Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9/183-2
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18366
Fax: +43 (1) 58801-18392
E-mail: nabil@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/∼nabil/pdf/

PRIP-TR-087 December 12, 2003

Image Compression using Hartley transform[1]

Ahmed Nabil Belbachir, Vienna University of Technology, Austria
nabil@prip.tuwien.ac.at
Ted Chilton, University of Surrey, Guildford, UK
Mark Dunn, Ruhr-Universität Bochum, Germany
dunn@neuroinformatik.ruhr-uni-bochum.de
Marc Nunkesser, Ruhr-Universität Bochum, Germany
marc@nef.wh.uni-dortmund.de
Sahbi Sidhom, University of Claude Bernard, Lyon, France
sidhom@enssib.fr
Georges Szajnowski, University of Surrey, Guildford, UK

Abstract

The report presents a novel approach for image compression using the Hartley transform (HT). The Hartley transform has the advantage of solving the problem of phase wrapping from which the Fourier transform suffers. The magnitude and phase compression using this transformation (HT) have proved better performance than those of the Fourier Transform. Magnitude and phase were processed separately. The quantization of frequency samples in less bits has increased the compression ratio.Furthermore, the distributions used to generate the noise significantly influence the result.
The lossy compression technique used seems not to degrade the image quality. A non-linear filter for smoothing the resulting image would be suitable for image enhancement. In general, the overall compression ratio is acceptable it compresses to about 15-30% the size of the original image. A lossless compression technique could be performed additionally to increase the compression factor.

[1]This Work has been performed in the frame of the ERASMUS Intensive Program at INSA-Lyon France on May 2000.

Contents

1 Introduction **1**
- 1.1 Typical image coder . . . 3
- 1.2 Some Standard Compression Techniques . . . 3
  - 1.2.1 Discrete Cosine Transform Method . . . 3
  - 1.2.2 Joint Photographic Experts Group (JPEG) standard . . . 3
  - 1.2.3 Wavelets and Image Compression . . . 4

2 Hartley Transform **5**
- 2.1 Hartley Transform Properties . . . 6

3 Proposed Techniques **6**
- 3.1 Description of the proposed technique . . . 6
- 3.2 Compression of the magnitude . . . 7
- 3.3 Noise generation . . . 7
  - 3.3.1 Random number generation . . . 9
  - 3.3.2 Quantization . . . 9
- 3.4 Compression of the Hartley phase . . . 9
- 3.5 Implementation . . . 10

4 Experimental Results **10**
- 4.1 Adding noise before quantization vs. quantization without noise . . . 11
- 4.2 Using one bit phase compression of phase vs. scaled phase coding . . . 11
- 4.3 Using rectangular windows for phase compression vs. using diamonds . . . 11
- 4.4 Magnitude compression with logarithmic quantization vs. with uniform quantization . . . 11
- 4.5 Noise generation with chaotic numbers vs. noise generation with random numbers . . . 11
- 4.6 Magnitude compression by noise substitution with different probability distributions . . . 11

5 Conclusion **14**

List of Figures

1 A Typical Lossy Signal/Image Encoder. . . . 3
2 JPEG Encoder Block Diagram. . . . 4
3 (a) Separable 4-subband Filterbank, and (b) Partition of the Frequency Domain 4
4 Illustration of the Hartley Transform 7
5 Flowchart of the proposed techniques (Magnitude and Phase Compression) 8
6 Definition of Regions of Phase Distribution 10
7 Experimental Results on Lena 12

1 Introduction

This report addresses the problems related to still image compression and presents a novel compression technique based on the frequency domain representation. Uncompressed multimedia (graphics, audio and video) data require considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. The recent growth of data intensive multimedia-based web applications not only have sustained the need for more efficient ways to encode signals and images but have made compression of such signals crucial to storage and communication technology. For still image compression, the 'Joint Photographic Experts Group' or JPEG [1] standard has been established by ISO (International Standards Organization) and IEC (International Electro-Technical Commission). The performance of these coders generally degrades at low bit-rates mainly because of the underlying block-based Discrete Cosine Transform (DCT) scheme. More recently, the wavelet transform has emerged as a cutting edge technology, within the field of image compression. Wavelet-based coding provides substantial improvements in picture quality at higher compression ratios. Over the past few years, a variety of powerful and sophisticated wavelet-based schemes for image compression, as discussed later, have been developed and implemented. Because of the many advantages, the top contenders in the upcoming JPEG-2000 standard are all wavelet-based compression algorithms. The numbers in Table 1 show the qualitative transition from simple text to full-motion video data and the disk space, transmission bandwidth, and transmission time needed to store and transmit such uncompressed data.

