An improved image processing chain for mobile terminals

Graduate Thesis

Siv.ing. Degree in
Information and Communication
Technology

Submitted to
Agder University College
Faculty of Engineering

By

Tommy Olsen and Jo Steinar Strand

Grimstad, May 2002

Abstract

Based on the proposal in the report by Tryggve Mikkelsen [Mikkelsen, 2001] this thesis investigates and discusses the implementation of a simplified image processing chain for modern mobile terminals. Previous work on image processing in this field of application has shown that the demosaicing operation, color space conversion and downsampling are resource demanding pre-processing operations. The simplification potential of the signal chain lies within the pictured scenario of using Bayer RGB from the image sensor directly as input to the compression scheme. By omitting these pre-processing operations, unnecessary use of limited resources of a mobile terminal is prevented, making it capable of reducing the computational costs.

The work has shown that the simplified image processing chain can be implemented two different ways. Either by using Bayer RGB as it appears from the image sensor as input to the compression scheme or by using composite Bayer RGB where the colors red, green and blue are distinguished into three arrays.
Both implementations reveal overall better results for SNR compared to classical JPEG when lossless compression is applied. Furthermore, both implementations are capable of better color reproduction than JPEG. Comparing the two implementations, however, Bayer RGB appears to be capable of the best overall performance. In fact, taken into account the expected computational savings of the simplified image processing chain, conventional JPEG falls short on most evaluation criteria in this work compared to Bayer RGB.

Preface

As a part of the ICT (Information and Communication Technology) program at Agder University College, this report is carried out as a partial fulfilment of the Siv.Ing. degree. The Siv.Ing. degree is by Agder University College considered to be equivalent to the international Master of Science degree. The thesis was performed for Ericsson AS in the period from 2002-01-10 to 2002-05-28.

We would like to thank our supervisors, Siv.Ing. Tryggve I. Mikkelsen at Ericsson AS and Assistant Professor Ola Torkild Aas at Agder University College for valuable support and advice during the work.

Grimstad, May 2002

Tommy Olsen *Jo Steinar Strand*

Table of contents

List of figures

List of equations

1 Introduction

1.1 Background

In the existing mobile communication technologies GSM, GPRS and the emerging UMTS, the development of applications tend to point towards multimedia functionality. The newly introduced property of GPRS to make the users capable of sending and receiving still images through MMS (Multimedia Message Services), and the opportunity users of GPRS have to download images and possibly videoclips from the Internet to be viewed on their mobile terminals, prove the convergence of multimedia and mobile communication technologies. Furthermore, the adaptation of digital cameras suitable to mobile terminals emphasises on the emerging trend in the world of mobile communication.

Given the limited amount of resources like battery capacity, processing power and transmission speed associated with a mobile terminal, one of the main concerns in the effort towards multimedia on mobile terminals is the implementation of an efficient image processing chain. A difficult challenge for today's mobile terminal developers is to keep the computational efforts to a minimum by designing efficient hardware and software. Designing such hardware and software leads to lower power consumption and longer standby time for the mobile equipment. The introduction of multimedia to modern mobile terminals encourages research for more efficient software implementations of image processing applications adapted to the mobile terminals hardware.

As in the world of digital cameras, the mobile terminal has adopted the CMOS image sensor. In order to reduce image processing complexity and cost, these sensors feature the Bayer RGB property and utilize a Color Filter Array (CFA) by letting each image-point be represented by only one of the additive colors red, green and blue. This Bayer RGB representation of an image is subsequently demosaiced in order to obtain full resolution RGB as input to the conventional JPEG compression block.

By changing or omitting improper image processing blocks, existing image compression standards can be adapted to fit mobile terminals.
With this in mind, a simplified image processing chain is suggested by Tryggve Mikkelsen [Mikkelsen, 2001] as shown in *Figure 1*. The idea is to leave out the image processing blocks demosaicing, color space conversion (RGB to YUV) and YUV-subsampling on the sender side and YUV-interpolation and color space conversion (YUV to RGB) on the receiving side. This way Bayer RGB will appear as the direct input to the compression scheme. The grey shaded area represents the classical image processing chain.

