OULUN YLIOPISTO
UNIVERSITY of OULU

Benchmarking and Modernizing Print-Cam Robust Watermarking Methods on Modern Mobile Phones

University of Oulu
Department of Information Processing Science
Master's Thesis
Anu Pramila
15.05.14

Abstract

Digital watermarking has generally been used for copy- and copyright protection, authentication and, more recently, value-added services. It has been restricted to digital world but by developing methods which are able to read a watermark from a printed image with a camera phone would enable the watermark extraction to be free of time and place.

The aim of this work was to compare print-cam robust watermarking methods and illustrate the problems that emerge when the watermarking methods are implemented on a camera phone. This was achieved by selecting three print-cam robust watermarking methods and implementing them on a camera phone. The robustness of the methods was tested and the results reported.

The obtained results showed that some changes are required to make the earliest proposed watermarking methods work on a modern camera phone. The robustness of the methods was in line with the intended applications as well as camera phone specifications.

Keywords
Camera phones, Android, print-cam robust watermarking

Foreword

I would like to thank my supervisor Antti Juustila for reviews and advices during the writing of this thesis. I also want to thank my thesis opponent Henrik Hedberg for comments and feedback of my thesis. Your comments were invaluable in improving this thesis.

I would also like to thank Valtteri for being there for me and listening my complaining during the studies and dragging me outside from my computer once in a while. Special thank you also to my parents and brother for understanding and time.

I never thought that I would even finish these studies but I did. Thank you all of my friends and colleagues who believed in me even though I was ready to give up.

This is my last master's thesis. I promise.

Anu Pramila

Oulu, May 14, 2014

Contents

List of abbreviations and symbols

AD/DA	Analog to Digital / Digital to Analog
ADK	Android Development Kit
API	Application Programming Interface
BER	Bit Error Rate
DRM	Digital Rights Management
GUI	Graphical User Interface
HVS	Human Visual System
JND	Just Noticeable Difference
JNI	Java Native Interface
JPEG	Joint Photographic Experts Group
MSE	Mean Square Error
NDK	Native Development Kit
NTT	Nippon Telegraph and Telephone
PSNR	Peak Signal to Noise Ratio
PSPNR	Peak Signal to Perceptible Noise Ratio
QR	Quick Response
RAM	Random Access Memory
SDK	Standard Development Kit
TIFF	Tagged Image File Format
VGA	Video Graphics Array

1. Introduction

From the first camera phones in the beginning of the century, mobile phones have evolved into mobile pocket computers. The amounts of megapixels have risen from 0.1 to maximum of 41 with zoom and autofocus features as well as big colourful touch-screens to ease usage. (Hill, 2011.) This development of technology has given people the possibility to capture and share their photos with speed, connect to Internet where ever they are and keep all their content and appointments organized in the mobile phone memory.

However, advancement of technology introduces also threats, such as piracy and non-authorized use of content. The concept of digital image watermarking was developed in the beginning of 1990's in order to fight illegal copying and distribution of digital images. (Cox, Miller and Bloom, 2002.) From then onwards, watermarking has been used for various application areas, including fraud detection, metadata embedding and value-added services.

Digital image watermarking is about embedding information in images in such a way that the information is not visible to human eye but a computer can read it. In digital image watermarking, one special case are the print-cam robust watermarking methods. These methods are robust to process of first printing and then capturing the watermarked image with a digital camera or a camera of a phone. This means that the watermark should be robust to multiple difficult attacks, most notably synchronization losses due to the 3D transformations and image compression (Pramila, Keskinarkaus and Seppänen, 2007).

These print-cam robust watermarks can be used as their predecessors, for example, for carrying copyright information, but they can also be used for linking analog content to digital, for example, printed images in a newspaper to content in Internet. This offers freedom of time and place of watermark extraction and freedom of image format ranging from purely digital images to digital displays and printed materials. The inherited feature of invisibility creates new application areas that are not possible for more conventionally used barcodes.

Only few print-cam robust watermarking methods have been proposed in the history (Nakamura, Katayama, Yamamuro and Sonehara, 2004; Kim, Lee and Seo, 2006; Takeuchi, Kunisa, Tsujita and Inoue, 2005; Pramila, Keskinarkaus and Seppänen, 2008; Pramila, Keskinarkaus and Seppänen, 2012). However, mobile phones have evolved during the last few years dramatically and therefore many of the proposed methods may be already outdated.

The purpose of this thesis is to study print-cam robust watermarking methods and possible implementation issues. The issues rise from the fact that the first of the print-cam robust watermarking algorithms have been developed nearly a decade earlier and camera phones have evolved tremendously since then. One of the first print-cam robust watermarking systems was jointly proposed by Nakamura et al. (2004) and Katayama, Nakamura, Yamamuro and Sonehara (2004). Their work was implemented on i-appli phone with a camera with a resolution on 288x352 (Katayama et al., 2004.) Takeuchi et al. (2005) applied a camera phone with VGA (Video Graphics Array) resolution and

Pramila et al. (2008) had a 2 megapixel camera and later (2012) a 5 megapixel camera phone.

Variation between camera phones used in the research publications and the fact that print-cam robust watermarking is relatively little researched area, makes comparing of different watermarking algorithms difficult. In addition, it is difficult to evaluate from the research papers if the algorithms are applicable and competitive in modern mobile phones. In this work, my aim to address these issues.

The research problem in this study is thus formulated as a question of how the development of camera phones has affected watermarking methods, are the first print-cam robust watermarking methods still valid as such, and are there significant performance differences between methods. These questions are further analysed in Chapter 3.

In order to address the questions, a research application was designed and implemented for a mobile phone. This was used as a basis for watermark implementations and will later be used for demo purposes and research of new methods. For this artefact, design science was used as a research method for guidance.

Later, three watermarking algorithms were selected from literature and implemented for the mobile phone. Due to the evolution of mobile phones and cameras all the algorithms could not be implemented as such. The changes that were required are reported and robustness of the algorithms after these changes tested and compared.

The contribution of this work is therefore two folded. First, it works as a comparative research of different print-cam robust watermarking methods. Second, it illustrates the problems that must be faced when a print-cam robust watermarking system is implemented on a camera phone. The products of this research are a system for testing different watermarking algorithms on the device and insight to print-cam robust watermarking methods at the moment.

In the next chapter, some terminology and background for digital image watermarking is explained and motivation for the work is build by discussing some applications and scenarios. Chapter 3 illustrates the research problems and discusses methods for conducting the research. Chapter 4 explains the watermarking methods that were used in this research with more detail. Chapter 5 shows the design science part of the research and Chapter 6 shows the comparative analysis of the watermarking methods. Chapter 7 contains summary and conclusion.

2. Digital Image Watermarking

In this chapter, digital image watermarking and concept of print-cam robustness are explained through literature. In addition, some watermarking applications and their requirements are discussed as a background and motivation for this work.

2.1 Applications

The first electronic watermarking algorithms were proposed as early as in the 1950's but the research on the field did not grow a lot until 1990's (Cox & Miller, 2002). Watermarking gained publicity in 1997 when a need for broadcast monitoring systems emerged. A scandal broke in Japan where television stations had overbooked air time and advertisers were paying for commercials that were never aired. (Cox, Miller & Bloom, 2000.) With watermarking, the advertisers could monitor their commercials by themselves so that the commercials are all aired according to contract without modifications.

Even though broadcast monitoring was one of the first application areas for watermarking, copy- and copyright protection was another that raised discussion. As e-commerce became more popular, content providers needed ways to battle against piracy and misuse of the data. Several DRM (Digital Rights Management) systems were developed and introduced. (Hartung & Ramme, 2000.) However, as DRM started to disappear from publicity as more and more of the content providers abandoned their use of DRM (Suehle, 2011), watermark researchers turned towards other application areas and the idea of copy- and copyright management with watermarking was largely moved aside.

More and more work has been put into watermarking in authentication applications. In many cases it is important to be able to tell if the image or other work has been altered in any way. One way of doing this is to use fragile watermarking, in which the watermark is destroyed if the content is changed. (Cox et al., 2000.)

Often when watermarking is discussed, steganography is also mentioned. Steganography differs from watermarking in that in steganography even the fact that something is embedded in the content is kept secret. Steganography is thus used in covert communication. Another factor that differentiates watermarking and steganography is robustness. Steganographic systems are rarely robust to any attacks, intentional or unintentional, whereas watermarking systems are usually robust to at least some unintentional signal processing attacks. (Hartung & Kutter, 1999.)

In the opposite side of the watermarking applications spectrum from steganography are the value-added watermarking systems. The idea of value-added watermarking is to offer users extra value, such as a freebie or some special information, i.e. something beneficial to the user so that he/she does not want to remove the watermark intentionally but that the user is aware of the existence of the watermark. Value-added watermarking systems generally are thus highly robust to unintentional attacks. (Cox & Miller, 2002.)

There are, therefore, multitude of applications for watermarking but it is not practically possible to build one watermarking system for all. In this work, we shall focus on value-added watermarking and robustness of the methods.

2.2 Basic watermarking algorithm

In the following chapters we shall discuss only image watermarking. Watermarking of audio, video or other content, is beyond the scope of this research. In digital images, the watermark can be embedded either in spatial domain, in some transform domain or, depending of the method, on multiple domains. The most common way is to divide colour image into colour and luminance components and then proceed in watermarking only to luminance component (Hanjalic, Langelaar & van Roomalen, 2000). However there are some methods that use colour space, i.e., Reed and Hannigan (2002).

The watermark signal is often a bit sequence of ones and zeros, possibly minus ones. The signal that is embedded in the image, is the message itself or a pseudorandom sequence, in which the message is encoded with a key, that does not correlate with the image content. The most basic method of embedding a watermark in an image is thus to add the signal to the luminance values of the image. (Hanjalic et al., 2000.) This generic method is illustrated in Figure 1.

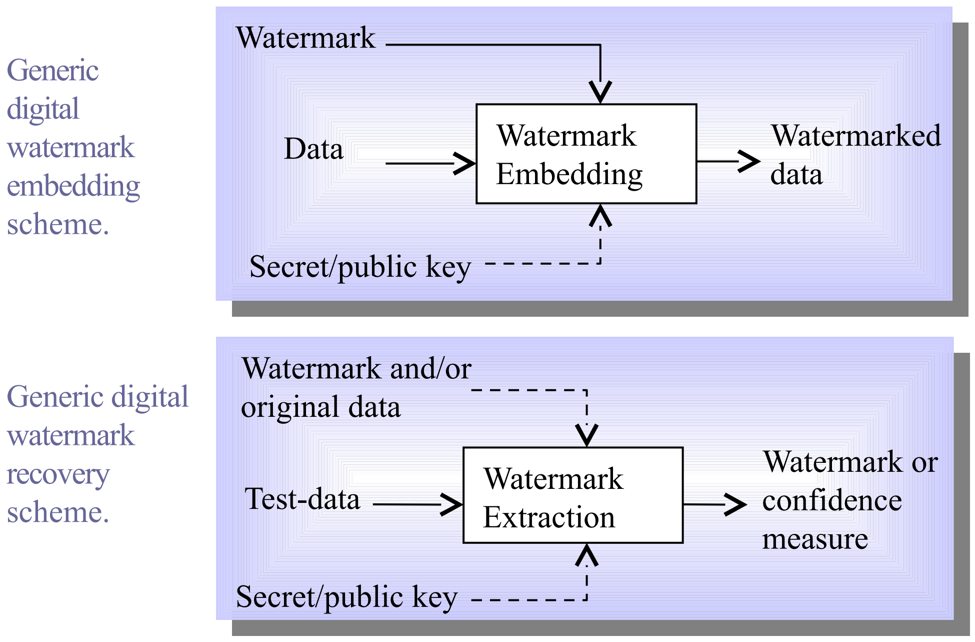

Generic digital watermark embedding scheme.
Watermark
Data
Watermark Embedding
Watermarked data
Secret/public key
Generic digital watermark recovery scheme.
Watermark and/or original data
Test-data
Watermark Extraction
Watermark or confidence measure
Secret/public key

Figure 1: General watermark embedding and extracting schemes

The generic method can also be formulated as

$$I_w(x,y) = I(x,y) + kW(x,y) \tag{1}$$

,where the watermarked image $I_w(x,y)$ is obtained by adding the original image $I(x,y)$ the watermark sequence $W(x,y)$ with an embedding strength k, also known as gain factor. The watermark can be extracted with cross-correlation

$$R_{I,W}(i,j)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} I_W'(m,n)*W(m+i,n+j) \tag{2}$$

between the received watermarked image $I_w'(x,y)$ and the pseudorandom sequence $W(x,y)$. If the cross-correlation result, $R_{I,W}$, exceeds some predefined threshold, then watermark is detected. (Hanjalic et al., 2000.)

However, this method compensates embedding and extracting of only one bit of information, i.e. test if watermark exists or not. In order to embed multiple bits, the data could be embedded for example in blocks. (Hanjalic et al., 2000.)

2.3 Watermark requirements

Watermarking methods are evaluated by three properties: imperceptibility, robustness and capacity. All of which are characterizing for a watermarking system but none can be maximized simultaneously without affecting others. As is illustrated in Figure 2 there will be a tradeoff between the properties. The values for each property depend heavily on the application. For example, in value-added watermarking high robustness is required. Therefore the method cannot have high capacity nor perfect imperceptibility. If also imperceptibility need to be maxed, then capacity is by necessity small.

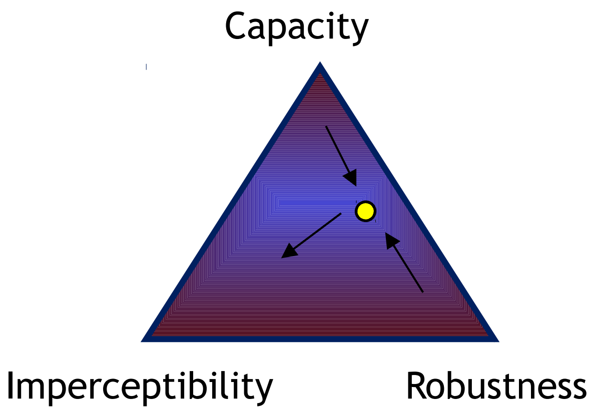


Figure 2: The tradeoff between watermark properties.

In the context of watermarking, capacity is the amount of bits that can be hidden in the data, imperceptibility governs how well the data is hidden and whether the data insertion is perceptible to human senses. The robustness includes how well the watermark survives intentional and unintentional attacks before being destroyed. A fourth property, security, can also be considered depending on the application and it is closely linked to all of the aforementioned properties. The security aspect is needed when the watermark should be able to resist intentional attacks aiming at watermark destruction or unauthorized access. In the next subchapters, these properties are further discussed from the point of view of digital image watermarking.

2.3.1 Imperceptibility

Imperceptibility varies from fully visible watermarks to steganographic applications in which it is desirable that only the sender and receiver are or will be aware of the message. The visible watermarks are useful in showing without delays that the image is copyrighted and the visible watermarks work in many ways similarly to paper watermarks. However, they are relatively easy to crop or the image can be modified to remove the watermark in addition to the fact that the visible watermark may disturb the aesthetics of the original artwork. It might be more advantageous to use an invisible watermark that cannot be removed without destroying the content.

As mentioned earlier, the most general way of embedding a watermark is to choose a constant for embedding strength and embed the watermark with this constant strength over the image (Hanjalic et al., 2000). However, the constant embedding strength means that the watermark can be visible in parts of the image with plenty of room for watermark in others. By taking into account the original image and human visual system, the watermark should be embedded with variable strength ensuring that the watermark is not visible anywhere in the image and that it still is embedded as strongly as possible in order to maximize robustness.

Chou and Li (1995) noted that there are primarily two factors that affect the error visibility threshold of each pixel. These are average background luminance and spatial non-uniformity of the background luminance. Human visual system (HVS) is more sensitive to luminance contrast instead of absolute luminance value and noise on the dark areas of the image tend to be less visible. These facts are better known as Weber fraction. In addition disturbance on highly textured regions are less visible.

Based on their observations, Chou and Li (1995) developed a JND (Just Noticeable Difference) model for watermarking. An example of the results are shown in Figure 3 in which the leftmost image is the original image and the rightmost image illustrates the JND measurements. In the JND image, the lighter the area the stronger the watermark can be embedded. In other words, it is possible to see from the images that it is preferable to embed the watermark more strongly in dark areas, such as hair and mirror edge, and in the textured regions such as the feather whereas the embedding strength should be lower on smooth areas, for example, on the shoulder to prevent visibility of the watermark.

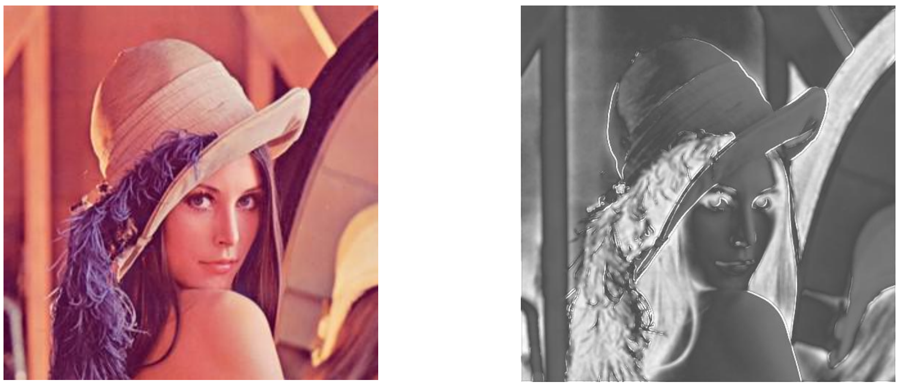


Figure 3: JND (Just Noticeable Difference) by Chou and Li

Imperceptibility is still highly subjective feature and difficult to measure accurately. The only acceptable methods for testing imperceptibility are user tests in which users are told to compare unwatermarked and watermarked images in order to find out if the watermark is truly invisible (Cox et al., 2002). These methods are, however, time consuming and there is always question if the tests are fully comparable with other user tests as the test settings may vary.

Objective measures for imperceptibility would be more convenient to use but nonexistent. The most widely used measure is PSNR (Peak Signal to Noise Ratio) which is derived form MSE (Mean Squared Error):

$$MSE=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-I_w(i-j)]^2$$
$$PSNR=10\cdot\log_{10}\left(\frac{255^2}{MSE}\right) \qquad (1)$$

PSNR calculates pixel difference across whole image but does not take into account JND. PSNR assumes that changes in each pixel are equally disruptive whereas JND models take advantage of the fact that they are not equal by changing other pixels more than the others. Thus watermarking methods that employ JND tend to get worse results from PSNR than those methods that do not use JND even though JND generally improves imperceptibility factor. PSNR thus does not accurately represent imperceptibility of the watermark (Chou and Li, 1995).

Chou and Li (1995) proposed a PSPNR (Peak Signal to Perceptible Noise Ratio) test which depended heavily on the JND model used. This brings us a problem of how to determine PSPNR value if JND model is incomplete. An all-inclusive JND model that accurately correspond to HVS is yet to be proposed.

2.3.2 Robustness

Watermark robustness means how strong attacks the watermark resists before it is destroyed. The preferable situation for robust watermarks would be when the watermark cannot be destroyed without destroying the content as well. On the other end of the spectrum are the fragile watermarks which detect tampering of the image and by design get destroyed very easily.

Attacks against watermarks can be generally divided into intentional and unintentional attacks (Cox, Miller and Bloom, 2002). The unintentional attacks consist of signal processing such as format changes from tiff to jpeg, lens distortions, relatively small geometrical distortions etc. The intentional attacks are launched towards watermarked contents with malicious intents and include collusion attacks, de-noising, remodulation and aforementioned geometrical attacks with sole purpose of destroying the watermark. In this work we will focus on unintentional attacks and especially geometrical attacks encountered during the print-cam process.