The examples above clearly illustrate the need for sufficient storage space, large transmission bandwidth, and long transmission time for image, audio, and video data. At the present state of technology, the only solution is to compress multimedia data before them storage and transmission, and decompress them at the receiver for play back. For example, with a compression ratio of 32:1, the space, bandwidth, and transmission time requirements can be reduced by a factor of 32, with acceptable quality.

Compression Principles

A common characteristic of most images is that the neighbouring pixels are dependent and therefore contain redundant information. The fundamental task then is to find less correlated representation of the image. Two basic components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image/video). Irrelevancy reduction omits parts of the signal that will not be perceived by the signal receiver, namely the Human Visual System (HVS). In general, three types of redundancy can be identified:

- Spatial Redundancy or correlation between neighbouring pixel values.
- Spectral Redundancy or correlation between different colour planes or spectral bands.

Multimedia Data	Size/ Duration	Bits/Pixel or Bits/ Sample	Uncompressed Size (B for Bytes)	Transmission Bandwidth (b for bits)	Transmission Time(with 28.8K Modem)
Page of Text	11” x 8.5”	Varying resolution	48 KB 48 KB	32-64 Kb /page	1.1 - 2.2 sec
Telephone quality speech	10 sec	8bps	80 KB	64 Kb/sec	22.2 sec
Greyscale image	512 x 512	8bpp	262 KB	2.1Mb/image	1 min 13 sec
Color image	512 x 512	24bpp	786 KB	6.29Mb/image	3 min 39 sec
Medical image	2048 x1680	12bpp	5.16 MB	41.3 Mb /image	23 min 54 sec
Full-motion video	640 x 480, 1 min (30 fr/s)	24 bpp	1.66 GB	221 Mb/sec	5 days 8 hrs

Table 1: Multimedia data types and uncompressed storage space, transmission bandwidth, and transmission time required

- Temporal Redundancy or correlation between adjacent frames in a sequence of images (in video applications).

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. Since we will focus our attentiononly on still image compression, we will not worry exploit temporal redundancy.

Compression Techniques

There are two ways of classifying compression techniques:

1. Lossless vs. Lossy compression: In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only a achieve a modest amount of compression. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving much higher compression. Under normal viewing conditions, no visible loss is perceived (visually lossless).

2. Predictive vs. Transform coding: In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily

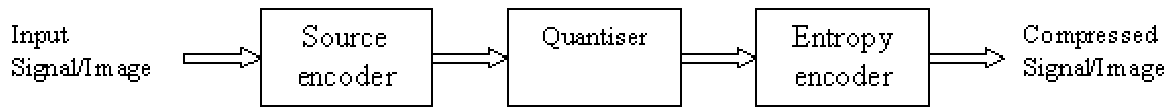


Figure 1: A Typical Lossy Signal/Image Encoder.

adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example of predictive coding. Transform coding, on the other hand, first transforms the image from its spatial domain representation to a different type of representation using some well-known transform and then codes the transformed values (coefficients). This method provides greater data compression compared to predictive methods, although at the expense of greater computational requirements.

1.1 Typical image coder

A typical lossy image compression system is shown in Figure 1. It consists of three closely connected blocs namely, (a) Source Encoder, (b) Quantizer, and (c) Entropy Encoder. Compression is accomplished by applying a linear transform to decorrelate the image data, quantizing the resulting transform coefficients, and (entropy) coding the quantized values.

1.2 Some Standard Compression Techniques

1.2.1 Discrete Cosine Transform Method

The introduction of DCT in 1974 is an important achievement for the research community working on image compression. The DCT can be regarded as a discrete-time version of the Fourier-Cosine series. Unlike DFT, DCT is real-valued and provides a better approximation of a signal with fewer coefficients. The DCT of a discrete signal x(n), n=0, 1, .. , N-1 is defined as:

$$X_t(n) \;=\; \sqrt{\frac{2}{N}}C(u)\sum_{n=0}^{N-1} x(n)\cos\left(\frac{(2n+1)u\pi}{2N}\right) \tag{1}$$

where, C(u) = 0.707 for u = 0, and C(u) = 1 otherwise.

1.2.2 Joint Photographic Experts Group (JPEG) standard

In 1992, JPEG established the first international standard for still image compression where the encoders and decoders are DCT-based. Figure 2shows the key processing steps in the sequential encoding for greyscale images. Colour image compression can be approximately regarded as compression of multiple greyscale images, which are either compressed entirely one at a time, or are compressed by alternately interleaving 8x8 sample blocks from each in turn.

