Mandatory project
DM71

Anders Chemnitz

Digital Watermarking

Introduction

The task for this project has been to investigate the field of image watermarking. This includes a general description of the usage of watermarking and the fundamentals for the different approaches that can be taken when using watermarking.

The major part of this project will be about a new method proposed for digital watermarking that utilises neural networks to embed and extract the watermark. This method seems ideal, in that it promises to embed watermarks that cannot be detected by the eye, and being able to extract the watermarks from images exposed to severe alterations.

This project only deals with the digital watermarking of images, though watermarking are also used for video, audio and other digital media.

Digital watermarking seems to be a rising field of reasearch. According to [1] research began in the early 1990's and slowly grew until the millenuim where interest in this field seems to have exploded. This interest might be reflected in an actual demand for efficient tools needed by companies that publishes digital media on the internet.

Fundamentals of watermarking

This section describes the different approaches that watermarking algorithms is based on. The main topics being wether to embed watermaks in the spatial domain or a transform domain. Advantages and disadvanteges are also discussed.

The ideal algorithm

Nothing is perfect, which also holds true for watermarking algorithms. But what if it was possible to construct the perfect or ideal watermarking scheme?

Listed here are a sum up of requirements to such an ideal algorithm, serving as introduction to the topic, but also emphasizing the many contradictions one encounters when dealing with watermarking.

Robust

Since watermarking is primarily used for copyright protection and proving ownership, the embedded watermark has to survive and be extractable after the marked image has been submitted to a variety of things, for example:

- Scaling of the image
- Converting a color image to grayscale

- Blurring, sharpening and other image-effect algorithms
- Lossy compression, for example JPEG, used widely on the internet

Transparent

There are some obvious reasons for wanting to embed the watermark, without being able to see any difference on the marked image contra the original.

Not being able to see the watermark, may keep some people from trying to remove it. If the image is used unrightfully, and your watermark can afterwards be extracted, you have a pretty good case against the copyright violator.

It is also desirable to preserve the quality of an image, even though a watermark is embedded in it. Imagine for example that beautiful pictures promoting a tourist website are severely distorted by the watermarking. Then the algorithm would be practically unusable.

Tamper resistant

Tightly linked to robustness, since any effort made to remove or deteriorate the watermark should result in the watermarked image being severely degraded in quality. There are different approaches for achieving a good level of robustness, which will be discussed later.

Cheap and easy implementation

For a watermarking algorithm to have success, it has to be relatively easy to implement, while not costing a fortune. An algorithm is of no use if it takes a day to mark a picture, and a day to extract the mark again. It has to be *usable in real life* which of course is application dependant.

Robust and fragile watermarks

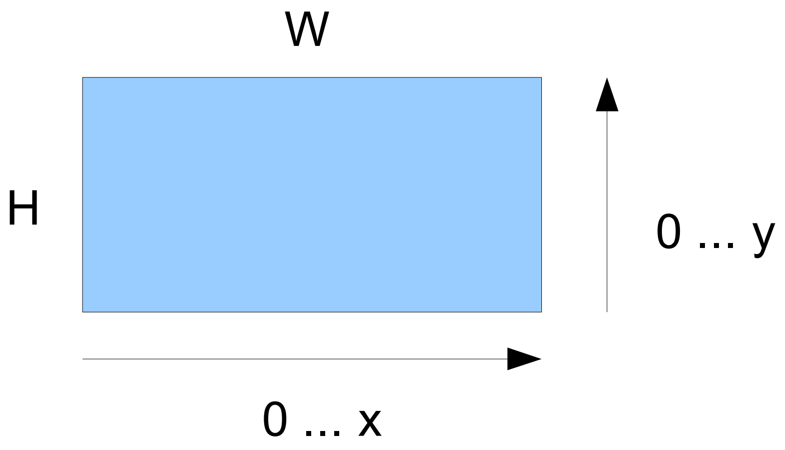
It seems that for most applications, it would be ideal to have a watermark that are able to survive transmission, usage and attacks. Such a watermark is named robust.