Print-scan process

The print-scan process, in which the watermarked image is first printed and then scanned can be considered as a prerequisite for the print-cam process. First of all, the watermarked image is printed which affects the watermark robustness from the beginning. Perry, MacIntosh and Cushman (2002) concluded in their work that printing quality varies across printers with different manufacturers and even across printers with identical models of the same manufacturer. In addition paper quality and ink density vary.

Solanki, Madhow, Manjunath and Chandrasekaran (2004) and later He and Sun (2005) studied the print-scan process and concluded that

1. The low and middle frequency coefficient are preserver better that the high frequency ones.
2. In the low and middle frequency bands, the coefficients with low magnitudes see a higher noise that their neighbours with high magnitudes.
3. Coefficients with higher magnitudes see gain of roughly unity.
4. Slight modifications to the selected high magnitude low frequency coefficients does not cause significant perceptual distortions.
5. Most textures can be preserved.
6. The dynamic range of intensity values is reduced and clipping of low and high values may occur.
7. The distribution of pixel values after the print-scan process looks roughly a spindle.

When an image is scanned, the image is placed on the scanner bed and the scanner reads the image line by line. It is practically impossible to place the image perfectly straight on the scanner, and even though the scanner operates only on two dimensions, it can be enough to destroy the watermark. Therefore, the watermark should be able to resist some rotation and translation, i.e., shift, as well as scaling, AD/DA transformation and noise addition.

There are few different methods for overcoming rotations and other geometrical distortions in print-scan process. One way is to find out what kind of transformations the image has gone through, such as was done by Pereira and Pun (2000). The other way is to employ a special transformation domain, such as Fourier-Mellin domain (O'Rhuanaidh and Pun, 1997).

Print-cam process

The print-cam process, in which a watermarked image is first printed and then captured with a digital camera or a camera of a camera phone, can be considered as an extension of the print-scan process. Rotations, translation, scaling, AD/DA transformation etc. occurring in the print-scan process are all possible attacks in the print-cam process. In addition, the print-cam process inflicts various other attacks on watermark including more severe rotations, lens distortions and lighting variations including reflections.

Pramila et al. (2007) listed several attacks and problems on the print-cam process. The most notable ones are geometrical attacks and especially rotation in 3D space. This is illustrated in Figure 4 in which an image is place on a wall and the captured image shows the effect of the tilt of the optical axis of the camera. Tilt of the optical axis results in loss of synchronization of the watermark because it is difficult to determine in which location the watermark exists in the captured image and in which direction it should be read.

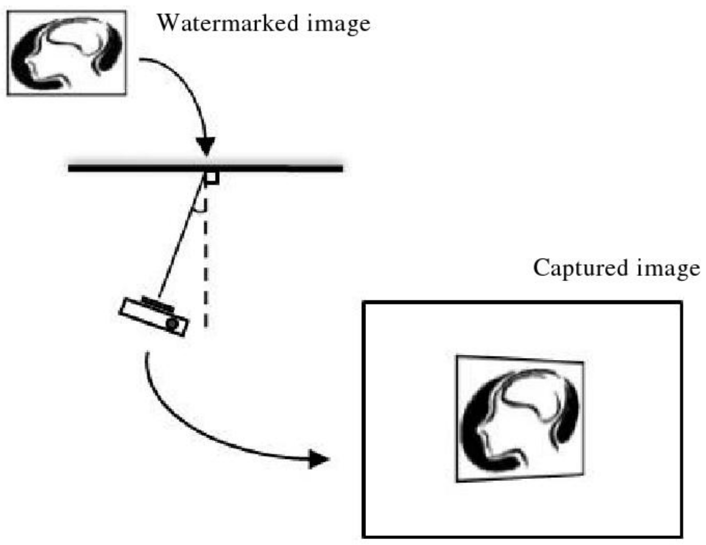


Figure 4: Effect of the tilt of the optical axis of the camera

In addition to loss of synchronization, reading a watermark from a captured image requires that the image is well in focus. It is difficult to read watermark from an unfocused image in the same way it is difficult to recover content of the unfocused image.

When dealing with cameras, the lens distortions must also be considered. Especially pincushion and barrel distortions that bow straight lines in real world to curves. In many cameras nowadays this distortion is negligible for eye but the watermark method must take this into account or the camera should be calibrated.

Nakamura et al. (2004) and Katayama et al. (2004) were among the first to propose a print-cam robust watermarking system. They relied on a frame around the image to correct the geometrical distortions. A frame was also used by Takeuchi et al. (2005) and Pramila et al. (2008). The advantage of the frame, as mentioned by Katayama et al., is that it shows to the user that a watermark is present. However, frame-based methods are vulnerable to cropping and depending on the application the frame is not always desirable.

A different approach was selected by Kim, Lee and Seo (2006) who embedded the message repeatedly in the image in a form of pseudorandom vector. The message was detected from the captured image by calculating autocorrelation function, which showed the lattice form of the embedded message. The message was then read with cross-correlation. However, their results were limited because a tripod was used to minimize amounts of geometrical distortions. In addition the image was resized and cut by hand to its original dimensions.

Autocorrelation was also employed by Pramila et al. (2012) who embedded the message in a form of directed patterns. The message was coded into angles of these patterns and this method had the advantage that is was robust to geometrical distortions, cropping and required to frame around the image. The method is also robust to logos placed on top of the image and does not require the image to be rectangular.

Out of the more recent print-cam robust watermarking techniques Thongkor and Amornraksa (2013) showed a method that was robust to partial glass reflections of top of the captured image. Their method requires the original image in order to align the image and correct geometrical distortions. Their method is also special in that it embeds the message in the blue channel of the image instead of luminance values.

Print-cam robust watermarking methods are few due to the extensive geometrical attacks and other problems that are yet to be to solved. In addition, low processing power in camera phones did not encourage image processing in phones. The modern mobile phones, however, are closer to mobile computers and people are getting accustomed to using their phones for ever increasing amount of tasks. This study, focuses mainly on value-adding watermarking, but this is only on possibility of what print-cam robust watermarks have to offer.

2.4 Scenarios and commercial background

Figures 5 and 6 show some possible user scenarios for the print-cam robust watermarking. The watermark can be embedded in any image and printed on periodical, newspaper, poster etc. The user then reads the message and receives some value from the interaction. The value can be an augmented reality application, a freebie or a sample of music or, in case of security applications, a validity measure or authentication.

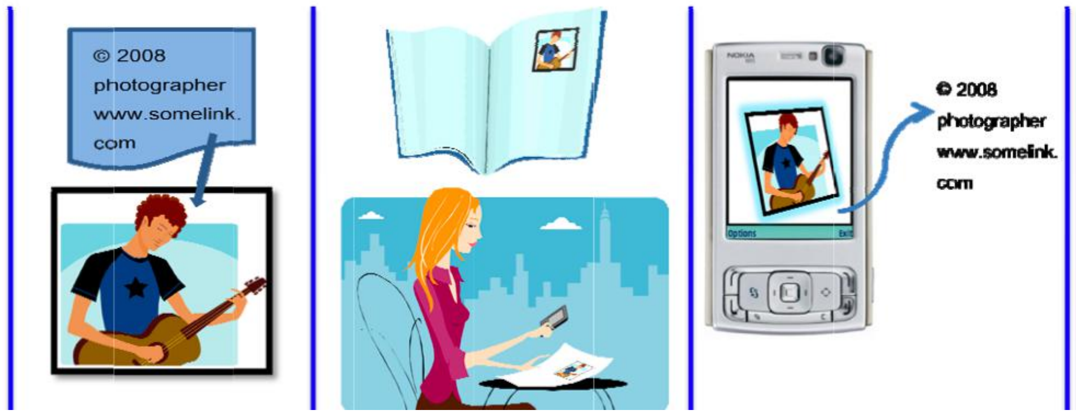


Figure 5: In this scenario, the user sees interesting add which provides a link to band's webpage with a sample of band's top hit music.

The applications for digital watermarking in general were discussed in Chapter 2.1 in order to give introduction and motivation to watermark research in general. However, usage of mobile phones and print-cam robust watermarking systems, makes a whole new area of applications possible and opens new business venues. The usage of watermark would not be anymore restricted to time and place but the watermark could be read anywhere at any time.

Commercially, print-cam robust watermarking has not received much publicity but few instances have been created and tested on market. The most famous watermarking related international company, Digimarc, offers security and brand protection from companies with watermarking as well as new business possibilities with their new Discover application (Digimarc Discover, 2014).

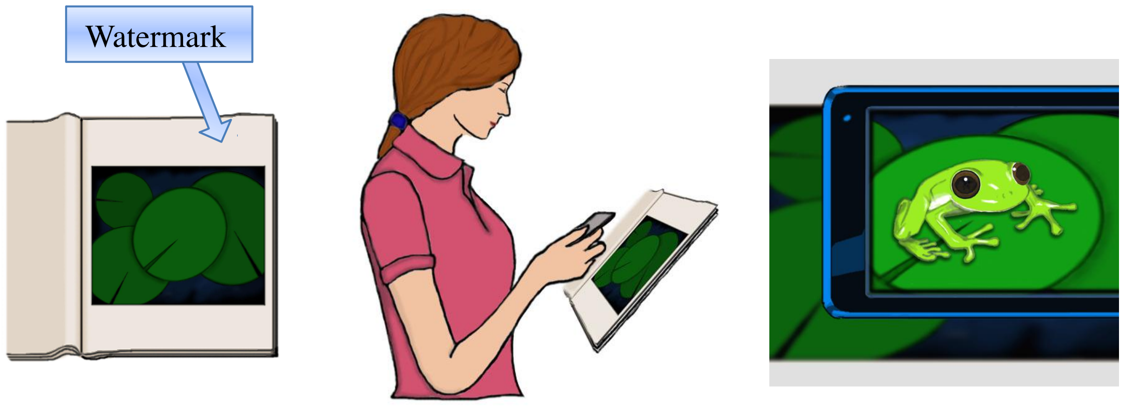


Figure 6: The watermark is embedded in image. The user can read the watermark with a camera phone, and the content is shown according to the watermark information. Here an augmented reality application is launched.

In the beginning of 2000, Digimarc initiated Digimarc MediaBridge, the idea of which was to bridge images on printed materials to digital world and offer value to the user (Perry et al., 2002). This would enable users to get from a newspaper to an Internet URL with one click so that the image would contain a watermark that is read with a camera phone. The phone would then send the information of the watermark to Digimarc registry to determine what to do with it: whether to direct the user to a website or some web based application. In order to use the service, the user was required to download the client software to his/her phone. The Digimarc's solution, at least initially, to the question of how to tell the user that a watermark is present, was to partner with an e-commerce or catalogue company. The users could then select catalogue items with their phones. (Marek, 2014.)

In 2006, Digimarc announced a launch of a digital watermarking demo in Japan in “Amusement Café Maid in Japan”. There, a company called MediaGrid had licensed their technology. The pilot offered customers a possibility to interact with digitally-watermarked print materials with their mobile phones. (Digimarc Mobile E-Commerce Pilot, 2014.)

Later, Digimarc has advertised their Digimarc Discover solutions on their website that is a software capable of reading watermarks from images and music, QR codes and barcodes. Technical details are difficult to find but their research papers indicate to a watermarking technique based on watermark on chromatic channels. Digimarc claims that their system is easy to use, the content is watermarked by using their online system and the users can read the watermark by downloading a Digimarc client application. (Digimarc Discover, 2014.)

NTT (Nippon Telegraph and Telephone Corporation) Cyber Solutions Laboratories developed Cybersquash in mid 2000. Like in Digimarc's MediaBridge and Discover, Cybersquash is embedded in a printed image image and the watermark contains an URL to some website. This watermark was then read with i-appli camera phone.

3. Research problems and methods

In this chapter, research problems are introduced and refined, the research methods are described and limitations to the research acknowledged.

3.1 Research problems

The aim of this research is to study various print-scan robust watermarking methods and see how time affects them, that is, are the methods created a decade ago still valid. The initial idea was to use two research questions:

1. Are print-cam robust watermarking methods camera dependent?

2. What are performance differences between different methods on same hardware?

However, the first question was found out to be imprecise. If two relatively recent camera phones are studied, we can note that they are very similar and the only differences would be processing power, i.e., the methods would only vary in processing times. Naturally camera specifications would also affect to watermark performance, but this would not require a change in the method itself. On the other hand, the research question can be understood to question if the methods have been designed in history to some specific camera phone and if development of camera phones have made the methods obsolete or too difficult to implement for modern phones.

The first research question was therefore reformulated into

RQ1: How the watermarking methods are affected when implemented on a modern mobile phone?

RQ2: Are the first watermarking methods proposed still viable?

The other initial research question was also further specified

RQ3: What are performance differences between the selected three different watermarking methods on same hardware?

3.2 Research methods

In order to answer the research questions, this research is build with two research methods. First, design science research is used in building the testing application, and then, the research if continued with comparative analysis on implemented watermarking methods.

3.2.1 Design science research in building the watermarking applications

In order to accurately conduct the research, an artefact must be build for testing and this can be achieved through design science research. Design science research offers a framework for building artefacts in information systems (March and Smith, 1995). Although, the application developed here has only limited impact as such on information systems research, the artefacts that hopefully stem from this research have clear business purpose. Therefore it is justified and advantageous to apply design science methodology on this artefact.

Hevner et al. (2004) proposed seven guidelines for design science:

1. Design as an Artefact - Design-science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation.
2. Problem Relevance - The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
3. Design Evaluation - The utility, quality, and efficacy of a design artefact must be rigorously demonstrated via well-executed evaluation methods.
4. Research contributions - Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.
5. Research rigour - Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artefact.
6. Design as a Search Process - The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
7. Communication of Research - Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Here the relevant artefact is the program for the mobile phone, the purpose of which is to provide a test tool for various watermarking algorithms. The aim is to create a tool, that has relatively high modularity and portability for future use and development. Especially for demo and research purposes.

Even though some print-cam robust watermarking systems have been proposed, they have been proposed so far apart in time that they have been implemented on very different devices thus making comparison difficult. The contribution of this research is therefore in part to bring these methods on same platform for comparison. In addition, testing of the watermarking methods becomes faster and easier if the code is already in the mobile phone, instead of requiring transfer of images to computer which then calculates the results.

The system is evaluated first by testing the implementation from software defiances and designs. Secondly, the results are roughly compared to the reported performances of respective research publications that are implemented. Large deviations from the reported performances mean unacceptable implementation.

The contribution of this artefact is, as mentioned before, to implement print-cam robust watermarking methods on same mobile phone for comparison and testing. The artefact will be later used for demo purposes and testing of new watermarking methods as well as comparing those methods against old ones. By looking at the literature, there are no researches or studies about print-cam robust watermarking that has brought different methods on same camera phone. The researchers have thus far relied on other publications while comparing the methods to others. This has obvious drawbacks while evaluating the proposed methods because the settings for experiments are hardly exactly the same.

The research rigour is attained, while conducting the research, by using readily available and confirmed methods, design diagrams and models for software design process and software implementation. In addition, the exactness of the implemented watermarking methods is achieved by carefully following the published materials of the methods while implementation. There is no reason to suspect that the peer reviewed publications are not truthfully explained

The building of the artefact is by definition highly iterative process. Although the design of the artefact is relatively simple, as seen in the following chapters, the implementation consists of a series of coding and debugging phases, testing and further coding. The aim of the research is not to blindly implement the watermarking methods and then see if they work, but to see if the algorithms are viable and if something needs to be changed in order for the algorithm to work. An algorithms designed for less than VGA resolutions may not work readily in 8 megapixel cameras.

The research has a strong background on mathematics and engineering sciences. The results of the research, however, must be understood by business people and engineers alike and this will be mediated in the discussion section of this work. The main audience of this work are therefore watermarking researchers as well as students who read this thesis as an example for their studies.

3.2.2 Comparative analysis of watermarking methods

The aim of this research is to compare existing watermarking methods to each other through performance and implementation. The comparison of the methods are done by implementing each of the watermarking methods in a mobile phone and then experimenting with the methods.

The experiments contain comparison of robustness to print-cam process as well as some performance measures. The robustness experiments shall give numerical data for comparison analysis. Unfortunately it is impossible to access exactly the same models of mobile phones that were used in the original watermarking publications. Therefore the data cannot be straight compared with the results reported on respective publications.

The methods that were selected for closer inspection, were methods that did not require user interaction in any phase other that when the image was captured. There are not many print-cam robust watermarking methods and out of those only few can be considered to fulfil this requirement. The research was further limited to colour images because methods for watermarking binary images and holograms are inherently different and thus they cannot be compared without difficulties.

Kim et al. (2006) proposed a method which was limited for using a tripod and required user interaction for cutting the image properly. The work by Takeuchi et al. (2005) is unfortunately unavailable because details of the method are only available in Japanese. The methods used in commercial applications, such as Digimarc Discover (2014), are naturally unavailable for reproduction.

Therefore the suitable methods were the method by Nakamura et al. (2005), Pramila et al. (2008) and Pramila et al. (2012). The methods were compared by calculating their robustness against geometrical distortions, in here, distance and 3D rotation with different resolutions of the camera. In addition speed of the watermark algorithms were compared. The speed gives indication of the complexity of the algorithms. In order to equalize the measurements, the same capacity was selected for each of the methods.

3.3 Limitations

There are some limitations that should be taken into account. First of all, the work is based only on three watermarking methods. Unfortunately, there is nothing to be done for this limitation as print-cam robust watermarking systems are rare.

Second, the test program is build on one specific mobile phone. It is difficult to say whether test results would be drastically different if implemented on a different phone or platform.

4. The watermarking methods

In this chapter, the three watermarking methods that were selected for implementation and comparison are explained.

4.1 Frame based method by Nakamura et al. (2004)

Nakamura et al. proposed one of the first print-cam robust watermarking methods in 2004. This paper was published jointly with Katayama et al. (2004) who proposed a fast method for detecting a frame around the image. Together the methods worked as a print-cam robust watermarking system that was developed and implemented for i-appli camera phones which had, compared to modern mobile phones, low processing power and small resolution camera. Nevertheless their method reportedly run under 2 seconds and was robust to rotations in 3D space.

The watermarking scheme is shown in Figure 7. First the message is repeated *l=N^2/n* times, where *N* is predetermined integer and *n* is the length of the error-coded message, and the obtained bit sequence is modulated with a pseudorandom sequence. The sequence is then scrambled and a 2D pattern is formed from two sine curves with frequency *f*. A 2D pattern is assigned to each of the message bits according to the message value and a larger, image sized, pattern is formed as illustrated in Figure 8. The pattern is embedded in the original image with

$$\begin{gathered} I'_{x,y} = I_{x,y} + aW_{x,y} P_{x,y}, \\ (x=0 \ldots Width-1, y=0 \ldots Height-1) \end{gathered} \tag{1}$$

,where $W_{x,y}$ is a weight matrix (Nakamura et al. 2000).