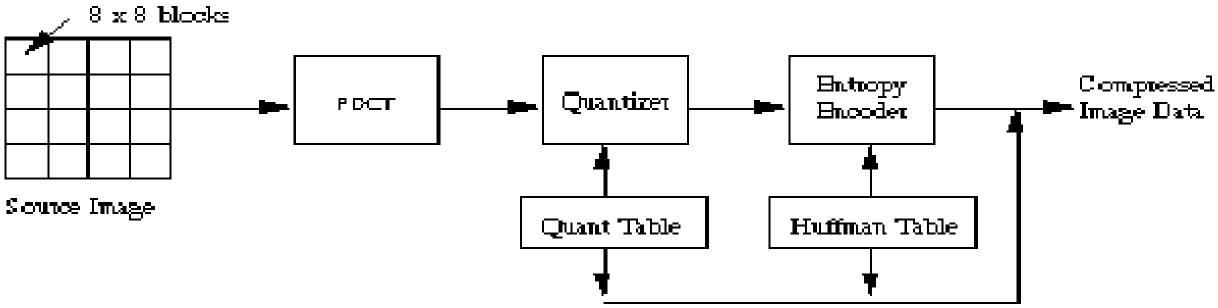
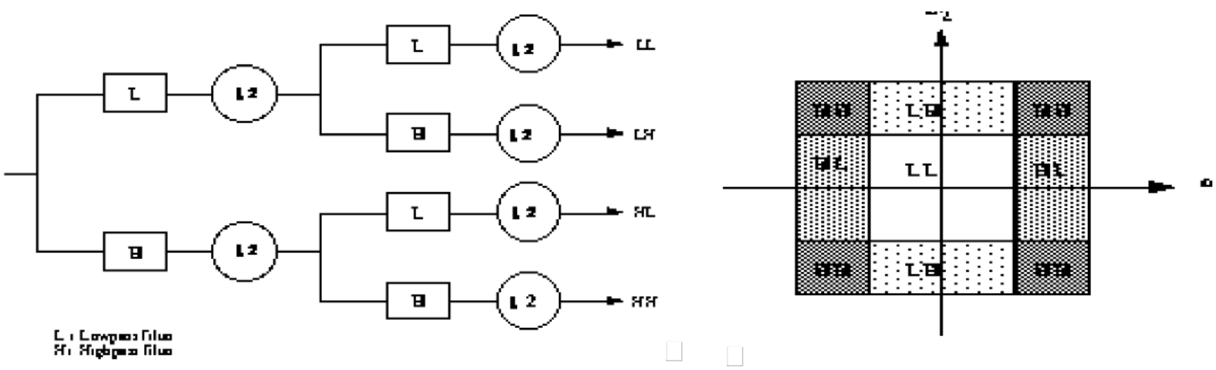


Figure 2: JPEG Encoder Block Diagram.



(a) **(b)**

Figure 3: (a) Separable 4-subband Filterbank, and (b) Partition of the Frequency Domain

1.2.3 Wavelets and Image Compression

Since the input image needs to be divided into blocs using JPEG compression, correlation across the block boundaries is not eliminated. This results in noticeable and annoying "blocking artifacts" particularly at low bit rates. Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. In many applications wavelet-based schemes (also referred as subband coding) outperform other coding schemes, such as the one based on DCT. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compression avoid blocking artifacts. Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. The fundamental concept behind Subband Coding (SBC) is to split up the frequency band of an image and then to code each subband using a coder with bit rate accurately matched to the statistics of the band. SBC has been used extensively because of its inherent advantages resulting from variable bit assignment among the subbands as well as coding error confinement within the subbands.

Figure 3-(a) Separable 4-subband Filterbank, and Figure 3-(b) Partition of the Fre-

quency Domain In the following sections, a novel approach for still image compression is proposed using the spectral decomposition called Hartley Transform.