On the other hand, watermarks is also used to detect if the image they are in, has been altered. That is watermarks that cannot resist any alteration. Such watermarks is called fragile.

Finally watermarks have been proposed, trying to combine robustness and fragility. That is a watermark that can survive some alterations, but would break if the image was cropped for example, or parts of another image was inserted into it.

Spatial domain

One approach is to embed watermarks in the spatial domain of an image.



The idea is that the number of bits you wish to embed, the watermark size, is stored in the pixels of the image to be marked.

Let us for example want to embed 40 bits of watermark information in a host image. Then we supply a seed to a PRNG (Pseudo Random Number Generator) and uses it to provide us with 40 sets of (x,y) with respect to the illustration, in other words we select 40 pixels.

Each of these pixels are now altered, exchanging a chossen bit of color information with our watermark bit. The choice of bit involves a trade-off:

- If we choose one of the lesser significant bits, our alteration can be held invisible in the marked picture, but image compression and other things might be attacking these less significant bits, which could mean that the watermark is not very robust.
- If a more significant bit is chossen, we would achieve much better robustness, but our alteration would be visible. It would look weird if the sunset picture had 40 black spots in it.

When you would attempt to extract the watermark at a given time, the PRNG would then be started with the same seed, and again produce the 40 pixels, from where we canextract our bits.

This method is easy to use, but has many drawbacks. Even small alterations to our image would destroy at least some of our watermark information. Therefore the method could be used for embedding fragile watermarks.

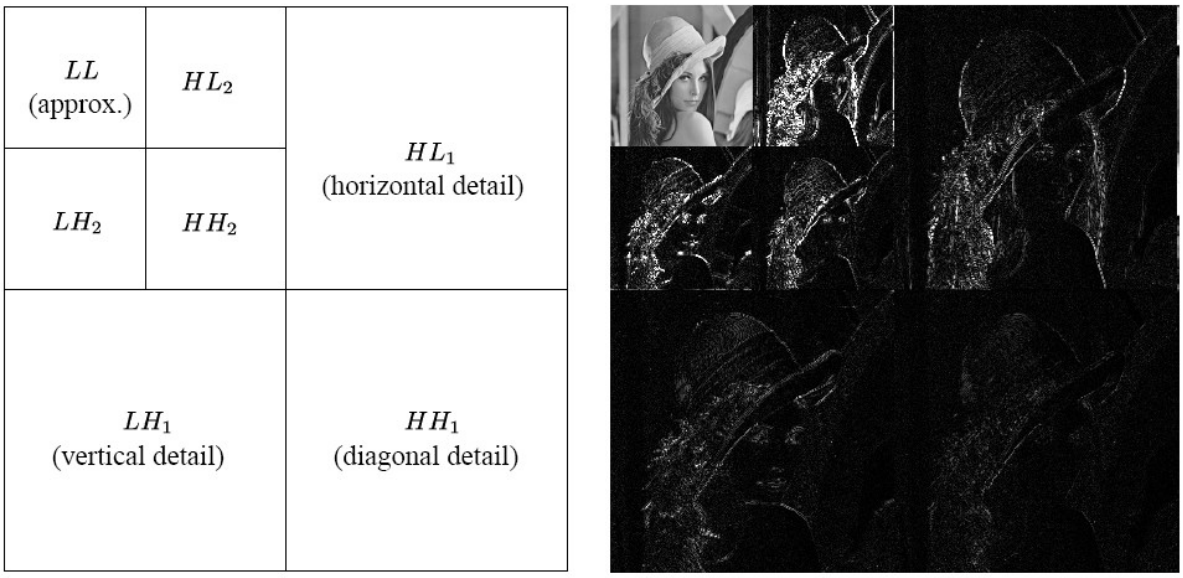
Transform domain

The idea is to subject the image to some mathematical function, that transforms the image into for example levels of different frequenzy.