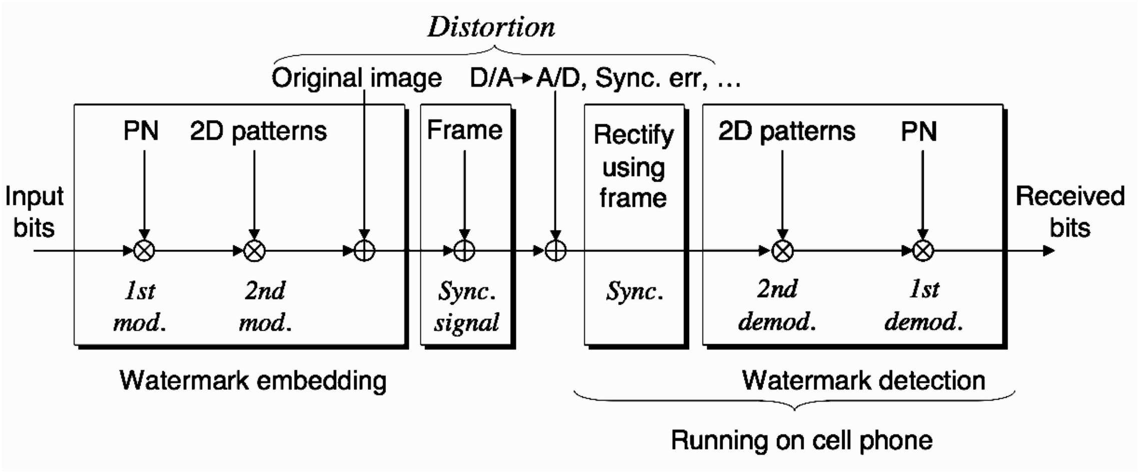


Figure 7: Watermarking scheme by Nakamura et al.

The message is detected with a process in which the image is first scaled to a size determined by the frequency of the 2D pattern used in modulation. The scaled image is then filtered with a 3×3 preprocessing filter and the result is clipped to contain only -1's, 0's and 1's. This obtained result is then divided into NxN individual blocks energy of the frequencies corresponding to the two 2D sine curves is calculated. The energy is calculated with two convolution operators effectively performing demodulation. The difference of these two energies tell the direction of the pattern and in the end the embedded bits. The message is then unscrambled and 1D demodulated. (Nakamura et al., 2004.)

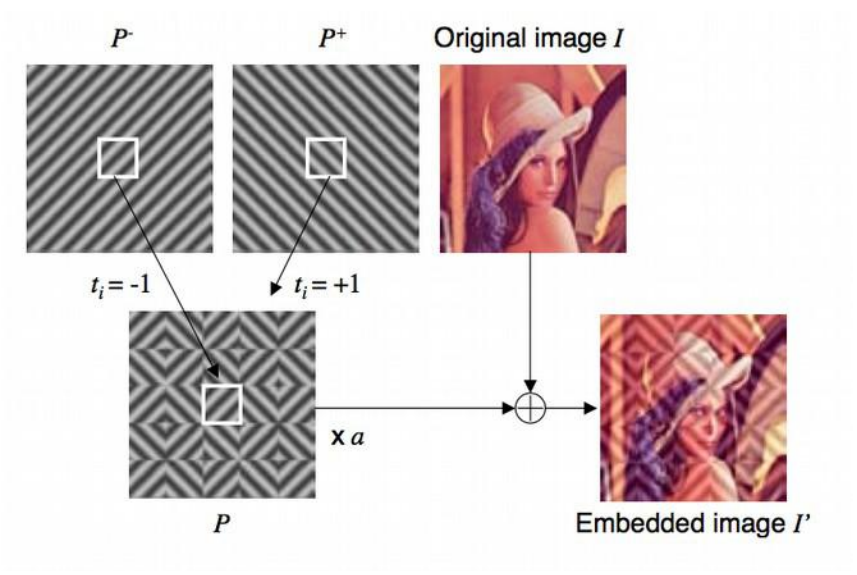


Figure 8: Pattern modulation for method by Nakamura et al.

4.2 Frame based method by Pramila et al. (2008)

The method by Pramila et al. (2008) is also based on frame around a watermarked image. The frame has advantage of showing the user that watermark is present as well as providing a robust way for watermark synchronization.

The message is embedded in wavelet domain (Pramila et al. 2008) by employing spread spectrum techniques. The original image is divided into first level sub-bands of Haar wavelets and the message embedded in the horizontal and/or vertical coefficients. By embedding the watermark in the detail coefficients and thus to the high frequency parts of the image, the imperceptibility of the watermark is better maintained.

The message is embedded with

$$Y'_{l,f}(n) = \begin{cases} Y_{l,f}(n) + \beta \cdot m(k), & messagebit = 1 \\ Y_{l,f}(n) - \beta \cdot m(k), & messagebit = 0 \end{cases} \tag{4}$$

,where $Y_{l,f}(n)$ is the sub-band of Y in the lth resolution level and fth frequency orientation, Y is the original image, β is a scaling coefficient and $m(k)$ is the m-sequence that controls the chip rate for spreading. (Pramila et al. 2008.)

When message is read, the image is wavelet transformed and the transformed coefficients divided into same size as the m-sequence used in embedding process. The segments are the cross-correlated and message read accordingly. (Pramila et al. 2008.)

4.3 Autocorrelation based method by Pramila et al. (2012)

Having a visible frame is not desirable in all applications and therefore it is advantageous to study optional methods. Pramila et al. proposed in 2012 a print-cam robust watermarking method that utilized no frames and instead relied on pseudorandom sequences and autocorrelation function. This enables the watermarked image to be of any size or shape, including a circular image as shown in the journal paper.

The watermark message embedding process is illustrated in Figure 9. The message is embedded in the spatial luminance domain and the image features are taken into account by first calculating JND values for each pixel. That is, a scale of how much each of the pixels can be changed before the change becomes noticeable. These values emphasise embedding strength on the vicinity of details of the image and thus the embedding strength varies across the image by enhancing embedding strength on the areas of the image where the watermark is less perceptible. These JND values are calculated with the method by Chou and Li (1995).

The message itself is Gray coded and quantized into degrees. A pseudorandom sequence is multiplied into a 2D periodic pattern which is then rotated respectively to each of the message degrees. These patterns are then embedded in the image with equation

$$Y_i(x,y)=X_i(x,y)+\delta_1\cdot JND(x,y)\cdot W_i^{\theta i}(x,y)+\delta_2\cdot(1-JND(x,y))\cdot W_i^{\theta i}(x,y) \qquad (5)$$

,where Yi is the *i*th block of the image, Xi is the preprocessed image, and the *Wi*θ represents the coded water- mark information in the form of directed periodic pattern and δ_1 and δ_2 are scaling factors for JND values. *JND(x,y)* is the JND value in pixel (x,y). (Pramila et al. 2012.)

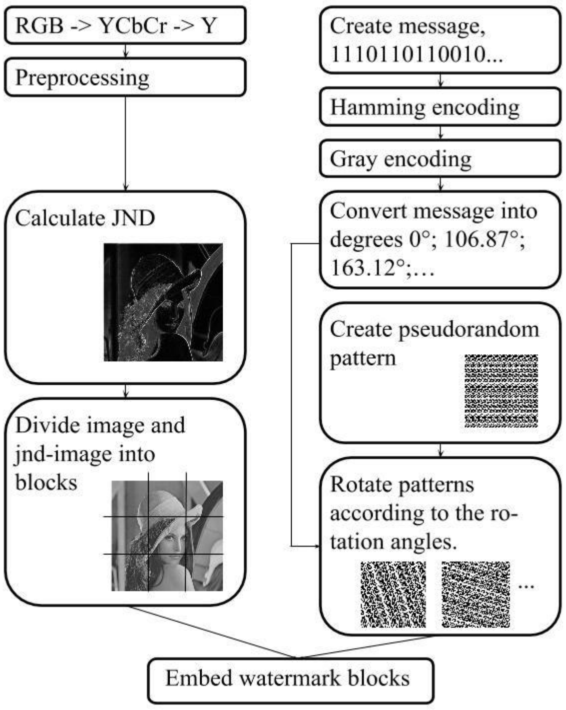

RGB -> YCbCr -> Y
Preprocessing
Calculate JND
Divide image and jnd-image into blocks
Create message, 1110110110010...
Hamming encoding
Gray encoding
Convert message into degrees 0°; 106.87°; 163.12°;...
Create pseudorandom pattern
Rotate patterns according to the ro- tation angles.
...
Embed watermark blocks

Figure 9: Embedding process for Autocorrelation based method by Pramila et al.

The watermark is extracted by first dividing the image into blocks as depicted in Figure 10. Each of the blocks is then processed separately and the direction of each of the patterns is determined. The full watermark reading process is shown in Figure 11. (Pramila et al. 2012.)

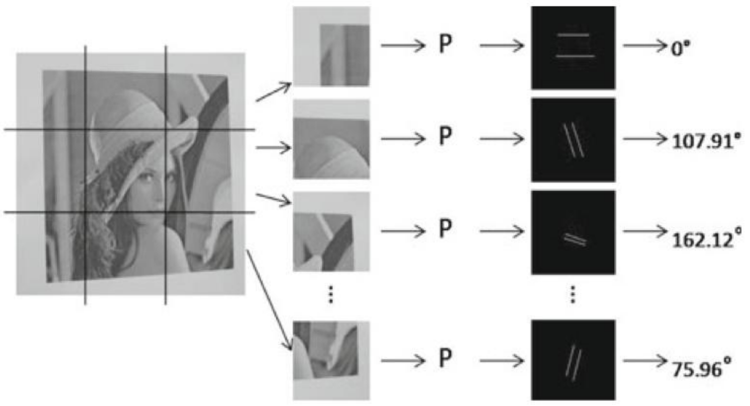


Figure 10: Reading the watermark block by block in the method by Pramila et al.

First each of the blocks is filtered with Wiener filter in order to reduce the effect of the original image from the watermark. Then autocorrelation function is calculated and possible peaks are enhanced with Laplacian of Gaussian filtering and morphological operations. The peaks are aligned according to the direction of the pseudorandom sequence pattern and this alignment is detected with Hough transform and line detection. These detected lines then give the angle of the pattern and thus the message. (Pramila et al. 2012.)

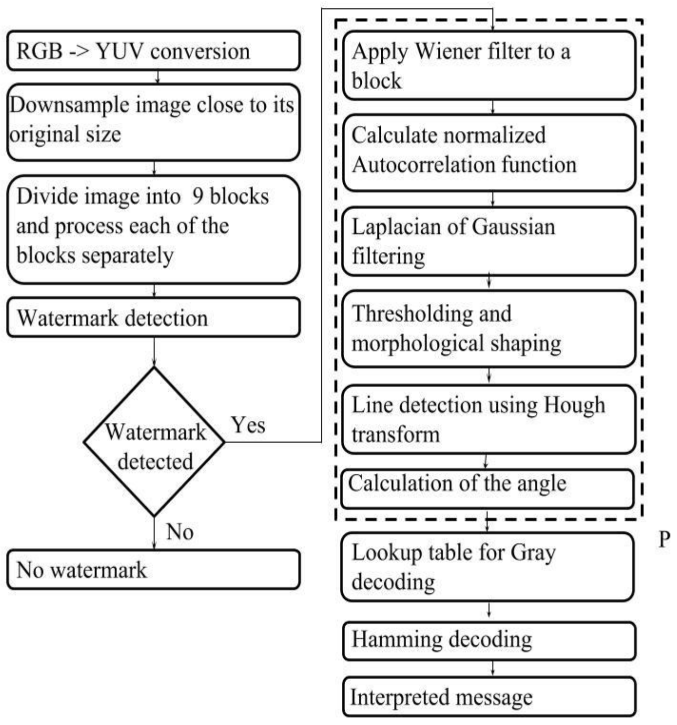

RGB -> YUV conversion
Downsample image close to its original size
Divide image into 9 blocks and process each of the blocks separately
Watermark detection
Watermark detected
Yes
No
No watermark
Apply Wiener filter to a block
Calculate normalized Autocorrelation function
Laplacian of Gaussian filtering
Thresholding and morphological shaping
Line detection using Hough transform
Calculation of the angle
P
Lookup table for Gray decoding
Hamming decoding
Interpreted message

Figure 11: Watermark reading process for method by Pramila et al.

5. Test program – design and implementation of the application

In this chapter, the test program, its framework, design process and implementation, is explained and evaluated.

5.1 Development tools and process

Coding for mobile environment has changed drastically in the last couple of years and as the mobile phones have evolved, there is no longer need for extra thought for memory and processing power issues. Mobile coding is now closer to normal desktop coding with only few special perks.

The initial thought in this research was to code everything for Nokia N9 camera phone which was available in the beginning. Coding for N9 was done with Qt C++ which is very close to standard C++ (Qt Project, 2014). However, as the work progressed, a Samsung S4 phone which has more processing power than the N9, became available. Most of the code had already been implemented and thus it was natural to continue work with the Android NDK (Native Development Toolkit). Only GUI was designed and implemented with Android Java.

The work requires also libraries for image processing, filtering, transformations, camera calibration and matrix calculations. OpenCV is an open source library for computer vision and machine learning and it contains over 2500 algorithms. It is widely used and supports multiple platforms, including Android. In here, C++ library was used. There is also a Java library for Android available but its use was not deemed necessary. (OpenCV, 2014a.) In addition, it is beneficial for the watermark system to be as portable as possible and C++ offers this more easily in the case of OpenCV.

5.2 Requirements

The requirements for the application were determined from the scenarios introduced in Chapter 2.4 as well as the fact that the intended users of the software were trained professionals and researchers. There are several requirements for the application:

1. The user should be able to see a viewfinder of the camera view.
2. The user should be able to take a picture according to the view.
3. Pictures should be saved for later.
4. The user should be able to change the watermarking method.
5. User should be able to see the watermark result
6. A progress of the watermark extraction process should be show to the user.
7. Most of the camera parameters should be automated.

8. Some camera parameters should be available for change if need arises while testing, for example resolution.

In addition, it was important to make such coding selections that addition of extra watermarking methods later is as easy as possible.

The requirements 1 and 2 are basic for any camera related application. The user should be able to see what is happening and direct the camera accordingly. The requirement 3 is for research purposes later on. Had this been a commercial application, this requirement is not essential and may even be a hindrance. The requirement 4 is needed if there are multiple watermarking methods available. One possibility is to use a separate program for each of the methods, but this introduces lot of redundancy of coding. The requirement 6 is of course needed but in commercial application this could vary according to the method. Some methods could show only the embedded text whereas another application would launch an Internet browser. In this research application, it is more practical to see the numeric or textual watermark message. The requirement number 7 is fairly ambiguous but means that the application need not to be a full camera application. It is not relevant for the user to be able to change for example exposure or white balance. In order to be able to accurately refine camera settings the user should have detailed knowledge of the watermarking method. This is not the aim of the print-cam robust watermarking methods where the methods are aimed to be as easy to use as possible.

5.3 Architecture and design of the application

Figure 12 shows an overview of the planned architecture. The graphical user interface is naturally set aside from other functionalities. Camera management system manages taking of the picture and saving it for the further use. It also sends all the requests for watermark extraction forward to the watermark system. The watermark system handles the extraction of the watermark according to the selected watermark method and connections to OpenCV library.

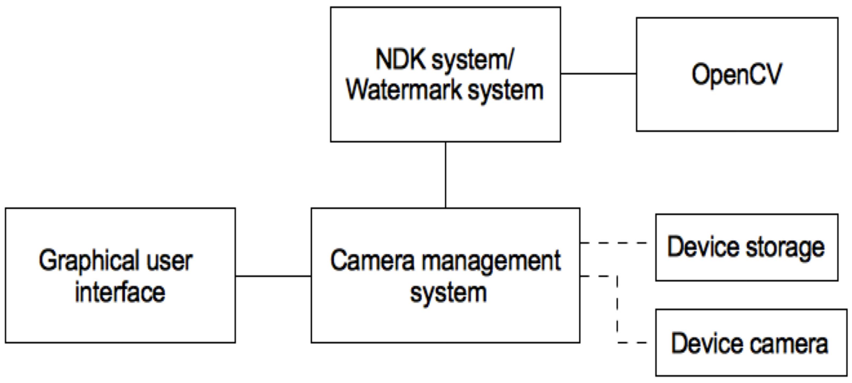


Figure 12: Overview of the system architecture

The requirements are fulfilled with the architecture. Camera viewfinder is show in the user interface as well as buttons for image capture and settings. Camera management system takes care that the images are taken with right settings, saved properly and requests are made for watermark processing and appropriate responses are show to the user. The watermark system takes care of processing the watermarks by using OpenCV libraries when needed.

A more detailed look is illustrated in Figure 13 with the class diagram of the application. The camera management is handled with Java and handles interfaces to device camera and storage. The watermark system is implemented with C++ and implements factory model in which different watermarking methods are products. This offers the possibility to add more watermarking methods later to the system.

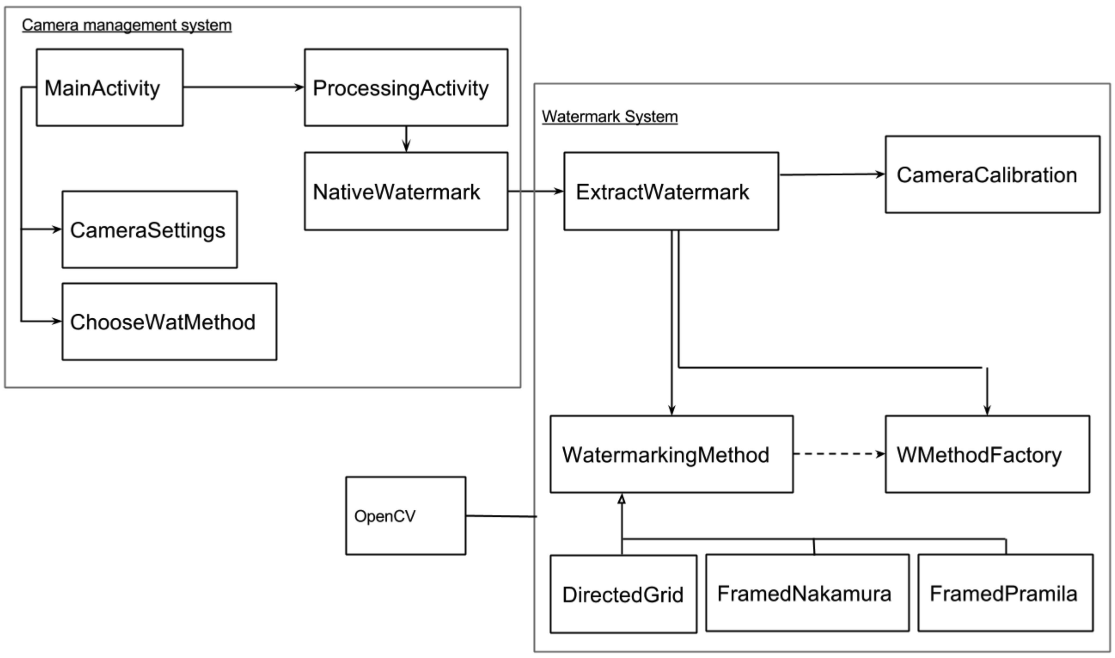


Figure 13: Class diagram of the application

5.4 Implementation of the applications

The implementation was realized on a Samsung Galaxy S4 GT-I9505 mobile phone and the code implemented by using Eclipse with Android ADT tools and Android SDK and NDK. Samsung S4 has a 1.9 GHz quad-core processor with 13 megapixel camera and 2 GB of RAM. The test phone had an Android 4.3 (Jelly Bean, API level 18) installed. Taking account the fact that even the most recent of the researched watermarking methods was proposed with 8 megapixel camera, these specifications should be enough.

The first prototype consisted only of simple camera image capture and one watermarking system. Later the rest of the watermarking systems were implemented along with more complete settings options. The use of factory model was clear from the beginning but the functioning of the watermarking methods were unknown. This was one of the research questions for study if they would work as such or if changes to reported methods were required. These changes have been reported on more detail on Chapter 6.

The GUI is shown in Figures 14, 15 and 16. Figure 14 shows the main GUI window which contains the viewfinder and camera button. Figure 15 shows the watermark processing -window, that has a timer for how long the watermark has been processed and an exit button for terminating calculations and returning to the main view. Figure 16 shows the list view of all the implemented watermarking methods that is shown when the user selects menu button in the main view.