2 Hartley Transform

Image processing as used here means improving images or pictures and generating special pictorial effects. These operations rely heavily upon matrix algebra used to implement, Fourier Transform, Hartley transform, etc. However, an attempt will be made to describe some of the basic operations utilized for image processing with the minimum of mathematics. The general principle of developing a compressing technique in our case is to digitize a picture into a small number of bits. The Fourier analysis is used to convert a picture into an equivalent form which can be more easily processed, filtered, compressed, etc., using suitable algorithms for manipulating its contents, particularly when including filters. In the case of the orthogonal transforms, Fourier transform is one of the transforms, with the following properties :

1. It must be reversible
2. It must be unique : one-to-one mapping
3. The two domains should be orthogonal

$$W(t,f) \;=\; e^{-j2\pi ft} = \cos(2\pi ft) \;-\; j\sin(2\pi ft) \tag{2}$$

To observe the similarity and the property of data (correlation) :

$$\int_{-\infty}^{\infty} s(t)\cos(2\pi ft)\;dt \;=\; S_R(f) \tag{3}$$

$$\int_{-\infty}^{\infty} s(t)\sin(2\pi ft)\;dt \;=\; S_I(f) \tag{4}$$

: REAL part of Fourier Transform : IMAGINARY part of Fourier T.

The complex Fourier spectrum is defined as:

$$S(f) = S_R(f) \;-\; jS_I(f) \tag{5}$$

The picture data are represented in two parts with the Fourier Transform approach :

1. Magnitude

$$M(f) \;=\; |S(f)| \;=\; \sqrt{S_R^2(f) \;+\; S_I^2(f)} \tag{6}$$

2. Phase

$$\phi(f) \;=\; \arctan\frac{S_I(f)}{S_R(f)} \tag{7}$$

The major problem of Fourier Transform is a computational limitation resulting from the phase spectrum (Phase Wrapping) : linearity that progresses to infinity. To resolve this computational problem related to the phase spectrum of Fourier transform, two types of solutions may be used :

1. the first solution consists of an algorithmic approximation to the phase,
2. the second solution consists to extend the Fourier transform with a best solving of its phase problem. Therefore, Hartley Transform was proposed.

2.1 Hartley Transform Properties

The Hartley Transform requires new transform operations to extend the Fourier approach :

$$W_H(t,f) \;=\; \cos(2\pi ft) \;+\; \sin(2\pi ft) \tag{8}$$

$$S_H(t) \;=\; \int_{-\infty}^{\infty} s(t)W_H(t,f)\,dt \;=\; S_R(f) \;+\; S_I(f) \;=\; M(f)\,(\cos\phi(f) \;+\; \sin\phi(f)) \tag{9}$$

$$S_R(f) \;=\; M(f)\cos\phi(f) \tag{10}$$

$$S_I(f) \;=\; M(f)\sin\phi(f) \tag{11}$$

The advantage of the Hartley Transform results from the function:, that contains information about the phase without a computational complexity.

The picture data are represented by the Hartley Transform as follows:

1. Magnitude :
$$M(f) \;=\; |S(f)| \;=\; \sqrt{S_R^2(f) \;+\; S_I^2(f)} \tag{12}$$
2. Phase : expressed in V(f)
$$V(f) \;=\; \frac{S_H(f)}{M(f)} \;=\; \cos\phi(f) \;+\; \sin\phi(f) \tag{13}$$

Figure 4 depicts an illustration for computing the Hartley transform based on the Fourier transform in one dimension.

3 Proposed Techniques

3.1 Description of the proposed technique

The parameters from an image obtained from Hartley transform are magnitude and phase. However, those two parameters are processed separately. The operations constituting the proposed techniques are shown in figure 5.