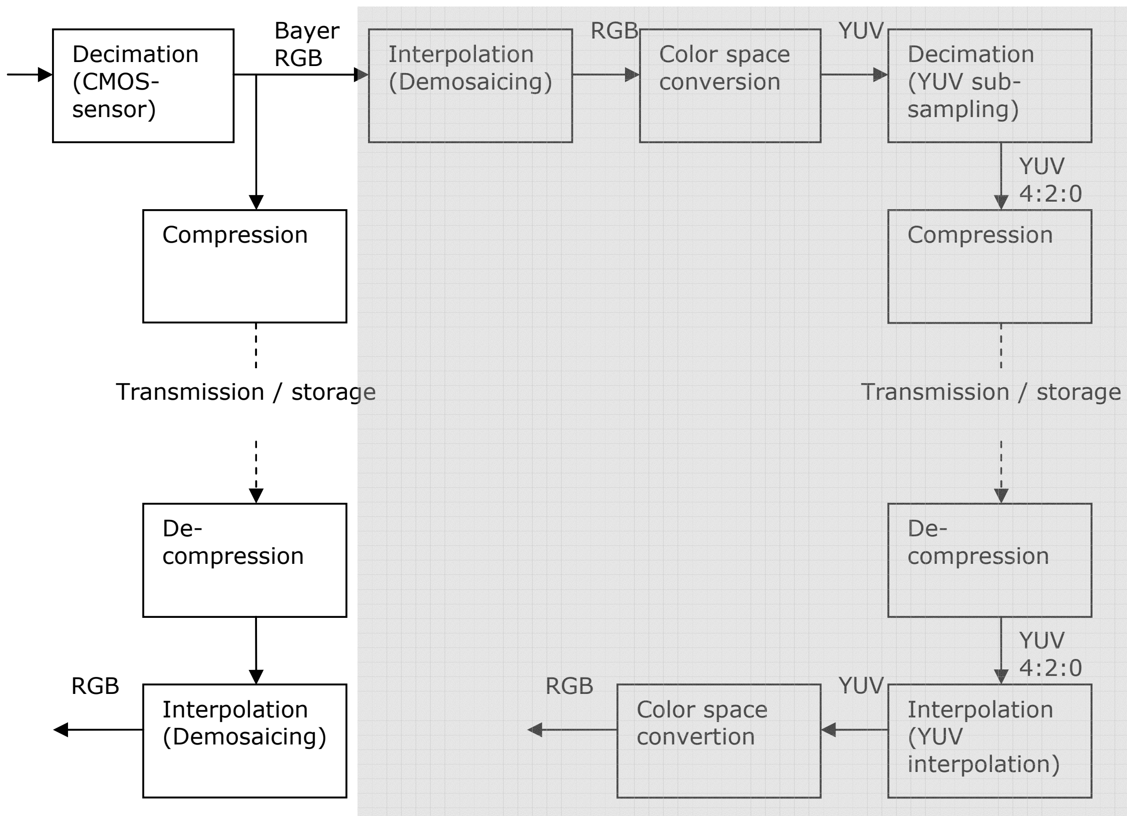

Figure 1 – The improved image processing chain

This thesis investigates the effects of using Bayer RGB directly as input to the compression scheme. It focuses on two different approaches of Bayer RGB as input; one processing the Bayer RGB as it appears from the sensor and the other distinguish the colors red, green and blue in three different arrays, then processing them.

The obvious reduction on computational effort is supposed possible due to the nature of the CMOS-sensor providing Bayer pattern RGB information. However, the effects on computational effort, compression ratio, visual reconstruction quality and other possibly suitable measures, need to be investigated in order for this seemingly proprietary compression scheme to be justified.

The overall goal of this thesis is to improve the camera functionality of mobile terminals by introducing a simplified image processing architecture that will result in additional computational savings compared to the classical way of processing images in mobile terminals without this being at the expense of visual reconstructed image quality.