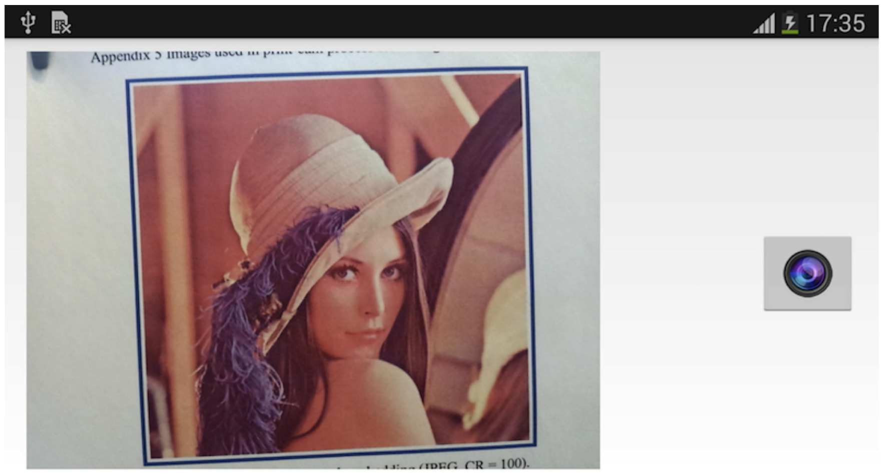
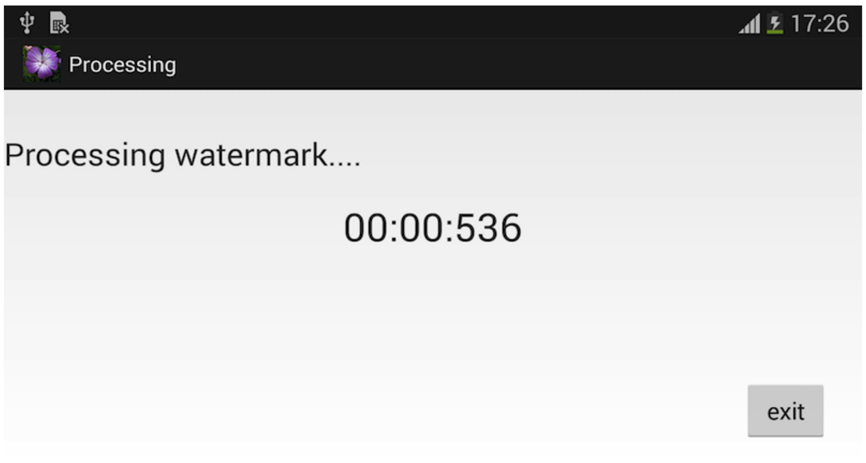


Figure 14: The Main GUI window showing viewfinder and camera button



17:26

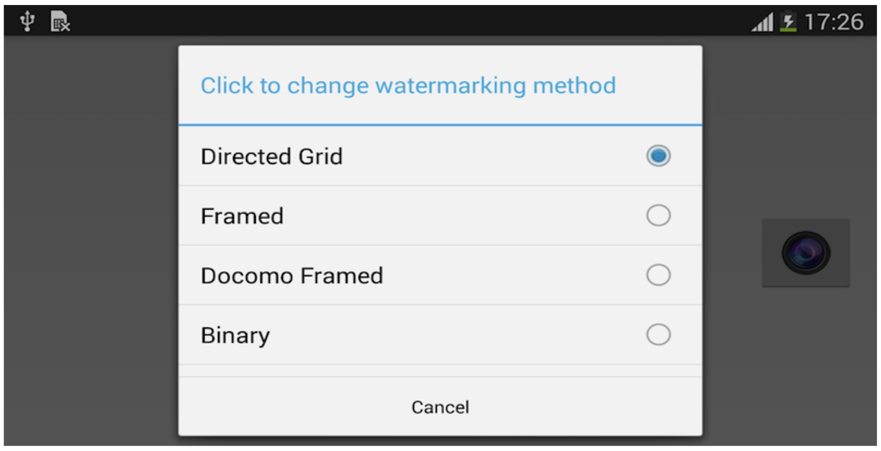
Processing

Processing watermark....

00:00:536

exit

Figure 15: After image is captured, watermark is read and the window with timer is shown. Watermark code is shown under the timer after the watermark is read.



17:26

Click to change watermarking method

- Directed Grid ●
- Framed ○
- Docomo Framed ○
- Binary ○

Cancel

Figure 16: The watermarking method is selected from a list that is shown when the user presses menu button.

5.5 Evaluation of the application and limitations

The system was tested against defiances and that each of the buttons and actions worked as designed. The application was also tested against memory leaks and all the requirements were fulfilled.

Some limitations must be noted about the system that were discovered while implementation and evaluation.

1. The software was locked on landscape mode to reduce ambiguity during watermark extraction and image capture. Most of the watermarking methods are not robust to over 90 degrees of rotations and using landscape mode reduces accidental errors that might occur when phone is rotated.

2. The lighting must be good because most of the watermarking methods require good lighting and do not work well with flash.

3. The camera detects automatically white balance and exposure from the image centre which may not be optimal for the watermarking methods. However, studying the effects of automatic white balance and exposure is beyond scope of this research.

5. The system cannot guarantee that watermarking extraction process is each and every time successful and does not recognize incorrect answers. This is due to the properties of watermarking methods.

6. Comparative analysis of the selected watermarking methods

In this chapter, implementation issues of the three watermarking methods are first explained. Then the robustness of the methods is tested and the obtained results analysed and discussed.

6.1 Implementation issues

The implementation of the method by Nakamura et al. (2004) and Katayama et al. (2004) was relatively straightforward. However, the fact that the method was originally created for a phone with a low resolution camera caused some issues. Foremost of the issues was that the frame detection method did not work as such. This is due to the resampling of the watermarked image during capture. In an image captured with small resolution, all the significant edges tend to be prominent and variation between pixels large. On the other hand, in high resolution images, the edges may be smoothed and may include artefacts as illustrated in Figure 17.

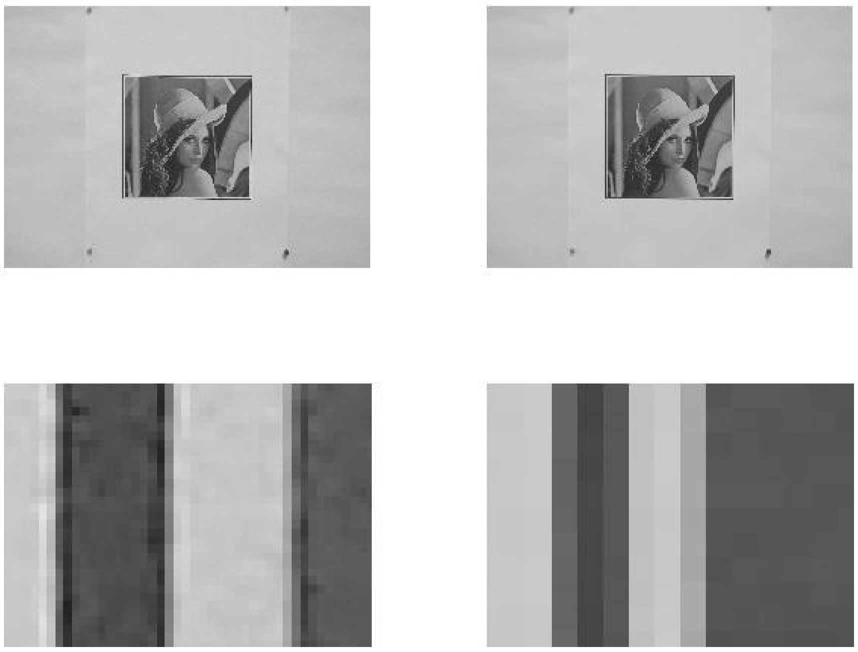


Figure 17: Frame in high resolution and low resolution images

In the frame detection method by Katayama et al. (2004) the frame was found by first searching the leftmost edge of the frame by starting at the midpoint of the left edge of the captured image and continuing towards right edge of the image. The edge was found by thresholding values and calculating the minimum and maximum values along the search line.

This works for low resolution images, but in high resolution images, there are a lot more variation that needs to be taken into account. In addition the edge may not be an exact point and is difficult to determine accurately. Here, more information means also more uncertainty and redundant data.

After the frame edge was found, a 3xn filter was created, in which the n is the width of the frame. This filter was then slid along the found frame edge until corners of the frame were found. This method is highly dependant on the accurate measurement of the frame width, which as mentioned, is difficult in high resolution images. In addition, the frame width changes in the captured image according to the distance and tilt of the camera. (Katayama et al. 2004.)

In order to achieve better robustness in frame detection a different method from Katayama et al. (2004) was developed and used. First, captured image is convolved with a small Gaussian kernel in order to smooth distortions occurring on the edged of the frame as can be seen from the Figure 18. Next the image is thresholded with an adaptive threshold and a black and white image is obtained.

$$dst(x,y)=\begin{cases} maxvalue, & if\ src(x,y)>T(x,y) \\ 0, & otherwise \end{cases} \tag{2}$$

where T(x,y) is a threshold calculated individually for each pixel. Finally, contours are found from the binary image with an algorithm by Suzuki and Abe (1985). These contours are then translated into polygons which are further simplified with Douglas-Peucker algorithm (Heckbert, P. & Garland, M., 1997) in order to obtain simple representations of the curves in the image. This is done so that the rectangle of the frame is identified by searching through the obtained polygons and finding the largest polygon with four corners.

After the four corners are found, the projective transformation can be formed with the following equations:

$$\begin{gathered} x'=\frac{a_1x+b_1y+c_1}{a_0x+b_0y+1} \\ y'=\frac{a_2x+b_2y+c_2}{a_0x+b_0y+1} \\ (x',y'):original\ picture\ position \\ (x,y):camera\ picture\ position \end{gathered} \tag{3}$$

,where the coefficients can be determined from the original and measured corner points.

The large resolution affected to the watermark extraction process also with another way. When the image was taken close enough but still well within the operation limits, distortions from printing process begin to affect the watermark. Figure 18 illustrates how the printing artefacts are seen in the printed and captured image. The artefacts are a result of half-toning process by the printer.

In order to remove these distortions in the image, all the captured images were filtered with a small Gaussian blurring filter. This 3x3 filter smooths the half-toning artefacts. It is unknown how big an effect this filtering had for the watermark, but on many cases the watermarking methods did not work at all without first removing the effects of half-toning.

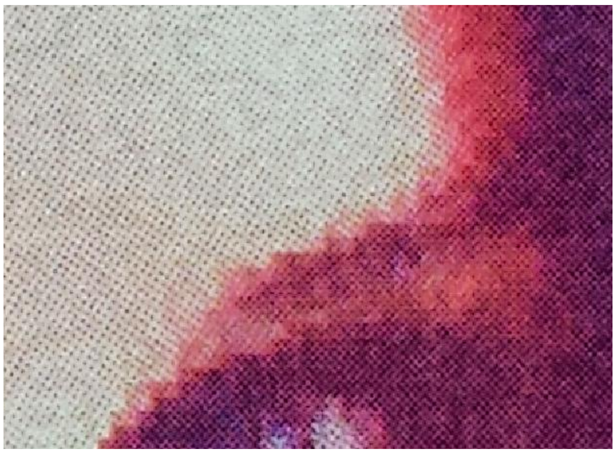


Figure 18: Zoomed detail of a watermarked, printed and captured image.

Similar problems were encountered with the other frame-based that was inspected in here, namely method by Pramila et al. (2008). The original method included a template for finding out the translation of the image (Pramila et al. 2008) in addition to the frame for synchronization. However, in this research, the template for correcting translations was deemed to be unnecessary and the new frame detection method accurate enough to correct all the perspective distortions, including rotation in 3D space and translation.

Other implementation issues include the effects of half-toning by the printer which were removed with the gaussian blurring filter as explained previously. In contrast to other watermarking methods discussed in here, the method by Pramila et al. (2008) requires correction of lens distortions, i.e., calibration of the camera. Fortunately, OpenCV library provides readily available functions for camera calibration (OpenCV, 2014b) that was used in this research.

The third method to be implemented was the autocorrelation based method by Pramila et al. (2012).The method was originally implemented for a Symbian S60 mobile phone with Symbian C++ programming language. Therefore, implementation with standard C++ was straightforward. Because the phone had 5 megapixel camera, implementation and testing with a 13 megapixel camera offered no surprises. Nevertheless, it was necessary to remove the effects of half-toning with the gaussian blurring filter. The half-toning effects were especially noticeable when the images were taken at close range.

6.2 Robustness experiments and results

In order to compare robustness of the three methods, some experiments are needed. Here we are only interested in unintentional attacks and especially robustness against 3D rotations, that is, rotations around optical axis of the camera. These rotations occur naturally as it is difficult for a human to take picture perfectly perpendicularly to the watermarked image. Another factor that may affect watermark robustness if the distance of the camera from the watermarked image. The relative size of the printed pixels in the captured image will get smaller as the distance is increased, eventually destroying the image details and the watermark.

The embedding of the watermarks was done on a computer with Matlab and the uncompressed images were printed with an office printer. Six different images were used to better evaluate visibility of the watermark and show that the methods work on various images, instead of only one. Six images were chosen for testing and these images are shown in Figure 19. The images were each watermarked with the methods explained in Chapter 4. Capacity was selected and fixed to 32 bits for each of the methods and included an error correction coding reducing the payload to 26 bits. The error correction coding used was Hamming (32, 6) error correction coding that is capable of correcting one bit or detecting two erroneous bits. Each of the images were embedded with the same watermark message so that the method itself do not disturb experiments.

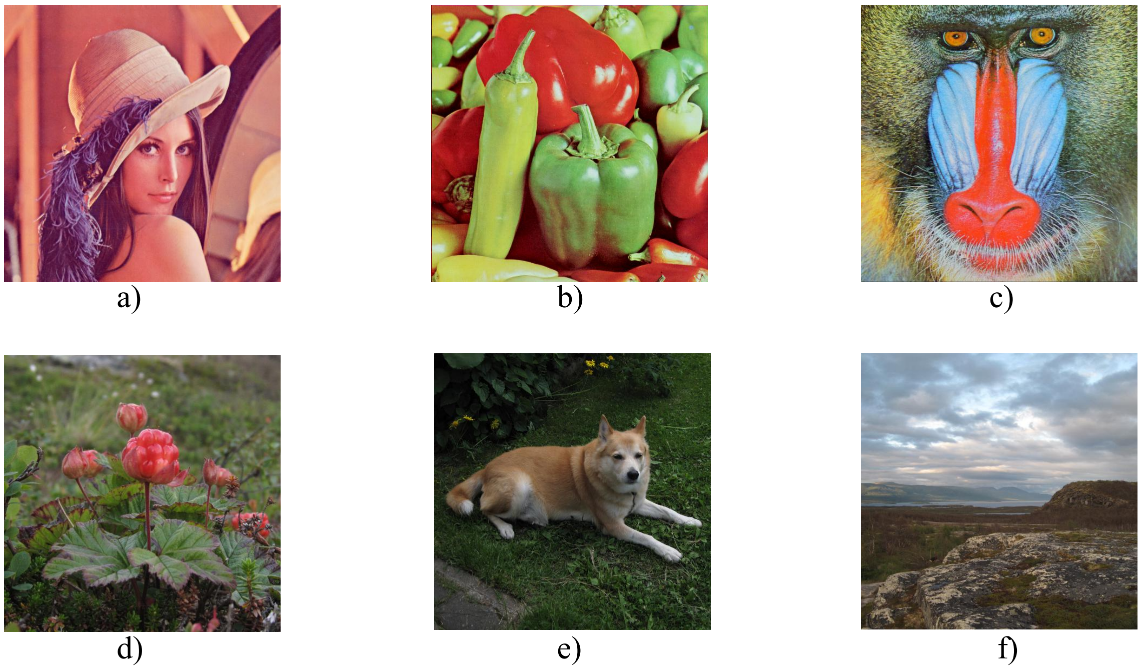


Figure 19: Original unwatermarked images a) Lena, b) Peppers, c) Baboon, d) Cloudberry, e) Dog and f) Abisko

The strength of the watermark and thus the imperceptibility factor of the watermark was selected as explained in each of the research papers respectively. The watermarked images used in the tests are collected and show in Appendix A. The printed size of a framed image was approximately 13.5cm.

The test setting is depicted in Figure 20. The watermarked image was fastened to a wall and the camera was placed in from of it on tripod. The tripod was used in order to accurately test effect of distance and angle of the camera to the watermark.

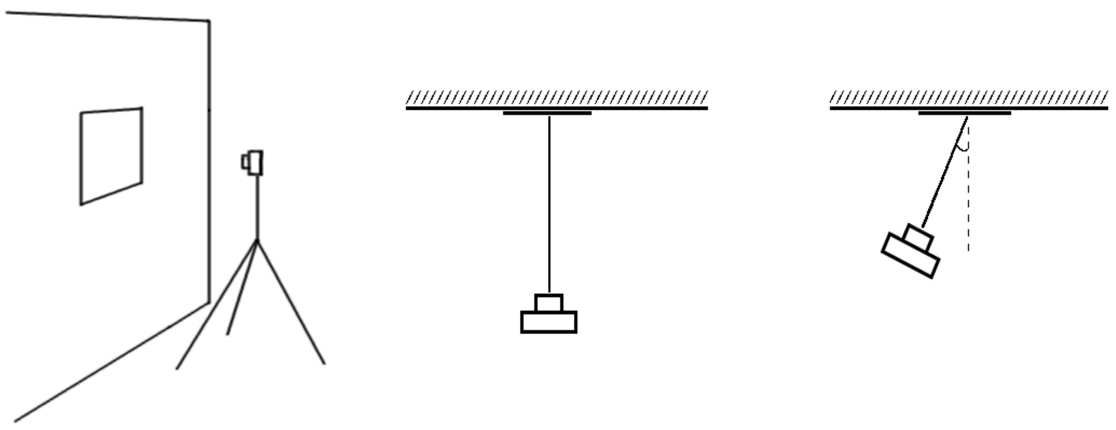


Figure 20: a) Test setting with camera on tripod and watermarked image on the wall b) Camera distance from the wall c) Camera angle to watermarked image. (Old styled camera used in this image for illustration purposes)

The tests were conducted by first gradually changing the distance of the camera to the wall with a small step size of 2cm from 10cm to around 40cm. After 40cm, the step size was increased in order to found the braking point of each of the methods. Each watermarked image was captured two times with full camera resolution of 4128x3096 (~13MPx) from each distance. This was done, so that unfocused images could be eliminated and more data collected. The obtained images were saved for future use and the watermark extracted. A BER (Bit Error Rate) value was calculated for each of the extracted watermarks and these obtained measures were collected to a diagram shown in Figure 21.