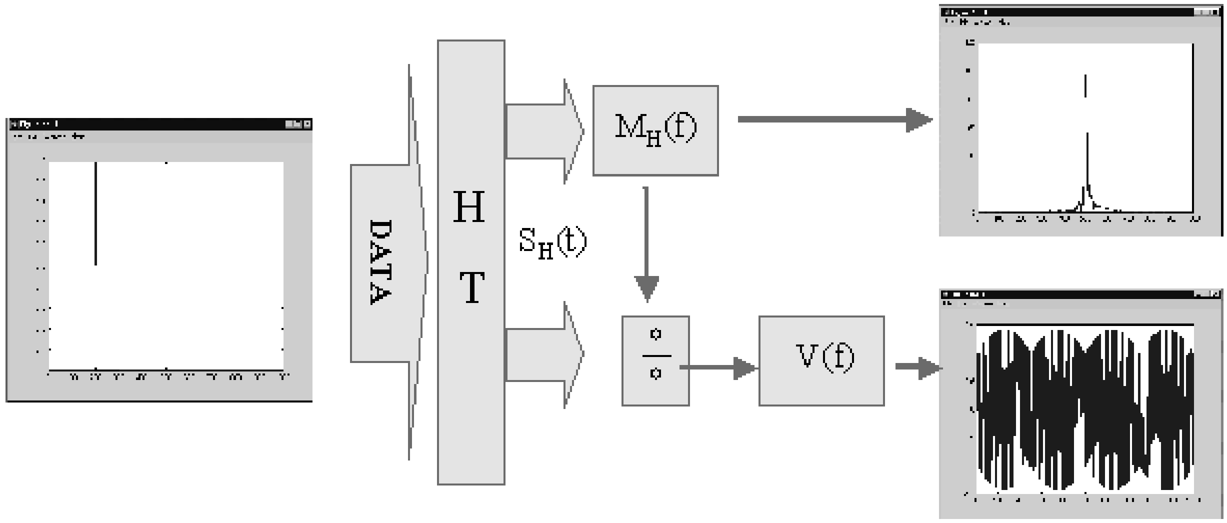


Figure 4: Illustration of the Hartley Transform

3.2 Compression of the magnitude

The magnitude is compressed by combining two methods:

1. Substitution of blocks by noise of a specific distribution.

2. Quantization.

The entire image is divided into square blocks with the size L. In each block the maximum value is determined and compared to a threshold t. If the value is smaller than the threshold, the first method is applied, otherwise the second one is used.

3.3 Noise generation

Each block has to be substituted by noise. The coordinates ($2 \cdot 10$ bits) and the parameters ($2 \cdot 8$ bits) of the distribution which are derived from the data in the block are stored. Four different distributions are considered

1. Zero-data distribution. All data in the block are set to zero

2. Uniform distribution. Adapt the distribution to the data by calculating the base b and the scale factor s where b is the minimum value in the block and s is the difference between the maximum and the minimum value.

3. Exponential distribution.

4. Rayleigh distribution. After having analyzed several histograms of blocks in real images, we observed their resemblance to the exponential or Rayleigh distribution.

