mass., 1994.

[99] Min Wu, Scott A. Craver, Edward W. Felten, and Bede Liu. Analysis of attacks on SDMI audio watermarks. In *Proceedings of the 2001 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, UT, USA, May 2001.

[100] Xiang-Gen Xia, Charles G. Boncelet, and Gonzalo R. Arce. Wavelet transform based watermark for digital images. *Optics Express*, 3(12):497, December 1998.

[101] H. Zou and Ahmed H. Tewfik. Parametrization of compactly supported orthonormal wavelets. *IEEE Transactions on Signal Processing*, 41(3):1423–1431, March 1993.

Curriculum Vitae

Werner Michael Dietl

wdietl@yahoo.com
http://student.cosy.sbg.ac.at/~wdietl/

Grazer Bundesstrasse 13A
5023 Salzburg
Austria / Europe

Education

October 2001 – Present *Applied Computer Science and Business*
Salzburg University, Austria
Research into Watermarking in the Wavelet domain.

September 2002 *Summerschool in Beijing and Shanghai, People's Republic of China*
Four weeks of education about chinese economy and history. Classes at local universities and field trips to joint-venture companies and historic sites.

August 1999 – August 2000 *Master of Science in Computer Science*
Bowling Green State University, OH
Graduate education in computer science with special interest in computer networks. Implemented a distributed computing system with Java Servlets and Java Applets as Masters project. Specific emphasis on how to distribute prime number testing and creation over a client and a server. Also produced a simple general computing server to which the client can upload any Java code that implements a simple interface. The server executes the code and notifies the client when the result is available.

October 1996 – August 1999 *Applied Computer Science and Business*
Salzburg University, Austria
Broad undergraduate and graduate education in applied computer science with a minor in business. Very good theoretical and practical education in mathematics, software engineering and project management. Special interest in distributed and parallel computing. Learned about Internet standards and development methods.

1991 – 1996 Polytechnical school for electronics and informatics, Salzburg, Austria.

Experience

October 2000 – July 2001 *Software Engineer*, Synapta Corporation, Palo Alto, CA.
Worked on server-side Java for multiple startup websites. All development involved web-based N-tier architectures with an application server as the middle-tier and a relational database as the backend. Went through the complete project lifecycle of an order management system for Atomic 29 (`http://www.atomic29.com/`) a business-to-business portal for printed circuitry manufacturers. Assisted in the framework design and database schema

development. Responsible for the manufacturer module for the management of and communication with the customers. Used Java Servlets, Oracle 8, MS SQL Server and JRun.
Worked with a large corporation on the final stage of the product rollout of a large Java Swing application that connects to legacy systems.
Researched Lightweight Software Methodologies and gave presentations about Extreme Programming.

August 1999 – August 2000 *Research Assistant*, Bowling Green State University, OH.
Maintained and improved a large information system for middle high German. Designed and implemented a database for German movies. Both systems used Java Servlets with Apache Tomcat/Jakarta as Servlet Engine, Oracle 7.3 as database and IRIX as OS. Developed the database schema and administered the Oracle database. Implemented data access, query and entry methods. Created a report module to produce a complete book using LaTeX.
See `http://mhdbdb.bgsu.edu/` for results.

November 1997 – September 2000 *Software Engineer*, SBS Software Ges.m.b.H., Austria.
Designed and implemented the hardware control layer for self-service machines using WOSA-XFS and the PC smartcard interface. Developed a large C++ project under an ISO 9000 software process to ensure the banking industries high quality requirements. Possess in-depth knowledge of WOSA/XFS, the PC smartcard interface and ISO 7816, the specification for chipcards. Experience completing large software development projects in a high-quality environment under aggressive deadlines.

Fall 1997 & 1998 *Tutor*, Programming III, Salzburg University.
Helped other students in learning the programming languages Haskell and Prolog. Created a script for the lecture using LaTeX.

July – September 1997 *Software Engineer*, Siemens AG Austria / PSE.
Developed a Windows NT service using Visual C++ and an interface to MS Access using Visual C++ and Visual Basic.

Summary of Skills

Programming Languages: Java, C++, Perl, Ada95, Haskell, Prolog, Simula, Intel Assembly Language

Operating Systems: Unix systems (e.g.: Linux, SGI Irix, Sun Solaris), Microsoft systems including Windows 98, NT 4 and Windows 2000

Applications: Unix tools (e.g.: Emacs, vi, LaTeX, gcc, flex, bison, cvs), Microsoft Office, MS Visual Studio, MS Visual Source Safe, Apache Web Server (httpd, Tomcat, JServ), JRun, Oracle DBMS, MS SQL Server

Other Software: Java Servlets, JDBC, Java 3D, Java Swing, HTML, XML, XSLT, CSS, JavaScript, PHP, CGI, SQL, PVM, OpenMP, Java RMI, WOSA/XFS, smartcards, PC/SC, ISO 7816

Natural Languages: German, English