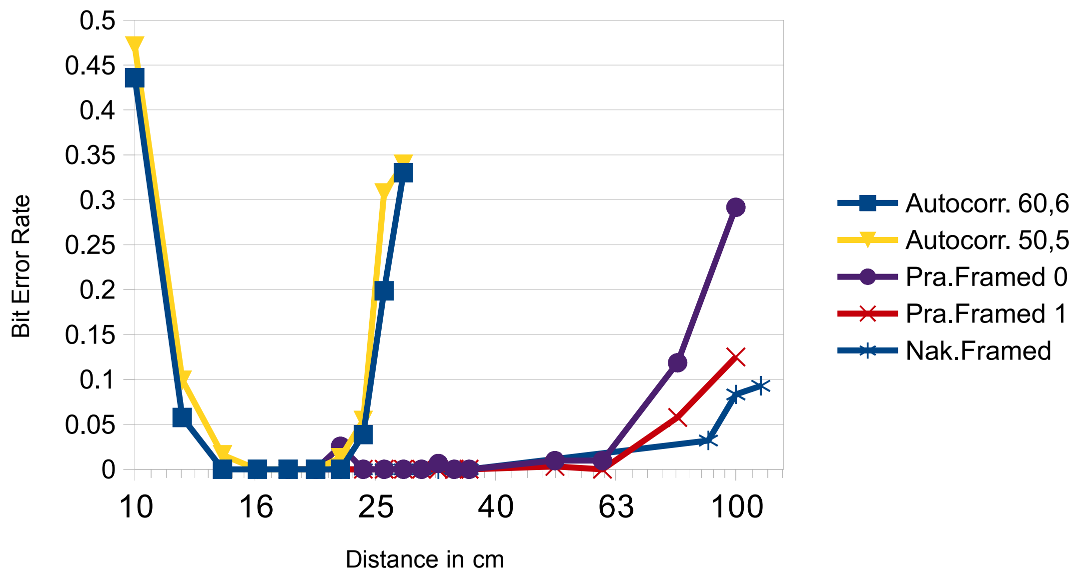


Figure 21: BER with resolution of 4128x3096 on y axis and distance of the camera to the watermarked image on x axis.

In Figure 21, Autocorr. means the autocorrelation based method by Pramila et al.(2012) and the messages were embedded with the same two strengths as in the publication. Pra.Framed 0 means the frame based by Pramila et al. (2008) in which the message was repeated 3 times and mean of each bit was selected in the extraction process. In this way the message was embedded only in horizontal wavelet details of the image. Pra.Framed 1 is a method in which the message was repeated 6 times and the message was thus spread among both horizontal and vertical wavelet details. This was done, because the method by Pramila et al. (2008) has much larger capacity than the other methods. Nak.Framed is the frame based method by Nakamura et al. (2005).

Because the watermark algorithms are realized with C++ in a separate module from GUI on the phone, it was possible to later rerun the tests on computer with different settings. The images were programmatically resampled to size of 2064x1548 (~3MPx) and watermarks extracted. The obtained results are shown in Figure 22.

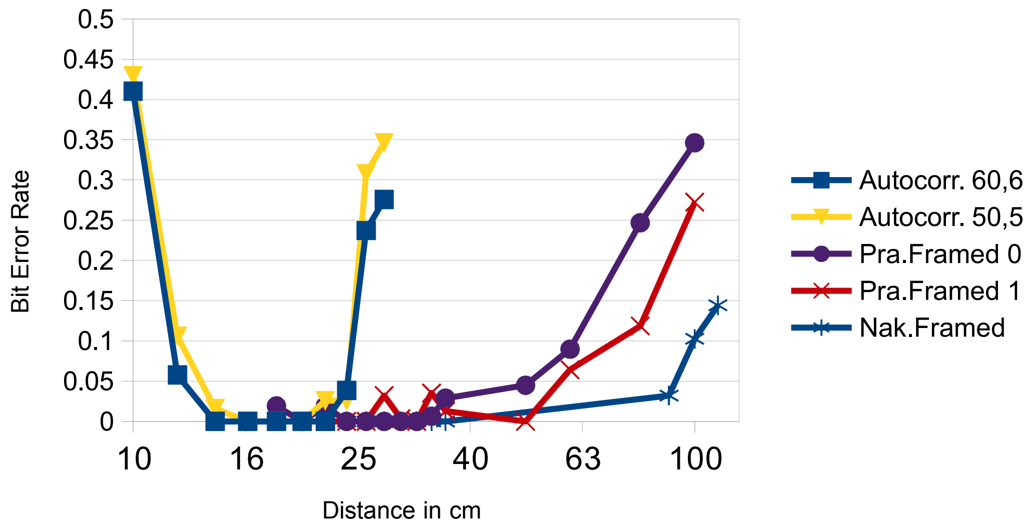


Figure 22: BER with resolution of 2064x1548 on y axis and distance of the camera to the watermarked image on x axis.

Figures 21 and 22 show that the autocorrelation based method by Pramila et al. (2012) works well on close distances but breaks fairly quickly when the distance is increased. The frame based methods do not work at all on close distances but work well until 50cm.

Some example images are included in Figure 23 from various distances during testing. These images show how the distance affects the watermarked image size in the captured image.

The effect of the angle of the camera to watermarked image was calculated by selecting a fixed distance to the camera and gradually changing the angle. Unfortunately it was impossible to choose one distance, because from the distance measures it was clear the autocorrelation based method worked best on distances from 12-22cm whereas other methods did not work at all before 18cm because the frame was otherwise cut of. For the framed methods, it was necessary to select such a distance that the whole watermarked image with frame fit into captured image even with large angles. The autocorrelation based method was robust to some cutting of the image and therefore the image could be captured closer to the wall. Therefore, distance of 18cm for the autocorrelation based method and distance of 24cm was selected for the other methods. Obtained results are shown in Figure 24 and Figure 25.

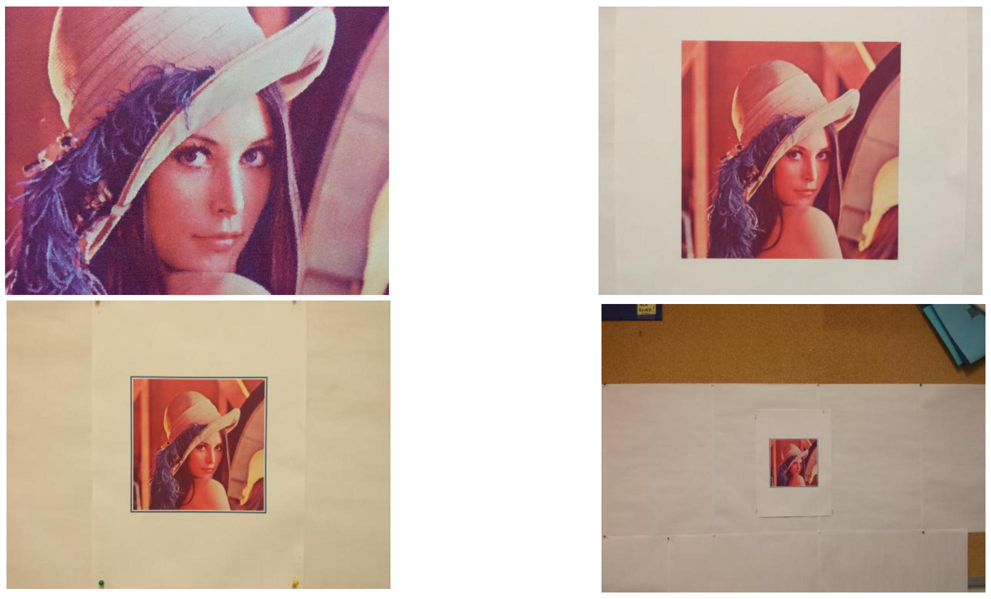


Figure 23: Example images during distance testing a) 12cm b) 22cm c) 36cm d) 100cm

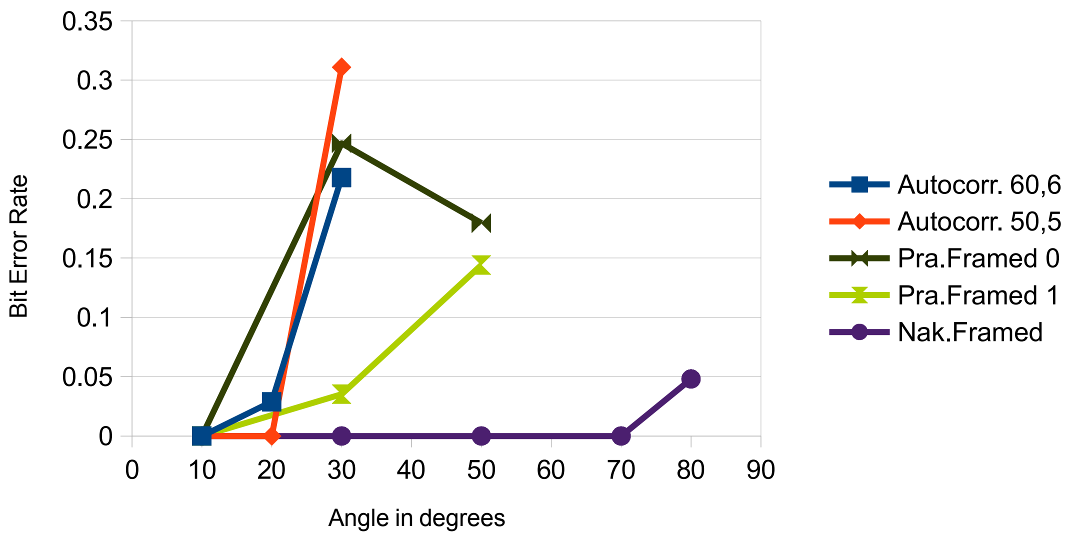


Figure 24: BER with resolution of 4128x3096 on y axis and angle of the camera to the watermarked image on x axis.

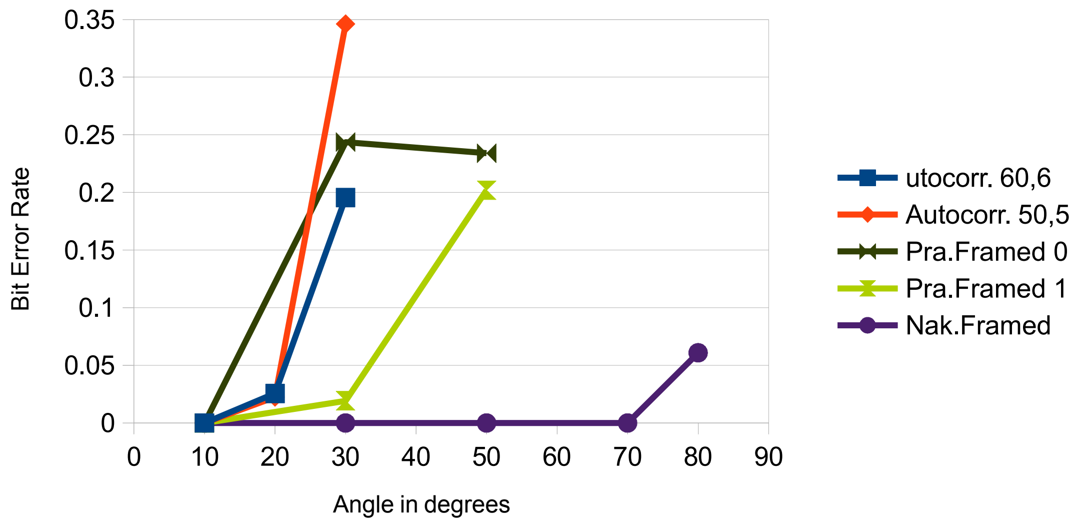


Figure 25: BER with resolution of 2064x1548 on y axis and angle of the camera to the watermarked image on x axis.

Figures 24 and 25 show that the autocorrelation based method by Pramila et al. (2012) which does not correct any distortions before extracting the watermark, is only robust to around 20 degrees of rotation. This is well in line with the reported measurements in the publication. The framed method by Pramila et al. (2008) also breaks around 20 degrees but it seems that using two different detail coefficients that are orthogonal to each other helps to preserve the message somewhat. The method by Nakamura et al. (2004) is the most robust of the methods against rotations.

Table 1 contains measurements for processing speeds of each of the methods with specified resolutions. The measurements were made by taking five images with each of the watermarking methods and calculating the mean time of those runs. The speeds are only approximate because the code is not optimized and the code was build with Android NDK with debug flag on.

Table 1: Processing speeds on Samsung Galaxy S4 for each of the methods with specified resolutions.

Method	**Resolution 4128×3096**	**Resolution 2064×1548**
Autocorrelation	3.02s	3.02s
Pra.Framed	8.09s	6.04s
Nak.Framed	4.03s	2.02s

These values are not exactly the same as on respective publications due to the changes made to the methods and different mobile platform as well as different image sizes.

6.3 Analysis and discussion of the results

The obtained results that were reported in the previous subchapter show significant performance differences between methods. First, when we inspect the distance measures, it is clear that the autocorrelation based method by Pramila et al. (2012) works well from distances 14cm to 24cm which is somewhat closer than the other two methods. The frame-based method by Pramila et al. (2008) shows robustness from 18cm to ~50cm and the other frame-based method, by Nakamura et al., show robustness at distances 18cm to ~80cm. The frame-based methods are not robust to cropping as the frame is needed for synchronization. The autocorrelation based method by Pramila et al. (2012) is robust to some cropping and thus the images can be captured a lot closer. However, the autocorrelation based method has no outside measure of synchronization or watermark location and thus it is not as such robust to large distance variations.

The method by Nakamura et al. (2005) seems to work very well when the angle of the camera to the watermarked image is increased. The other two methods do not work so well. The autocorrelation based method by Pramila et al. (2012) suffers from the same shortcoming as in distance testing, that is, there is no outside measure to help with synchronization. The frame-based method by Pramila et al. (2008) is surprisingly sensitive to rotation as well. However, this time the reason is probably the fact that the message is embedded in line by line basis instead of in blocks. As the image is rotated, the line get closer and closer to each other on one side of the image and thus their separation becomes difficult.

The method by Nakamura et al. (2005) is not, however, perfect either. When we look at the embedded images as show in Appendix A, it is clear that the method is visible to the eye. Other two methods are almost imperceptible but in the images embedded with the method by Nakamura et al. there is visible distortion due to the periodic sine curves that were used while embedding.

The PSNR values are not calculated here, because they would not show well the discrepancies in visibility of the methods. PSNR values rely on calculating differences between each pixel over whole image. However, the JND values or other that spatial embedding domains are used, the embedding strength varies between each pixel and the total about of distortions is larger even though visibility has not been affected. The payload is spread unevenly across the image and PSNR cannot show this.

The embedding strengths were chosen by reported embedding strength in each of the methods separately. It would have been beneficial to conduct user tests and embed each of the methods with similar imperceptibility value. This, however, would also have needed a similar JND method for calculating the strength for each of the pixels and it would have been difficult for wavelet domain.

The method by Nakamura et al. (2005) seems to be superior in robustness to other methods. However, when other views have been taken into account the selection is no longer clear. Autocorrelation based method by Pramila et al. (2012) is robust to cropping and works on multiple image shapes, whereas frame based method by Pramila et al. (2008) has superior capacity compared to other methods. The publication of Pramila et al in 2008 showed that original digital watermarking methods may work very well on print-cam surroundings if synchronization is properly addressed. This increases the amount of possible print-cam robust watermarking methods and especially to security applications, in which, e.g., the border of an identity card works as a synchronization frame.

The limits of the updated frame detection method did not become evident until rotation of more that 60 degrees and distance of more than 100cm. After these, separately, the frame detection method begun to fail and extraction of the watermark became more difficult. However, also the watermarking methods themselves begun to fail. While testing, situations when the frame detection method failed, were ignored as it was not considered to be part of the watermark reading process itself. These failures did not occur, however, until very late during the testing with high amounts of distortion, when it had already become evident that the watermarking methods had also failed.

The processing speeds were reported for the two different resolutions and the times varied approximately form two seconds to eight. The change of resolution had an effect to processing speeds of the two frame-based methods. This is due to the fact that searching the frame and normalizing the size of the watermarked image from the captured image is time consuming. After the frame has been found, the processing speed of the watermark extraction process if constant as in the autocorrelation based method. The method by Pramila et al. (2008) is the slowest because the watermark is processed in as big image as possible. However, the other two methods are scaled down by design in order to speed up the processing.

The images in Figure 23 show that exposure varies between captured images. This is due to the fact that the exposure and white balance are calculated from the centre of the viewfinder and thus may not offer accurate representation of the whole situation. However, it is difficult to determine which settings for exposure and white balance would be better that others. These values depend not only surroundings of the test setting but also the image properties. For a dark image, the values would be different than for light image. The automatic approximation algorithms also struggle if a dark object is placed on light background. This, however, is well known problem in all imaging and it is not exclusive to print-cam robust watermarking. it would, nevertheless, interesting to do some research on the subject and how different values affect watermark robustness.

As a conclusion, it can be said that the autocorrelation based method by Pramila et al. (2012) has mediocre imperceptibility, low capacity and mediocre robustness, but it is not restricted by image shape. The frame based method by Pramila et al. (2008) has high imperceptibility, high capacity but low to mediocre robustness to 3D rotations and requires calibration of the camera. Although it is highly robust to distance variations. The frame based method by Nakamura et al. (2004) has low imperceptibility, low capacity but high robustness. These results are well in line to the reported robustness experiments of each of the publication. It is up to the application which method works the best.

7. Summary and conclusion

Digital image watermarking has gained popularity constantly from mid 90's. Application areas have evolved from broadcast monitoring and DRM based copy and copyright applications to more sophisticated authentication and value-added systems that aim at being beneficial to the user.

Print-cam robustness of a watermark would bring the watermark extraction closer to the user and make the extraction process independent of time and place. Most of the watermarking systems work only on digital work but a lot of images are often printed for various applications on paper. For example, posters advertising upcoming events, periodicals showing up to date fashion and identity cards authenticating people with access to different locations. Watermarking is not visible and thus disruptive of aesthetics unlike barcodes and thus have a different application space.

In this work, few existing print-cam robust watermarking methods are implemented on mobile phone and tested. Previously, it has been difficult to compare the methods, because all of them have been developed for slightly different applications and feature sets. However, one of the aims of this research is to bring the old print-cam robust watermarking systems up to date through implementation and to report what changes needed to be done when methods were updated to a modern mobile phone and if the performance of the methods has changed.

The first research question was *how the watermarking methods are affected when implemented with a modern mobile phone?* And the related second research question *are the first watermarking methods proposed still viable?* It was possible to implement all of the three selected watermarking methods for a modern mobile phone. However, some changes were required in order to make the methods work properly.

The largest factor while implementation was the resolution of the camera. It would seem that larger resolution is only beneficial but it also means longer processing times when the software has more pixels to process and more room for noise in the images. The first watermarking methods were developed for less than 0.3 megapixel cameras and the modern mobile phone used here had a resolution of 13 megapixels. This disparity on image sizes was noted to be the main factor that affected to the watermarking extraction processes.

The notable changes to the watermarking methods involved a new frame detection method and gaussian filtering to blur half-toning effects of the printing process. The watermark embedding and extracting methods themselves were not affected.

The third research question asked *what are performance differences between the selected three different watermarking methods on same hardware?* The obtained results showed that all the three methods were designed for different applications and there were quite large performance differences. Robustness to distance variations and rotations were tested and the results were reported. In addition, processing speeds were measured.

The fact that tests were conducted with only one camera phone is clearly a limitation, but as the phone was selected as one of the top mobile phones on the market, the test results should be valid some time to the future. The test program that was developed during the research worked well and within requirements set for it, and the results were in line with the respective watermarking methods. The deviations in processing speeds can be explained with differences in image sizes.

Another limitation of the work and possible cause of future work is the fact that the watermarking methods were all embedded with embedding strengths according to each of the publications separately instead of selecting a fixed watermarking strength. This was done in part for simplicity as it is difficult to determine a variable JND observing watermarking strength that would work with each of the watermarking methods. Another reason was that watermarking methods generally are designed as a packet or a black box, with everything from imperceptibility, capacity and robustness being part of the packet. Changing large parts of the method would require more extensive testing that this work had resources and time for.

However, it is clear from the implementation and obtained results that few changes were required to update the methods. One of the most important changes was the frame detection method that can be recognized as being non-optimal, but robust for even high amounts of rotation and distance variations of the captured images. It was not tested of how much distortion around the images the watermarking methods can handle including the frame detection method.