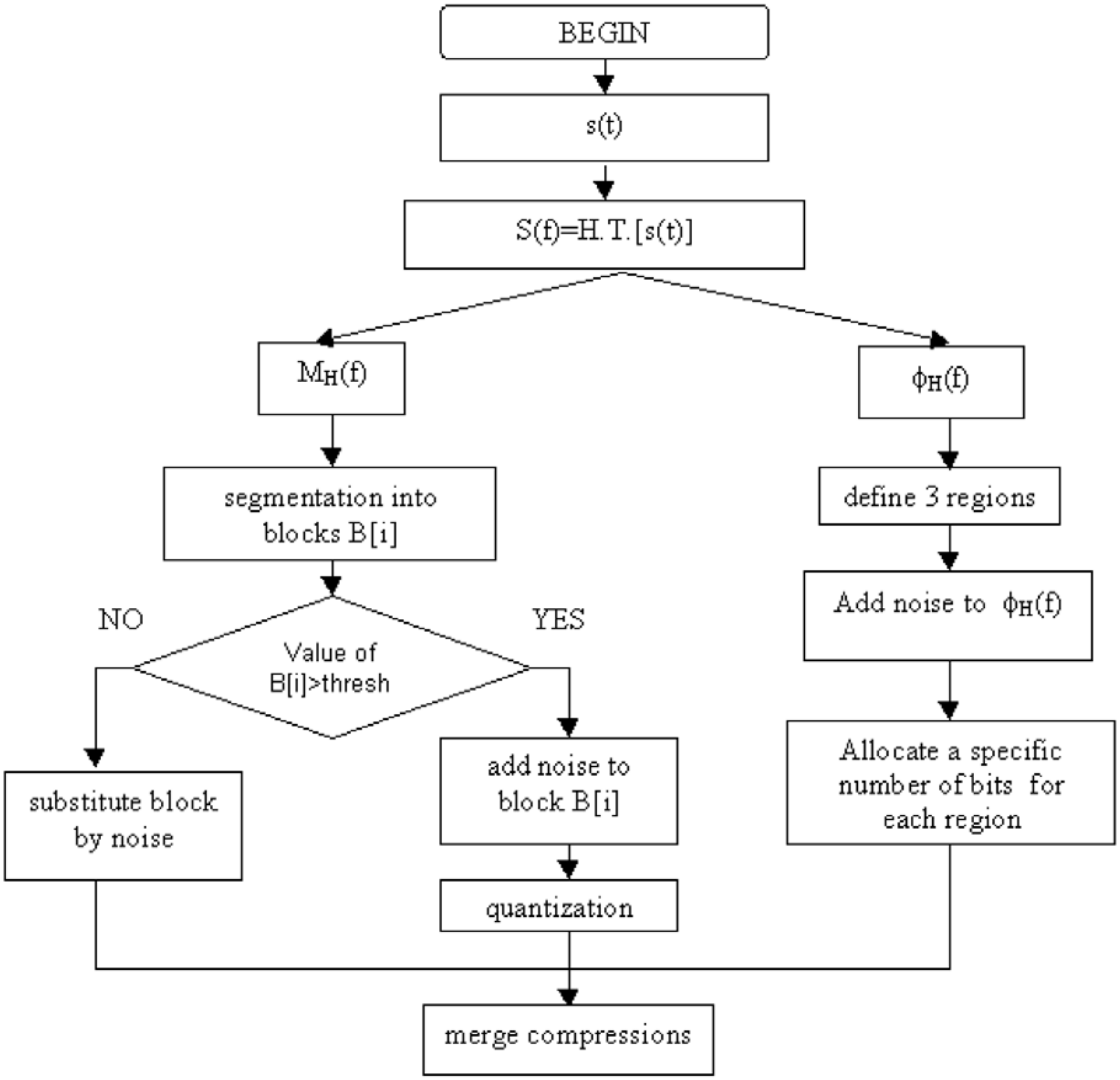


Figure 5: Flowchart of the proposed techniques (Magnitude and Phase Compression)

However, the number of samples is not sufficiently large to justify this statement. But it is interesting to see the effect on some reference pictures. The parameters for both distributions are the scaling factor a, which is the mean of the data, and the offset b, which is the minimum of the data.

3.3.1 Random number generation

Instead of using Matlab standard random number generator for uniform distribution we use a chaotic number generator, based on the tent map function. In a real implementation this could decrease the computational complexity of the compression algorithm, if the chaotic number generator produces results similar to a uniform number generator. The chaotic number generator can be implemented in hardware very easily. The histogram and the correlation of one million samples show the characteristic of a uniform distribution.

3.3.2 Quantization

The values of the blocks, which are not substituted by noise, are represented by fewer bits. Before the quantisation uniformly distributed noise with an amplitude equal to the LSB is added. This idea is derived from signal processing technique where an n bit quantiser can be replaced by an m bit quantiser (m¡n), if noise is added to the signal and then is oversampled. After suitable averaging, one will get a similar result for both techniques. For sampling, a logarithmic scale is used. After having observed the magnitude spectrum of several images, it is concluded that a logarithmic scale seems more appropriate than uniform one.

3.4 Compression of the Hartley phase

For Hartley phase, a different approach is chosen, since the information in the phase is more important and it has some characteristics we want to exploit. Hartley phase has the property of being constrained to the interval of $[-\sqrt{2}, \sqrt{2}]$ and its variance is 1. Therefore, one approach is to code every value by one bit, and to set it to 1 or -1 for decompression so that the variance is preserved. Another method is to code the lower frequency phase with more bits than the higher frequency phase, what we call scaled phase coding. As the phase at low frequencies is more important than the phase at high frequencies, this could be a promising approach. A disturbance of higher frequencies will just result in a slight blur of edges, whereas a disturbance of lower frequencies will probably corrupt the picture. We implemented the one bit-encoding scheme, which is quite straightforward as well as the variable quantization. For the latter case, three areas of constant quantization are specified. Hartley phase is calculated then, a sampling rate in the frequency domain is chosen such that the number of samples is equal the number of pixels in the picture. Hartley phase is represented in a rectangular array that has exactly the dimensions of the original picture. In this array the low frequencies are at the edges and the high frequencies are in the middle. The areas are defined either by two rectangles or by two diamonds as sketched in Figure 6. The position of the borders of the areas are kept variable, in the