1.2 Final thesis proposal

In understanding with the supervisor, the final thesis definition is defined to consist of the following elements:

- A study of fundamental digital image processing. The study should cover the signal chain from pre-processing through compression and possibly post-processing.
- A thorough discussion of the main tasks found in the compression scheme. This could include case studies of standard compression formats as JPEG and JPEG2000.
- A discussion of the effects of using the Bayer RGB as input to the proposed compression scheme.

This final thesis proposal is intended to result in a report that presents the pre-processing chain in general terms. It is important to notice that the main focus will be on the functionality of the compression scheme and the effects on using Bayer RGB as input to the proposed compression algorithm.

2 Fundamental digital image processing

2.1 Introduction

In natural imagery and graphics all representations are continuous. However, compared to computer graphics and imagery, the representations of images differ from aspects of conventional graphics because computer representations of images are digital and discrete.

The motivation for this chapter is the need for a basic understanding of the main operations found in the digital image processing chain. See *Figure 2*. Due to the limited scope of this thesis all of the operations are not considered to be of equal importance. This chapter is therefore meant to serve as a concise summary of preprocessing operations that are relevant for the work presented in this thesis.

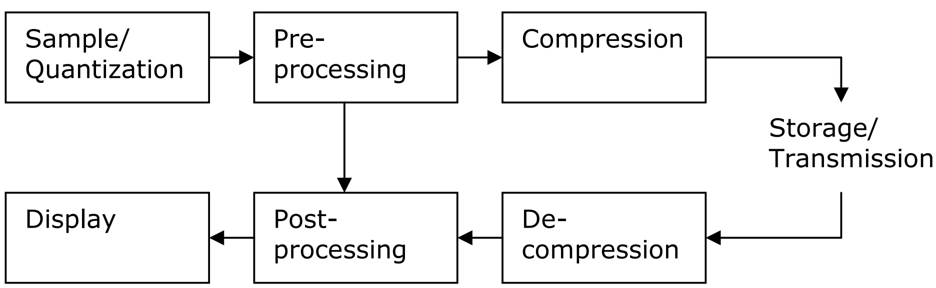

Figure 2 – General image processing chain

Image pre-processing includes operations that compensate for errors introduced during image capturing of the CMOS sensor and prepares image data for compression. To represent an image digitally, a color space need to be specified. *Chapter 2.2* describes the two color spaces relevant for this thesis; the RGB color space and the YUV color space. *Chapter 2.3* contains information about some common image sensors suitable to mobile terminals while *chapter 2.4* elaborates on the operations found in the pre-processing block.

Compression is of great concern in the work presented in this report. Many compression algorithms of interest exist today, but the need of a conventional and widely used standard resulted in the choice of JPEG compression. It consists of extensive functionality and in order to give a fair description of the operations, the JPEG algorithms is explored in detail in *chapter 3*.

Post-processing is not considered to be of relevance to the work of this thesis as it focuses on tasks mainly found in conjunction with the presentation of reconstructed images. Post-processing is therefore omitted.

2.2 Color space models

"Color is the perceptual result of light having wavelength from 400 to 700 nm that is incident upon the retina."

- Charles Poynton

The human retina has three types of color sensitive cones, which responds differently to various wavelengths. Because there are three different color receptors in the human eye, three components are necessary and sufficient to describe a color. The Commission Internationale de L'Éclairage (CIE) has defined a system to compute a triple of numerical components that can be considered to be the mathematical coordinates of color space. CIE specifies three monochromatic primaries at the wavelengths 700nm (red), 546,1nm (green) and 435,8nm (blue). Any system for color specification must be intimately related to the CIE specifications. The systems useful for color specification are all based on the CIE XYZ color space [Poynton, 1997].