In addition, only one physical size of the images was tested. It would have been enlightening to use several different sizes of printed images. However, this rouses a question of watermark embedding. Should the watermark embedding process change when image size is changed? For example, the autocorrelation based method by Pramila et al. (2012) embeds the watermark in nine blocks but if the physical image size is increased more blocks could be used which may or may not affect the watermark robustness. Therefore amount of testing not only doubles but increases even more as different ways of embedding should be considered. This was deemed to be out of the scope of this research.

As a future work, the methods should be studied more carefully and best parts of each method discussed. Different methods will be invented that will rival these existing methods. However, these existing methods have already taught us that print-cam robust watermarking is possible and robust, however, application dependent.

References

Android NDK (2014) Retrieved from URL: https://developer.android.com/tools/sdk/ndk/index.html

Chou, C.-H. & Li, Y.-C. (1995) A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile. *IEEE Transactions on circuits and systems for video technology.* Vol.5, Issue 6, pp. 467-476

Cox, I.J., Miller, M.L. & Bloom J.A. (2002) Digital watermarking. Morgan Kaufman publishers, Academic Press, USA, 542 p. ISBN 1-55860-714-5.

Cox, I.J. & Miller, M.L (2002) The first 50 years of electronic watermarking. Journal of Applied Signal Processing, 2, 126-132.

Cox, I.J., Miller, M.L. & Bloom J.A. (2000) Watermarking applications and their properties. Proc. of the International Conference on Information Technology: Coding and Computing.

Digimarc Mobile E-Commerce Pilot (2014) Digimarc Mobile E-Commerce Pilot Debuts at Popular Tokyo-based “Maid in Japan” Café. Press Release on 25.7.2006 Retrieved from: http://www.thefreelibrary.com/Digimarc+Mobile+E-Commerce+Pilot+Debuts+at+Popular+Tokyo-Based+%27%27Maid...-a0148613093

Digimarc Discover (2014) Digimarc Discover , smartphones can instantly *see, hear* and *engage* with all forms of media and connect users to exciting interactive experiences. Retrieved from: http://www.digimarc.com/discover

Google. (2014). Core App Quality Guidelines. Retrieved from http://developer.android.com/distribute/googleplay/quality/core.html

Hanjalic, A., Langelaar, G.G., van Roosmalen, P.M.B., Biemond, J. & Langendijk, R.L. (2000) Image and Video Databases: Restoration, Watermarking and Retrieval. Elsevier Science B.V., Amsterdam, Netherlands, 445 p. ISBN 0-444-50502-4

Hartung, F. & Kutter, M. (1999) Multimedia Watermarking Techniques. *Proceedings of the IEEE*, vol.87, no.7, pp.1079-1107

Hartung, F. & Ramme, F. (2000) Digital rights management and watermarking of multimedia content for m-commerce applications, *Communications Magazine, IEEE*, vol.38, no.11, pp.78-84

He, D. & Sun, Q. (2005) A Practical Print-scan Resilient Watermarking Scheme. *IEEE International Conference on Image Processing (ICIP),* Sept. 11-14, 2005, Vol. 1, pp. 11-14

Heckbert, P. & Garland, M. (1997) Survey of Polygonal Surface Simplification Algorithms, *Multiresolution Surface Modeling Course, ACM Siggraph Course notes.*

Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004) Design Science in Information Systems Research, *MIS Quarterly*, vol.28, no. 1, pp.75-105

Hill, S. (2011) From J-phone to Lumia 1020: A complete history of the camera phone. Retrieved November 17, 2013, from http://www.digitaltrends.com/mobile/ camera-phone-history/

Katayama, A., Nakamura, T., Yamamuro, M. & Sonehara, N. (2004) New High-speed Frame Detection Method: Side Trace Algorithm (STA) for i-appli on Cellular Phones to Detect Watermarks. *Proc. of the 3rd International Conference on Mobile and Ubiquitous Multimedia, ACM International Conference Proceeding Series*, Maryland, USA, pp.109-116

Kim, W.-G., Lee, S.-H., & Seo, Y.-S. (2006) Image Fingerprinting Scheme for Print-and-Capture Model. *Proc. of Advances in Multimedia Information Processing, 7th Pacific Rim Conference on Multimedia*, Hangzhou, China. Vol. 4261, pp.106-113

March, A.T. & Smith, G.F. (1995) Design and natural science research on information technology. *Decision Support Systems*, 15(4):251-266.

Marek, S. (2014) Camera Phones Click With Marketers. Retrieved from: http://www.forbes.com/feeds/infoimaging/2004/10/04/infoimagingcahners_2004_10_04_eng-cahners_eng-cahners_092746_4862914162732535113.html Wireless Week, Oct. 4, 2004

Nakamura, T., Katayama, A., Yamamuro, M. & Sonehara, N. (2004) Fast Watermark Detection Scheme for Camera-equipped Cellular Phone. *Proc. of the 3rd international conference on Mobile and ubiquitous multimedia, ACM International Conference Proceeding Series,* Maryland, USA, pp.101-108

Nakamura, T., Ogawa, H., Tomioka, A., & Takashima, Y. (2000) Improved Digital Watermark Robustness against Translation and/or Cropping of an Image Area, *IEICE Trans. Fundamentals*, Vol. E83-A, No. 1

NTT Corporation (2014) NTT Develops "CyberSquash" Internet Access Platform using Electronic Watermarks. URL: http://www.ntt.co.jp/news/news03e/0307/030707.html News release on 7.7.2003

OpenCV (2014a) Main page. Retrieved from http://opencv.org/

OpenCV (2014b) Camera calibration with OpenCV. Retrieved from http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

O'Ruanaidh, J.J.K. & Pun T. (1997) Rotation, scale and translation invariant digital image watermarking. *IEEE Proceedings of International Conference on Image Processing*, Santa Barbara, California, USA, Vol. 1, pp. 536-539.

Pereira, S. & Pun, T. (2000) Robust Template Matching for Affine Resistant Image Watermarks. *IEEE Transactions in Image processing*, Vol. 9, Issue 6, pp. 1123-1129.

Perry, B., MacIntosh, B. & Cushman, D. (2002) Digimarc MediaBridge – The birth of a consumer product, from concept to commercial application. *Proceedings of SPIE Security and Watermarking of Multimedia Contents IV*, Jan 21-24, San Jose, California, USA, Vol. 4675, pp. 118-123.

Pramila, A., Keskinarkaus, A. & Seppänen, T. (2008) Watermark robustness in the Print-Cam process. *IASTED international conference on Signal Processing, Pattern Recognition and Applications*, Insbruck, Austria, 60-65.

Pramila, A., Keskinarkaus, A. & Seppänen, T. (2007) Camera based watermark extraction – Problems and examples. *Finnish Signal Processing Symposium (FinSig2007)*, Oulu, Finland, On CD.

Pramila, A., Keskinarkaus, A. & Seppänen, T. (2012) Toward an interactive poster using digital watermarking and mobile phone camera. *Springer journal of Signal, Image and Video Processing*, 6(2), 211-222.

Qt project (2014) Main page. Retrieved from URL: http://qt-project.org/

Reed, A. & Hannigan, B. (2002) Adaptive color watermarking. Proc. SPIE 4675, Security and Watermarking of Multimedia Contents IV. Doi:10.1117/12.465279.

Solanki, K., Madhow, U., Manjunath, B.S. & Chandrasekaran, S. (2004) Estimating and Undoing Rotation for Print-scan Resilient Data Hiding. *IEEE International Conference on Image Processing (ICIP),* Oct. 24-26, Vol. 1, pp. 39-42.

Suehle, R. (2011) The DRM graveyard: A brief history of digital rights management in music. Retrieved from http://opensource.com/life/11/11/drm-graveyard-brief-history-digital-rights-management-music

Suzuki, S. & Abe, K. (1985) Topological Structural Analysis of Digitized Binary Images by Border Following. *CVGIP* 30 1, pp 32-46

Takeuchi, S., Kunisa, A., Tsujita, K. & Inoue, Y. (2005) Geometric Distortion Compensation of Printed Images Containing Imperceptible Watermarks. *International conference on Consumer Electronics*, *Digest of Technical Paper.* 411 – 412.

Thongkor, K. & Amornraksa, K. (2013) Image Watermark Extraction for Capture Image With Partially Glass Reflection. *10Th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. 1-6.

Appendix A. Watermarked images

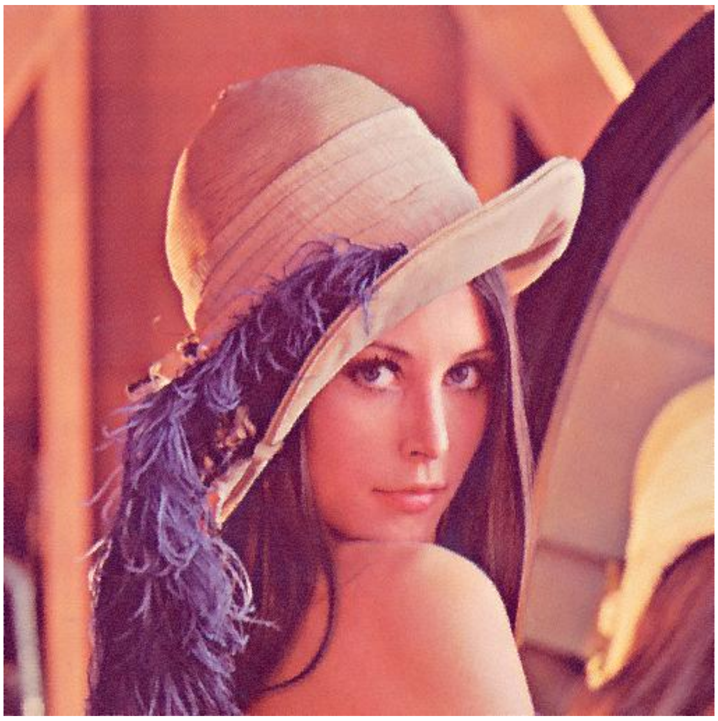


Figure A.1: Watermarked image (method Autocorrelation 50,5)

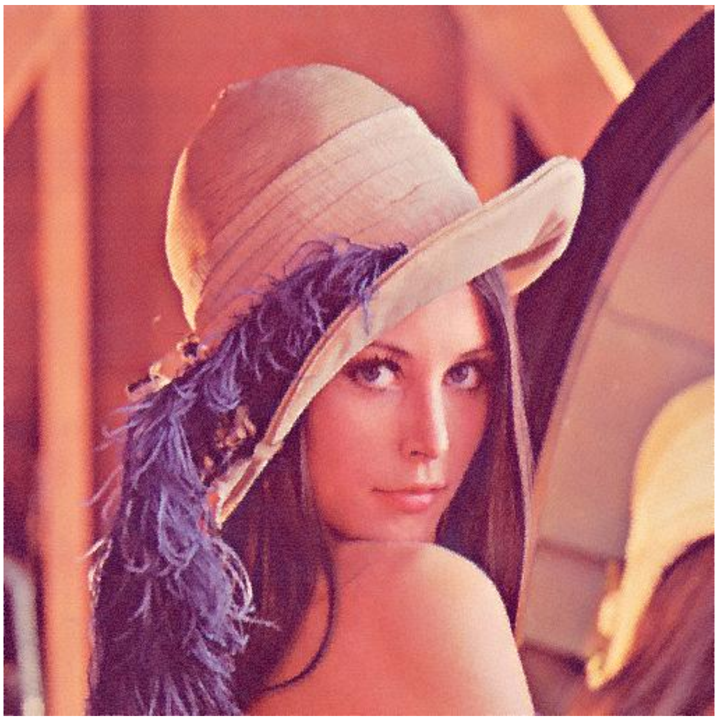


Figure A.2: Watermarked image (method Autocorrelation 60,6)

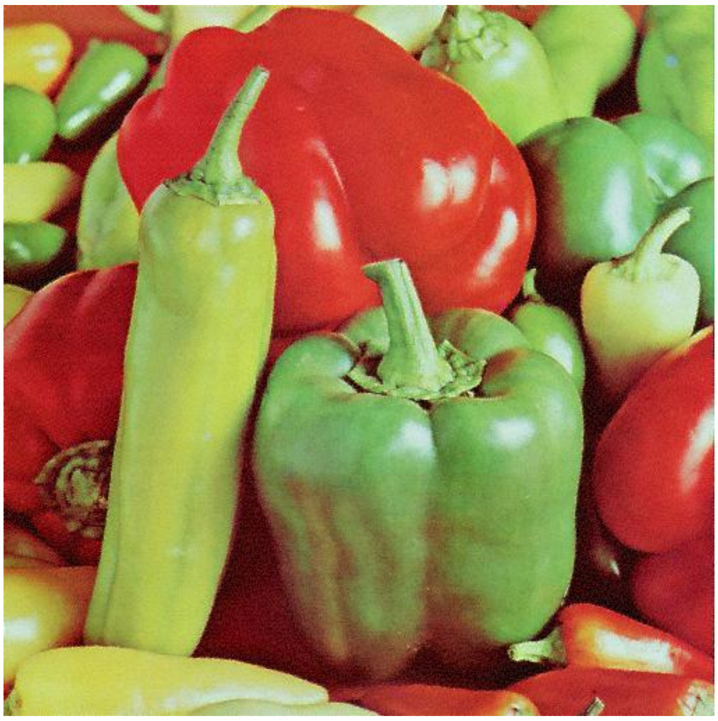


Figure A.3: Watermarked image (method Autocorrelation 50,5)

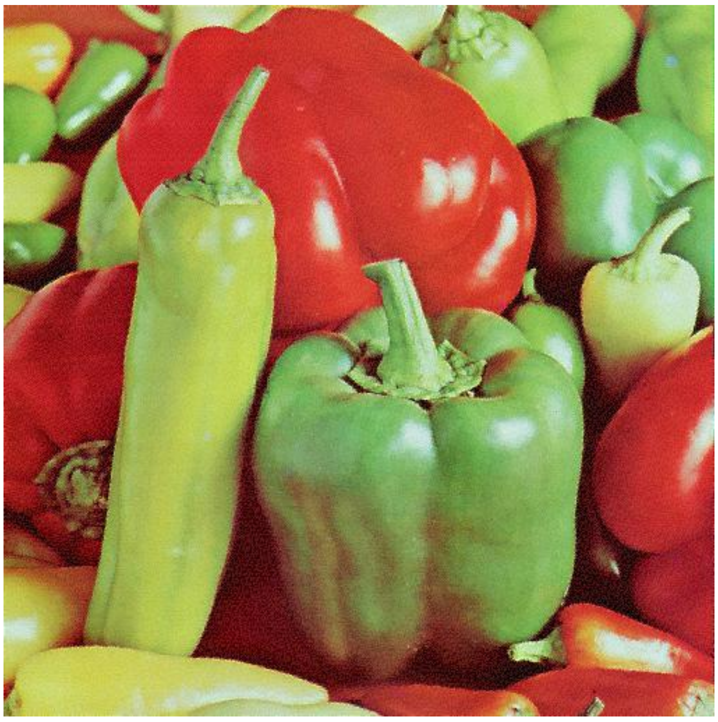


Figure A.4: Watermarked image (method Autocorrelation 60,6)

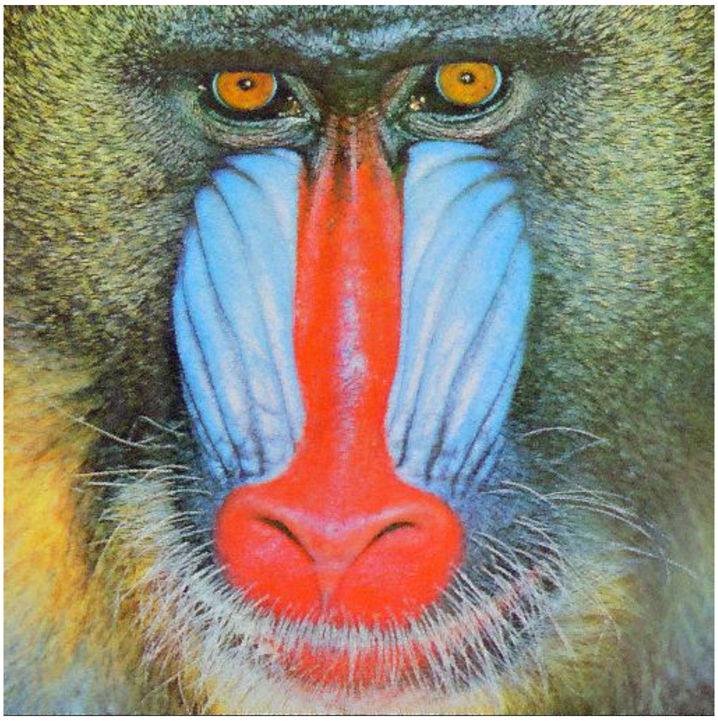


Figure A.5: Watermarked image (method Autocorrelation 50,5)

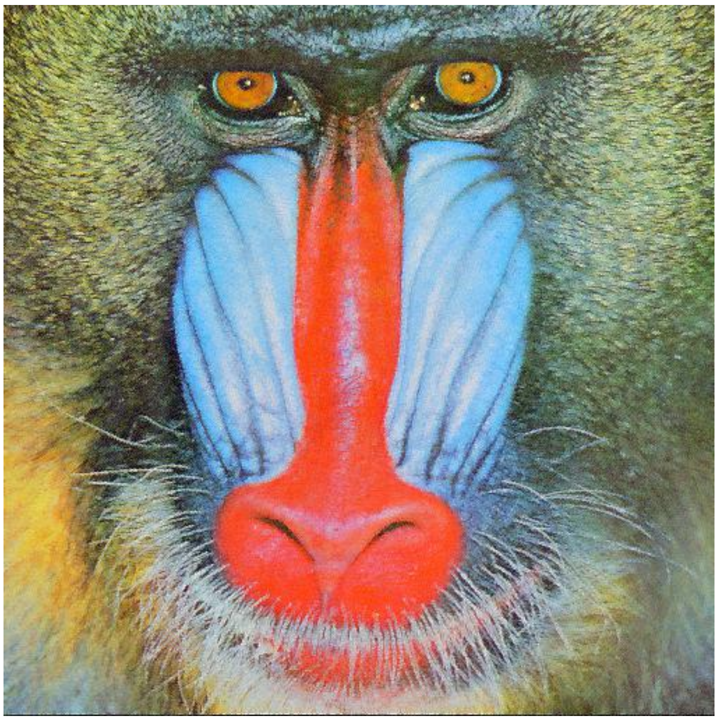


Figure A.6: Watermarked image (method Autocorrelation 60,6)

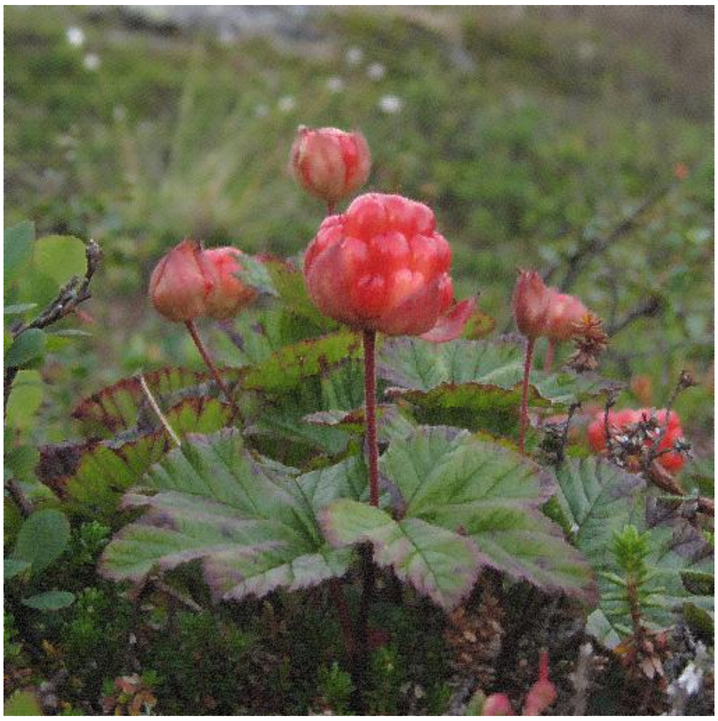


Figure A.7: Watermarked image (method Autocorerlation 50,5)