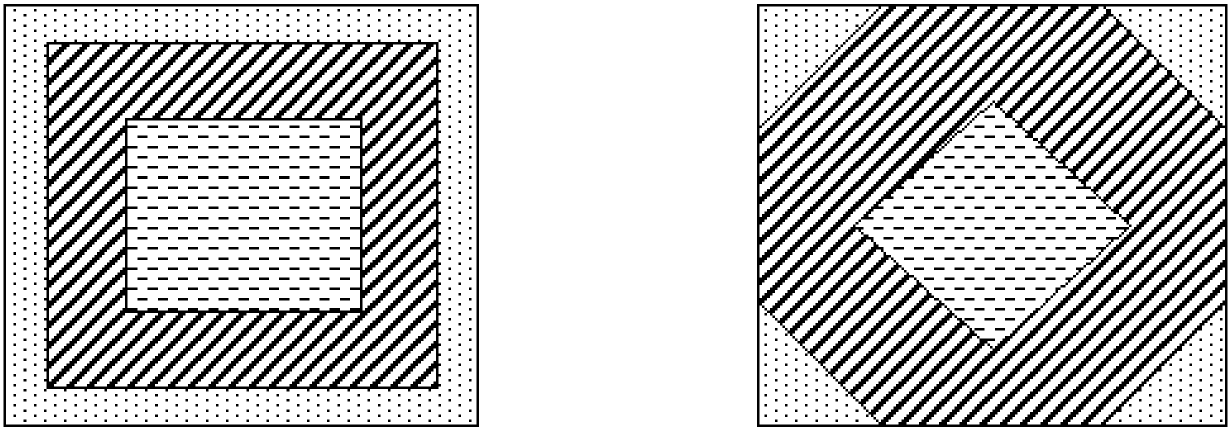


Figure 6: Definition of Regions of Phase Distribution

current implementation they can be set interactively by the user. As in the case of the magnitude we add uniformly distributed noise in the order of the LSB. For decompression the bits that were lost by the quantisation are also randomized.

3.5 Implementation

The Matlab implementation of the software was performed. The compression and decompression of the image is done in one step. This is equivalent to simply destroying all information that would be lost, if the proposed technique would be used to compress the image, transmit it over a channel and decompress it. The exact compression rate is calculated. As for the magnitude, the number of replaced blocks rb is counted as follows:

$$rb = \Big(|xPosition| \ + \ |yPosition| \ + \ \sum |parameters| \Big) \tag{14}$$

where $|x|$ is the length of the bit representation of x. For the uniform distribution, $|xPosition| = |yPosition| = 10\ bit$. $\sum(|parameters|)$ is 16, since we code each of the parameters by 8 bit. For the quantization of magnitude and phase the calculation of the necessary bits is straightforward. If this technique would really be used, one needs a program that really produces the compressed representation, which is only implicitly contained in this program. Still the quality of the transmitted image and the compression rate would be exactly the same as in our emulation.

4 Experimental Results

Different experiments to test the software modules are performed. As the quality of a compressed image is a highly subjective measure, we did not try to map the quality to some pseudo-objective error measure; in fact, we relied on our own judgement. These are the experiments we carried out:

4.1 Adding noise before quantization vs. quantization without noise

Images that were compressed using the extra noise before quantization looked more blurred than those which were directly quantized. Thus, a method that yields good results when directly applied to image data gives poor results when applied to magnitude and phase quantization.

4.2 Using one bit phase compression of phase vs. scaled phase coding

The one bit phase compression resulted in very bad images; it is not appropriate in our case. The scaled phase coding performed much better. For the same compression rate (1/8) the image quality is acceptable. Generally, we used 0 bits for the phase in high frequencies, which allows for a more accurate encoding of the phase of low frequencies.

4.3 Using rectangular windows for phase compression vs. using diamonds

According to the idea that the most important phase information in the frequency domain is at the edges of our phase array, diamonds are expected to perform better than rectangles, because they represent the transition from low to high frequencies more accurately. In fact, the difference between the two partitions is perceptible. The diamonds partition gives slightly better results.

4.4 Magnitude compression with logarithmic quantization vs. with uniform quantization

There is a significant difference between logarithmic scaling and uniform scaling, as one might expect because the typical magnitude histogram resembles much more an exponential distribution than a uniform one.

4.5 Noise generation with chaotic numbers vs. noise generation with random numbers

Pictures compressed with either chaotic or random numbers cannot be distinguished, so that the use of chaotic numbers for this compression scheme is a possible choice.

4.6 Magnitude compression by noise substitution with different probability distributions

The quality of the compressed images can be improved significantly by choosing the appropriate probability distribution for the noise. Several tests are made using a set of images.

Figure 7: Experimental Results on Lena

A typical example is the image of Lena. In order to test the magnitude compression we turned off the quantization. Thus, the magnitude is compressed to about 15% (combined with quantization the compression would have been better). In Figure /reffig-lena, there is the original image(top-left), compressed image with uniformly distributed noise (top-middle), compressed image without noise (top-right), compressed image with exponentially distributed noise(bottom-left), and compressed image with noise according to the Rayleigh distribution (bottom-right). The compressed pictures were (subjectively) ordered by increasing quality. In general the Rayleigh distribution gives good results whereas a uniform distribution performs quite poorly at higher compression rates. Setting the noise level to 0 yields blurred pictures.