An example of such a transform is the discrete wavelet transform (DWT). The idea is that a signal is split in a high- and low frequenzy part. The low frequenzy part from this operation can again be split. This process can be continued until the signal is entirely decomposed, or a set level of resolution is reached, say for example five passes.

What is achieved from this proces is a composition of an image in terms of frequencies. The frequenzy is the change in contrast over pixels. The faster the change, and a higher contrast change, gives a higher frequenzy. Large surfaces even in contrast has a low frequenzy. Therefore most images consist mainly of low frequenzy information. The high frequenzy parts are found around edges and textures, where contrast or color changes rapidly.

The result of a DWT is a build up similar to this figure (illustration from [1]):



(a) decomposition structure

(b) decomposed image

A robust method using neural networks

The book [2] proposes a method for doing very robust and imperceptible watermarking, using DWT and neural networks.

Embedding process

1. The image to be marked is decomposed using DWT, to the required resolution level. This resolution level are by [1] suggested to be five for most watermarking applications.

2. The hierarchical decomposition is made into a tree representation, where each node has four children and is associated with a coefficient in the DWT decomposition.
3. A required amount of coordinates is selected from the DWT decomposition. The selection is made with a PRNG which is given a seed k. No coordinates from resolution level 1 is selected, since this will distort the image. To prevent disordering siblings of a node where a coordinate is selected, can no longer be chosen.
4. A training set for the neural network is prepared

   $$T=\{t_k\{x_{k,0},x_{k,1},\cdots,x_{k,7},y_{k,0},y_{k,1},y_{k,2},y_{k,3}\}\}$$
   $$k=0,1,\cdots,\frac{W_H*W_W}{4}-1$$

   This training set consists of eight input vectors, and four outputs. The first four inputs corresponds to the siblings of chosen coordinate, the last four to the siblings of the chosen coordinates parent in the tree. The four outputs are the coefficients corresponding to the chosen coordinates four children.
5. The trained neural network can now be used to embed the watermark. The eight input vectors for each selected coordinate is given to the neural network, resulting in four output vectors. The watermark information is then embedded by replacing the original coefficients with the output from the neural network, adjusted by a constant.
6. To obtain the watermarked image, an inverse DWT is performed.

Extraction phase

1. Transform the watermarked image using DWT.
2. Build the tree representation.
3. Use the seed k to start a sequence with the PRNG. This gives a set of coordinates.
4. The corresponding vectors are fed to the neural network, which results in four output vectors. The difference between the expected output and the actual output of the neural network provides information of the hidden watermark.

Results of using the method

According to [2] there are no visual difference between the original and the marked image. If no alterations has been performed, the watermark extracted has a Bit Correct Ratio (BCR) of 98,87%. The BCR is a per pixel comparison between the original watermark and the extracted.

The embedded image is then ”attacked” in various ways, and listed here is the results:

- JPEG compression: The extracted watermark has a BCR of 88,43%
- Blurring algorithm: Watermark has a BCR of 89,25%
- Sharpening algorithm: BCR of 95,12%
- Resized image after a shrink: BCR of 78,58%

Conclusion

As can be seen in the test results achieved with the neural network and DWT based watermarking algorithm, the extracted watermark, even after heavy alteration like shrinking and resizing, can be clearly recognized. The watermark has over 75% of its pixels in the original locations.

The authors of [2] highlights that this algorithm meets the requirements of modern watermarking systems. A further advantage is that the original image is not needed to extract the watermark.

Litterature

[1]: Master thesis ”Digital Image Watermarking in the Wavelet Transform Domain”

Peter Meerwald, Januar 2001

Available through: http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/MasterThesis

[2]: ”Intelligent Watermarking Techniques” 2004

Jeng-Shyang Pan et.al.

ISBN: 9812387579