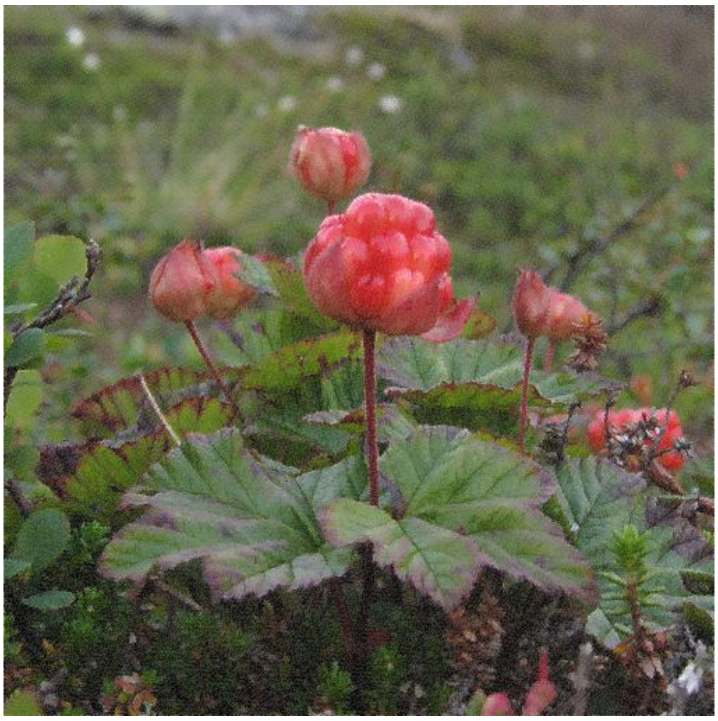


Figure A.8: Watermarked image (method Autocorrelation 60,6)

Figure A.9: Watermarked image (method Autocorrelation 50,5)

Figure A.10: Watermarked image (method Autocorrelation 60,6)

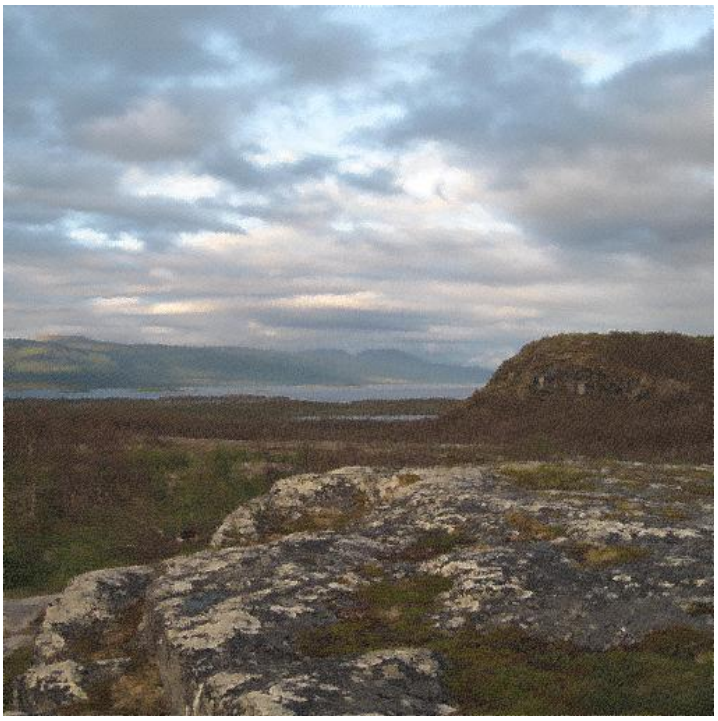


Figure A.11: Watermarked image (method Autocorrelation 50,5)

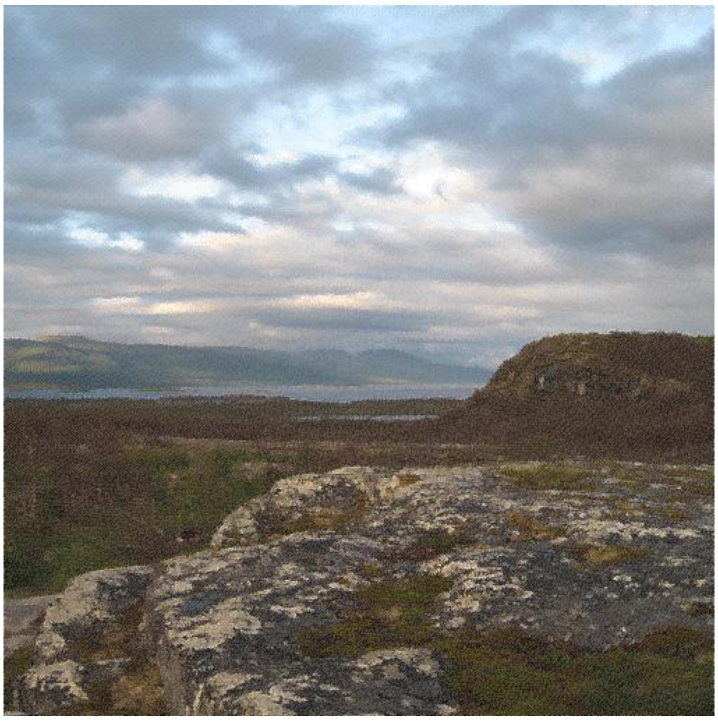


Figure A.12: Watermarked image (method Autocorrelation 60,6)

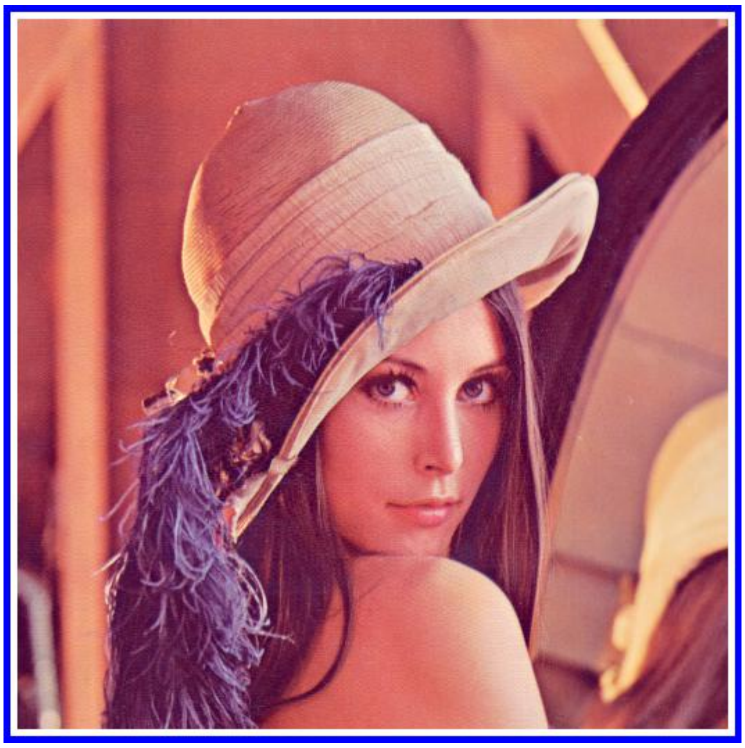


Figure A.13: Watermarked image (method Pra.Framed 0)

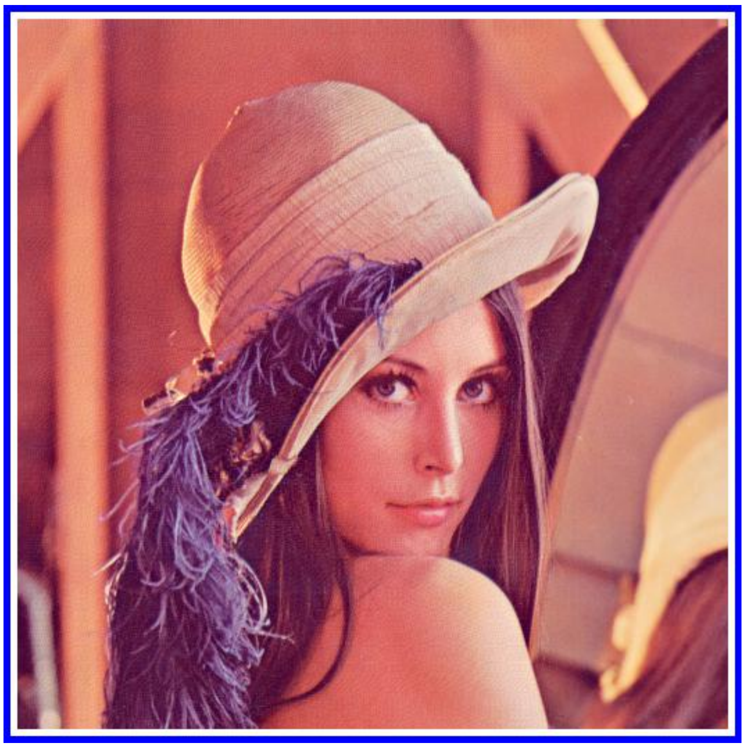


Figure A.14: Watermarked image (method Pra.Framed 1)

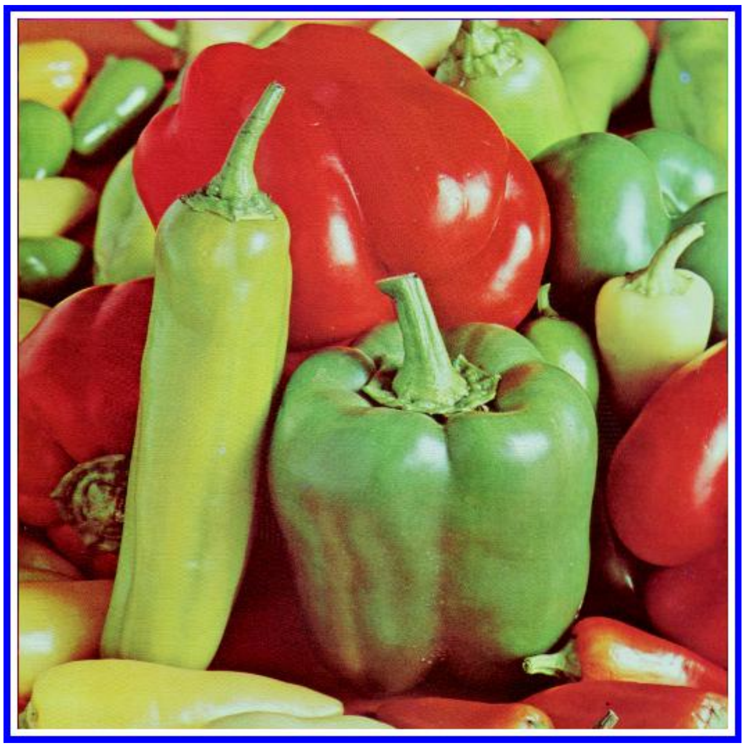


Figure A.15: Watermarked image (method Pra.Framed 0)

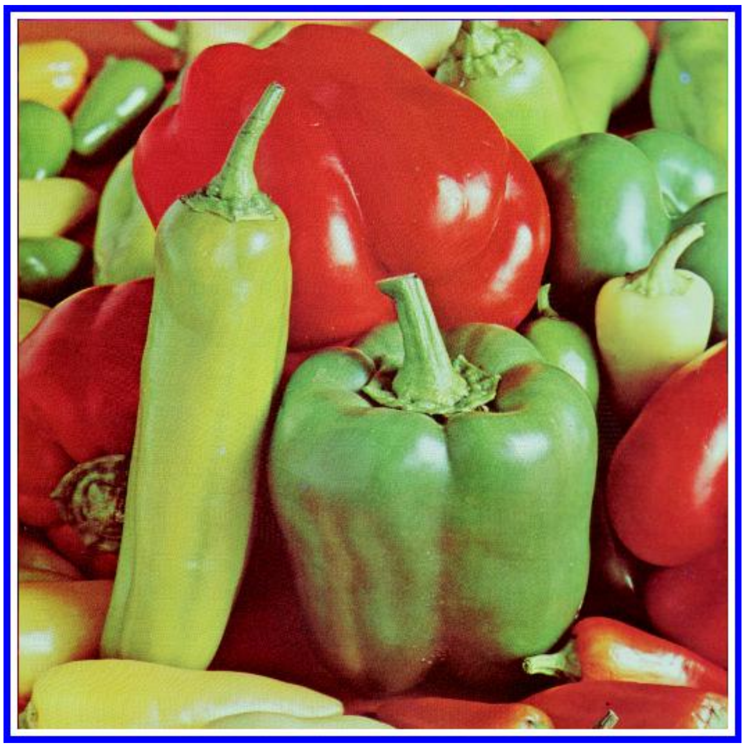


Figure A.16: Watermarked image (method Pra.Framed 1)

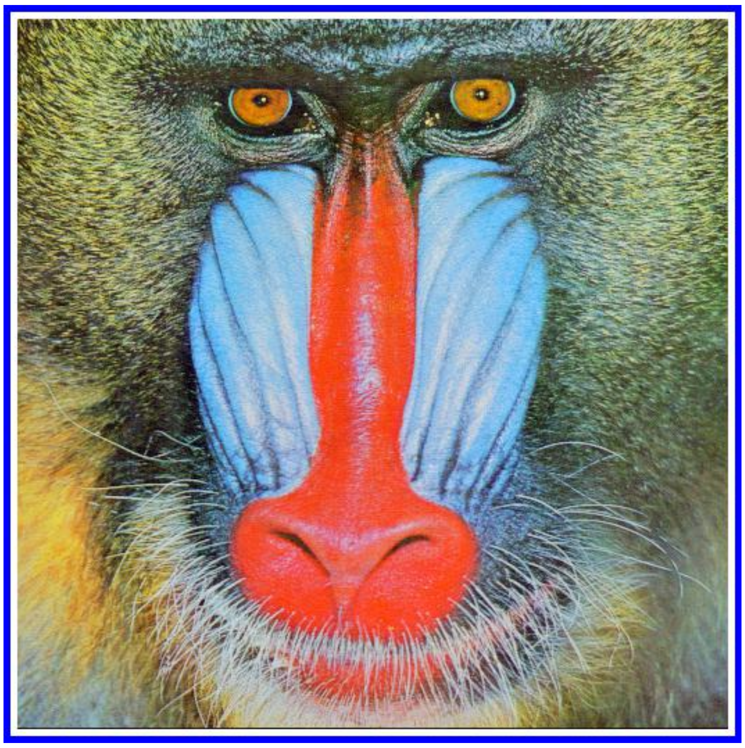


Figure A.17: Watermarked image (method Pra.Framed 0)

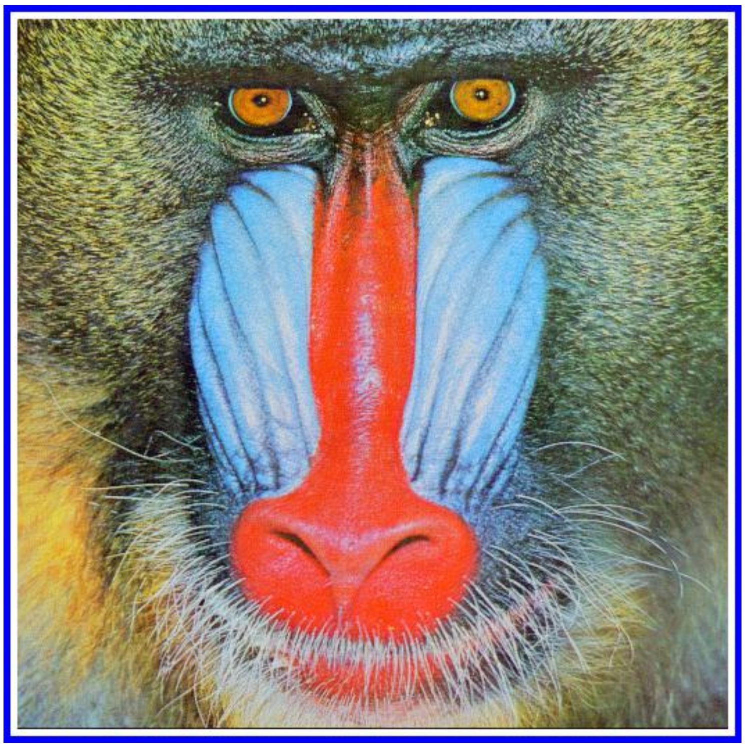


Figure A.18: Watermarked image (method Pra.Framed 1)

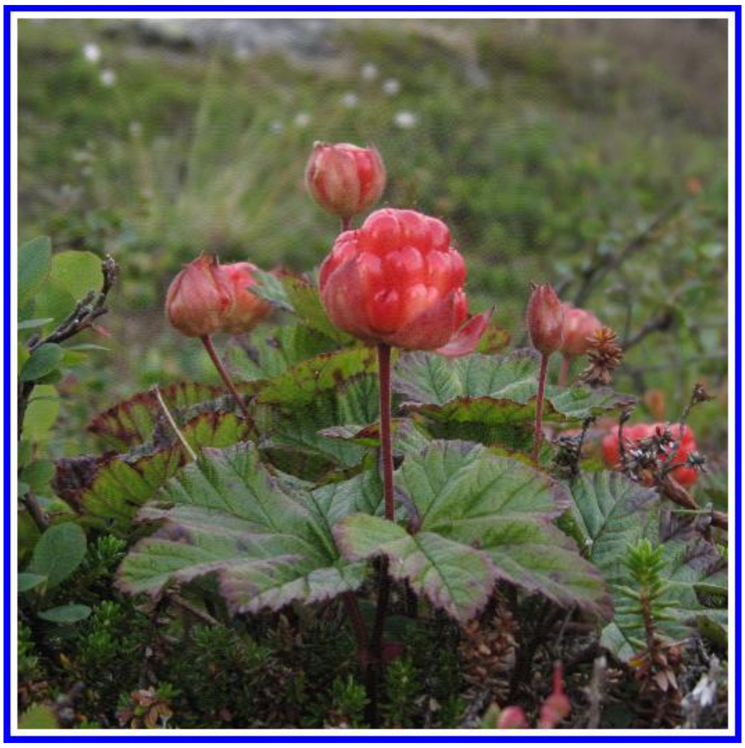


Figure A.19: Watermarked image (method Pra.Framed 0)