Using different block sizes and threshold values for decision between two magnitude compression schemes. It is clear that the higher one sets the threshold value, the higher the resulting compression rate will be, since more and more blocks get substituted.

Table /reftab:exp represents the experimental results of the compression software on a set of images. It is clear that the compression ratio depends on the image. In fact, the software performance seems to be promising. Conclusion This report presents a novel

Picture (Size)	Magnitude Compression Ratio	Phase Compression Ratio	Observation/Remarks
Lena (228x228)	0.31 (threshold 14000 blocksize 20)	1.00	no quantization, with different distributions none : blurred uniform : blurred, additional texture exponential : a lot of additional texture Rayleigh : additional texture
Olympia (530x460)	0.16 (threshold 14000 blocksize 20)	1.00	as Lena, but Rayleigh and exponential distribution do not differ much
Lena (228x228)	0.15 (threshold 50000 blocksize 20)	1.00	no quantisation, with different distributions none : grainy and blurred uniform : severly distorted exponential : very grainy, a lot of additional texture Rayleigh : grainy, additional texture
Lena (228x228)	0.22 (0.17) (threshold 21150 blocksize 20 8 bit (6 bit))	1.00	with quantisation and Rayleigh distribution quite ok (only additional texture)
Lena (228x228)	0.11 (threshold 21150 blocksize 20 4 bit)	1.00	with quantisation and Rayleigh distribution: very poor quality
Lena (228x228)	0.11 (threshold 50000 blocksize 20 6 bit)	1.00	with quantisation and Rayleigh distribution: decreased but tolerable quality
Santana (327x210)	1.00	0.28 (0/0/8 (inner /middle /outer) rectangle)	rectangular regions for phase encoding: a lot of additional texture
Santana (327x210)	1.00	0.28 (0/0/8 rectangle)	rectangular regions for phase encoding: a lot of additional texture
Santana (327x210)	1.00	0.26 (inner bits: 0 middle bits: 0 outer bits: 8 rhombus)	rhombical regions for phase encoding: additional texture
Lena (228x228)	1.00	0.10 (0/2/4 rhombus)	rhombical regions for phase encoding: additional texture
Olympia (530x460)	0.092 blksize 20 threshold 20000	0.084 (0/2/4 rhombus)	good quality

Table 2: Experimental Results on Several Images

approach for the image compression using the Hartley Tranform. The magnitude and phase compression using this transformation has proved a good performance. The lossy compression technique used, seems to not degrade as well the image quality. A non-linear filter for smoothing the resulting image would be suitable for the image enhancement. In general, the overall compression ratio is acceptable. A lossless compression technique could be performed additionally to increase more the compression factor. As future work, this compression method will be used for the complete Hartley frequency spectrum instead of separately processing the magnitude and the phase.

5 Conclusion

The report presents a novel approach for image compression using the Hartley transform. The magnitude and phase compression using this transformation have proved a good performance. Magnitude and phase were processed separately. The quantization of frequency samples in less bits has increased the compression ratio. Furthermore, the distributions used to generate the noise influence the result significantly. The lossy compression technique used seems not to degrade the image quality. A non-linear filter for smoothing the resulting image would be suitable for image enhancement. In general, the overall compression ratio is acceptable it compresses to about 15-30% the size of the original image. A lossless compression technique could be performed additionally to increase the compression factor. More levels of different bit length would probably improve the results. As future work, this compression method could be used for the complete Hartley frequency spectrum instead of separately processing magnitude and phase.

References

[1] ISO/IEC/JTC1/SC29/WG1 N390R, JPEG 2000 Image Coding System, Mar. 1997, http://www.jpeg.org/public/wg1n505.pdf.

[2] Azriel Rosenfeld, Avinash C. Kak, Digital picture processing, volume 1, second edition, in Computer science and applied mathematics,1982.

[2] http://ourworld.compuserve.com/homepages/JDebord/fourier.htm

[3] http://www.ese-metz.fr/metz/eleves/themes/imagerie/fourier/som.html

[4] http://www.prism.uvsq.fr/ hpc/Enseignement/InfoMedia/Cours/node1.html

[5] http://www.aei.ca/ sylvain/TFourier/tmatiere.html

[6] http://www.iro.umontreal.ca/ pigeon/Ondelettes/TransformeesRapides.html

[7] http://mathworld.wolfram.com/HartleyTransform.html

[8] http://www.iro.umontreal.ca/ pigeon/Ondelettes/Hartley/Hartley.html