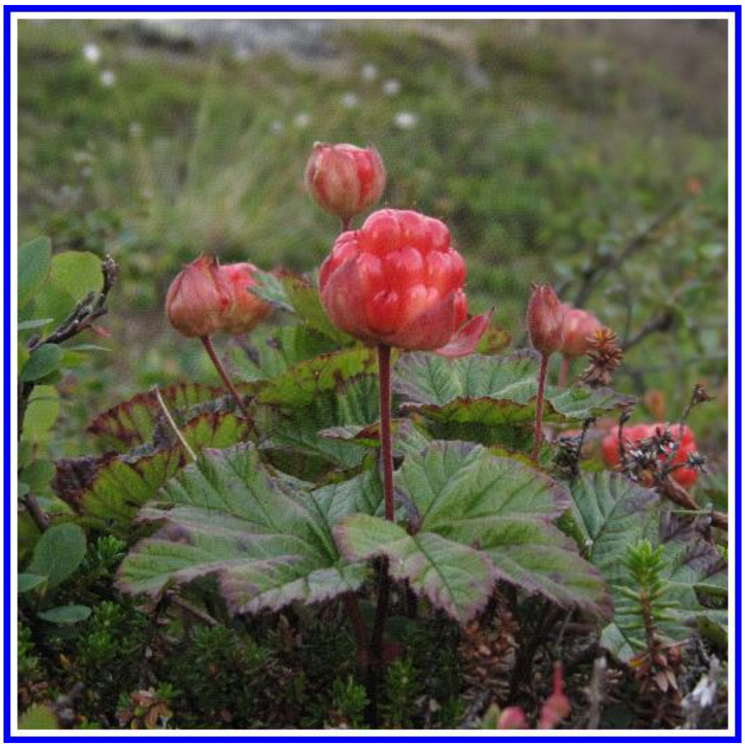


Figure A.20: Watermarked image (method Pra.Framed 1)

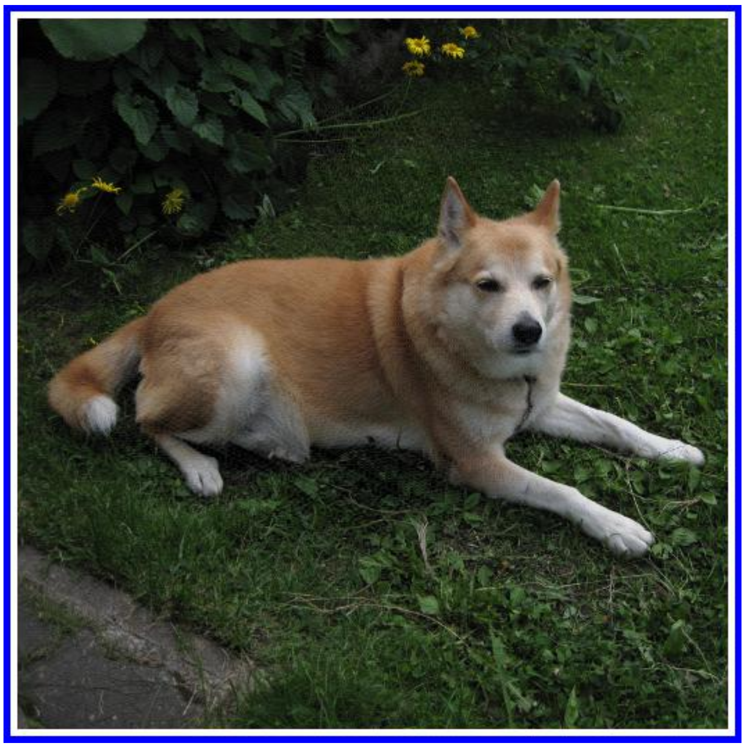


Figure A.21: Watermarked image (method Pra.Framed 0)

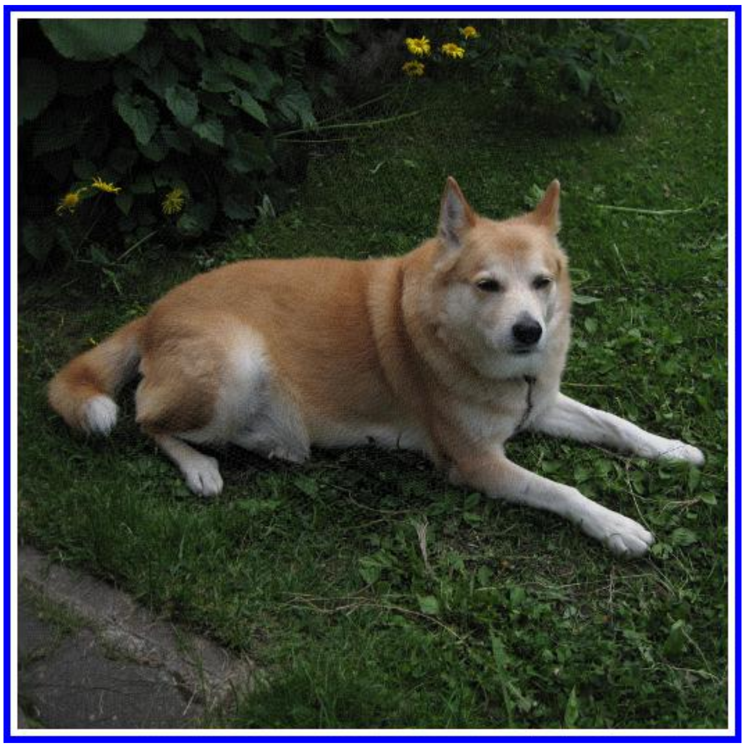


Figure A.22: Watermarked image (method Pra.Framed 1)

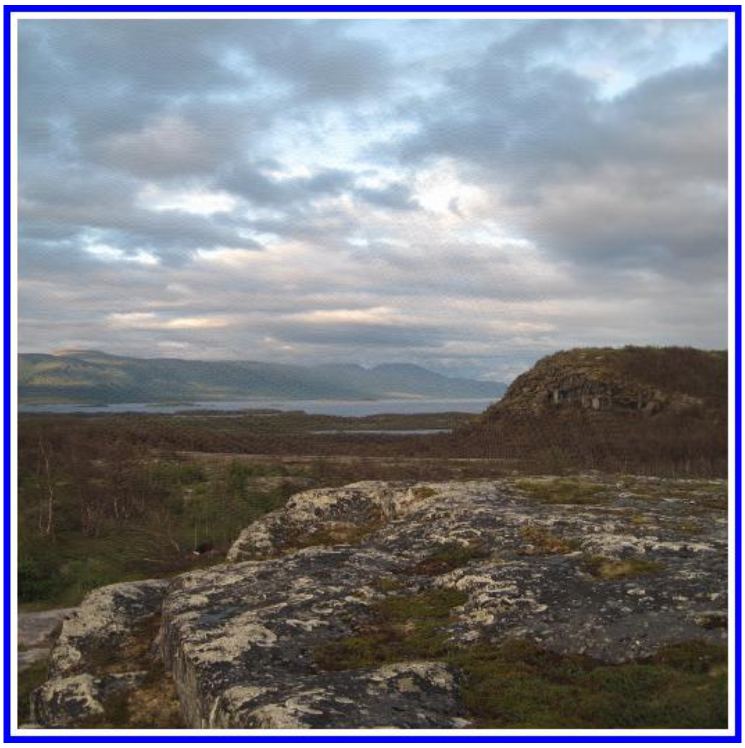


Figure A.23: Watermarked image (method Pra.Framed 0)

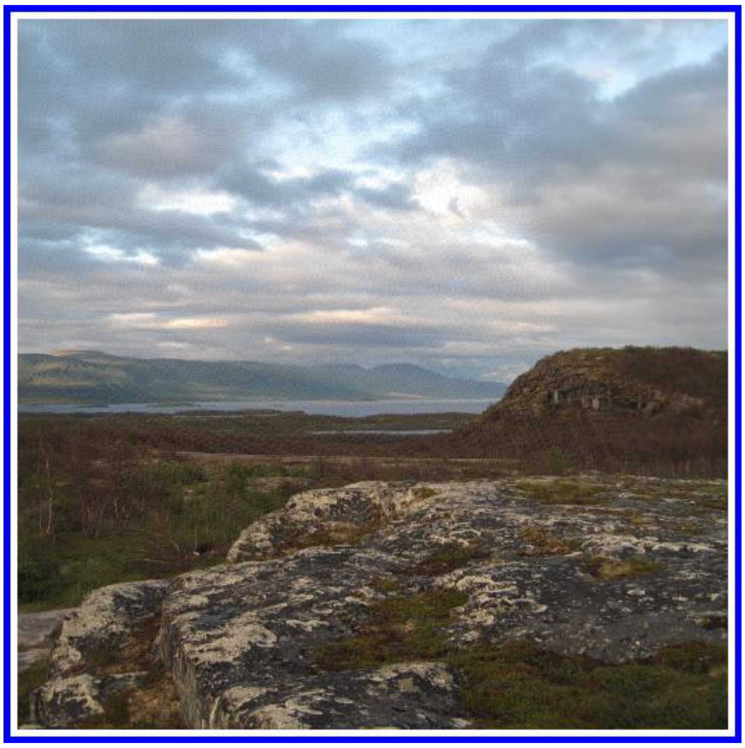


Figure A.24: Watermarked image (method Pra.Framed 1)

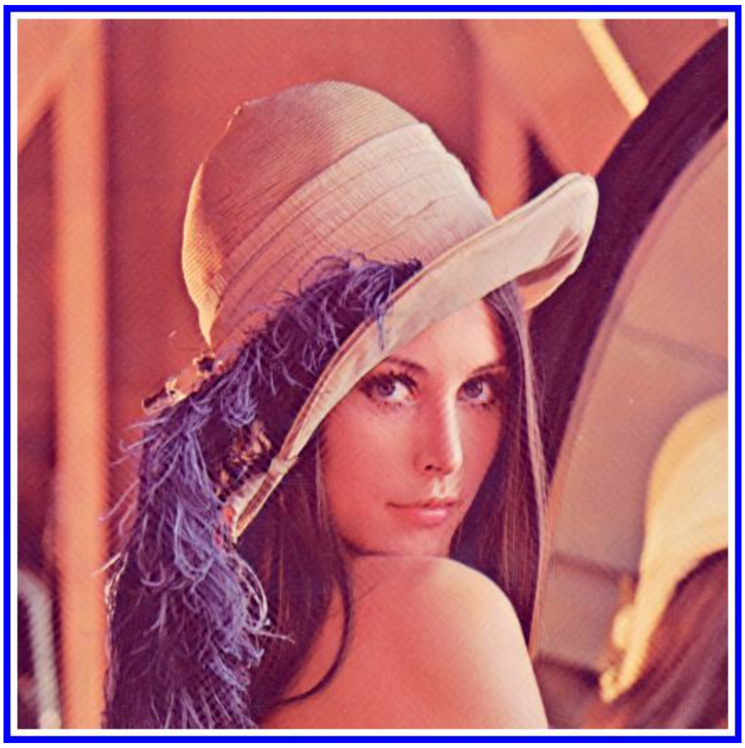


Figure A.25: Watermarked image (method Nakamura, Katayama et al.)

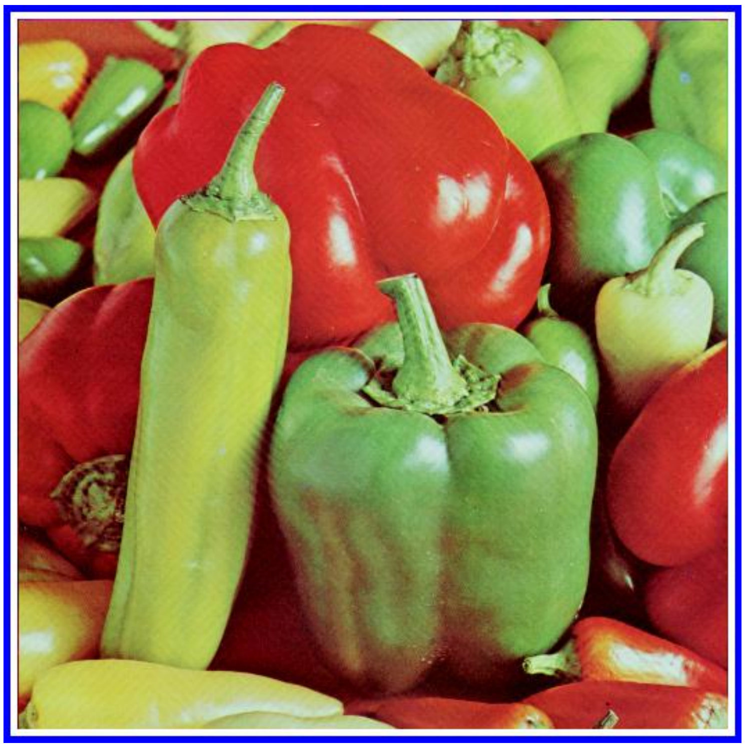


Figure A.26: Watermarked image (method Nakamura, Katayama et al.)

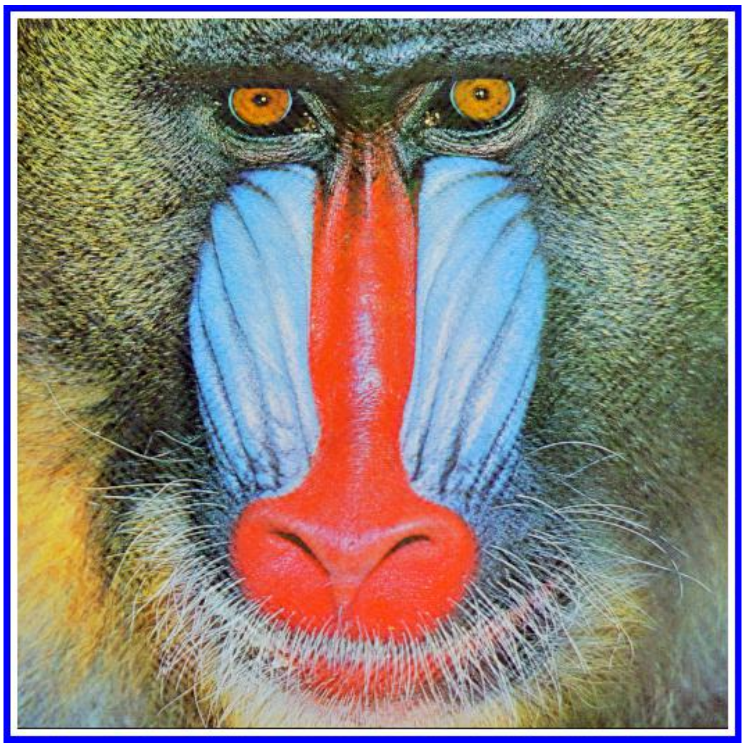


Figure A.27: Watermarked image (method Nakamura, Katayama et al.)

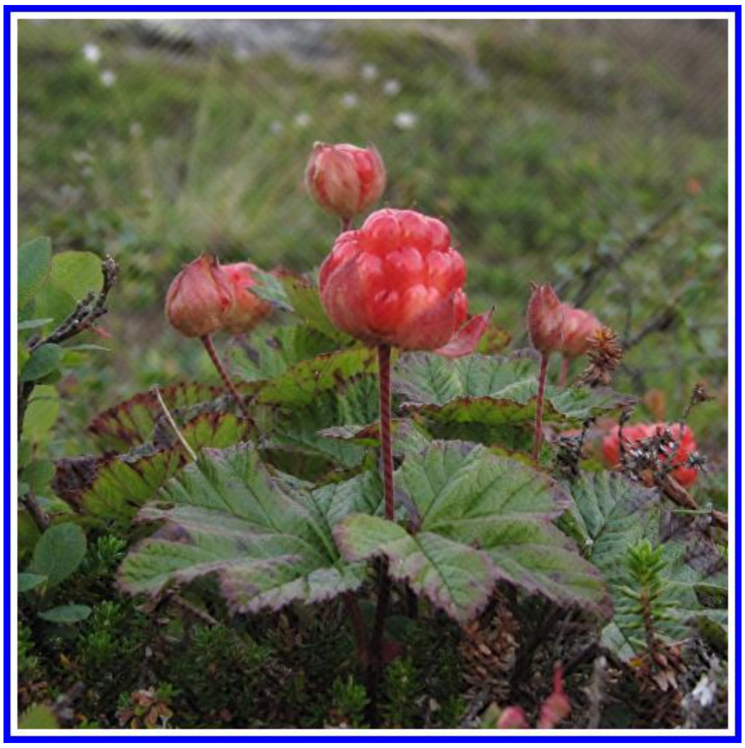


Figure A.28: Watermarked image (method Nakamura, Katayama et al.)

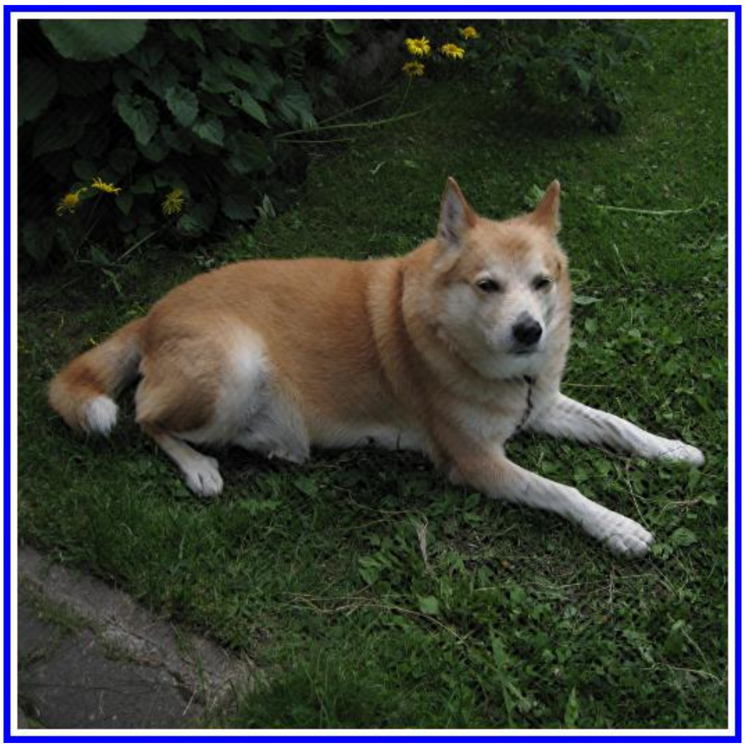


Figure A.29: Watermarked image (method Nakamura, Katayama et al.)

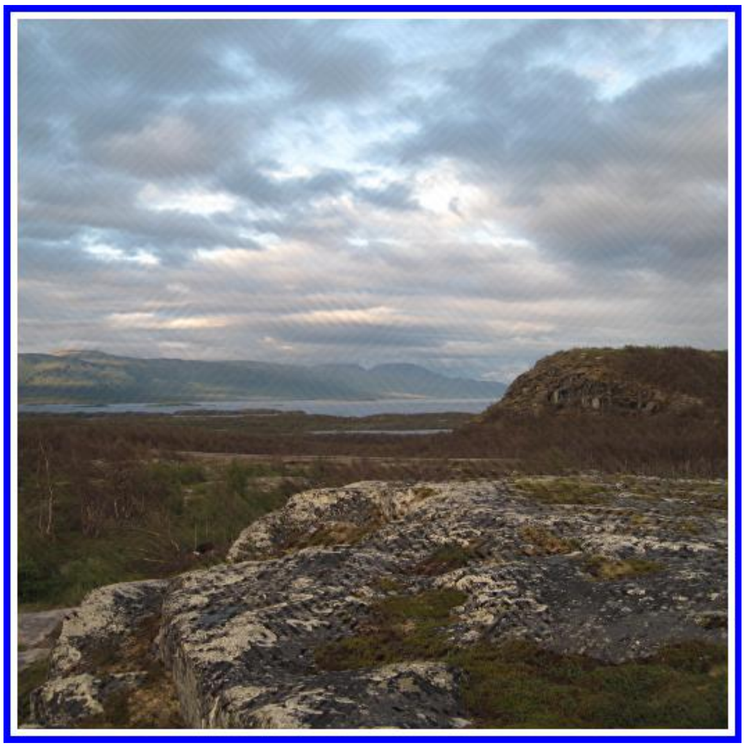


Figure A.30: Watermarked image (method Nakamura, Katayama et al.)