2.2.1 The RGB color space

There are actually two different RGB color spaces; the RGB color space which has a linear relation to the CIE XYZ color space, and the R'G'B' color space which has a nonlinear relation to CIE XYZ. Gamma correction is an operation that leads to a nonlinear relation between CIE XYZ and RGB, and also between RGB and R'G'B.
However, the difference between the linear RGB and the nonlinear (gammacorrected) R'G'B' is almost insignificant. We have not distinguished between RGB and R'G'B' color space in this thesis. When referring to RGB in this report, R'G'B' is in almost any case the correct term as the RGB color space only exist early in the preprocessing chain, before the gamma correction block.

2.2.2 The YUV color space

The YUV color space has a linear relation to the RGB color space, and its transform looks like this:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Equation 1 – RGB to YUV conversion

By converting an image from RGB to YUV color space it becomes possible to separate the image's luminance and chrominance information. The luminance information is represented by Y while chrominance are represented by U and V.
From the transformation matrix we see that Y = 0.299R + 0.587G + 0.114B. It's obvious that green is the biggest contributor to luminance. This is as a result of the human eyes higher sensitivity to green than other colors. We also see that for U blue is the biggest contributor, while red contributes most to the V component.

Saying that Y is luminance is however not always true; when moving from the nonlinear R'G'B' color space to YUV, Y will not represent luminance. As a consequence of the problem described in *chapter 2.2.1*, Y is not the same as CIE luminance because gamma correction is introduced.
Luminance is therefore not the correct term to use.
According to Poynton [Poynton, 1997], programmers and engineers who are reimplementing video technology in the computer domain are liable to draw the wrong conclusions from careless use of terms. Users suffer from this because the exchange of images is compromised. Poynton urges to avoid using terms as YUV and luminance, and rather use the correct terms $Y'C_BC_R$ and luma. We have however not used the term luma in this report.

2.3 Image sensors

2.3.1 The CCD image sensor

The CCD (charge-coupled device) image sensor was introduced in 1974, and is the most common sensor used for TV-cameras and digital cameras as it provides high quality low-noise images. When reading from such a sensor, the pixel values (charges) are transported across the sensor, line by line, and then shifted into an analog-to-digital converter turning each pixel's value into a digital value.

Bayer pattern color filter array

For this type of image sensors, only one color can be acquired per pixel. By adding a color filter array (CFA) to the sensor, the CFA let only one color pass through from the lens to the sensor. The most used CFA is the Bayer pattern as shown in Figure *3*.
Red and blue are subsampled at a rate of ½ in both horizontal and vertical direction, while green is subsampled at a rate of ½ only in horizontal direction. Because of the eyes higher sensitivity to green, 50 % of the Bayer pattern CFA consists of green information. For limiting scope, the Bayer pattern CFA is the only CFA discussed in this thesis.

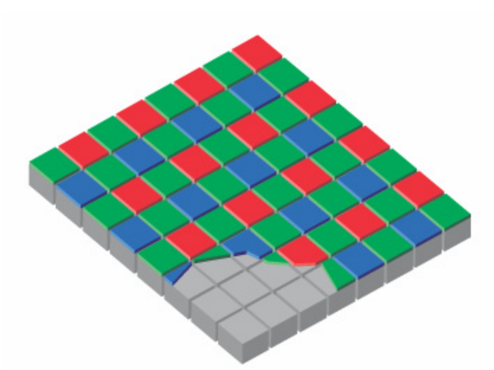

Figure 3 – The Bayer pattern CFA

2.3.2 The CMOS image sensor

The CMOS (Complementary Metal Oxide Semiconductor) image sensor is an inexpensive sensor providing medium-quality images more susceptible to noise than a CCD captured image. For a CMOS sensor, each pixel can be read individually, making it more flexible than a CCD sensor.
The sensor is built by use of the same processes and the same equipment as modern computer processors such as Intel Pentium 4 are. This makes the sensor cheap to produce, and sensors with an increasing number of pixels per square can be made as new production technology for processors are developed.
The Bayer pattern CFA shown in Figure *3* is the most used color filter array for CMOS sensors.

2.3.3 The fullcolor image sensor

On February 11, 2002 Foveon inc. released their line of Foveon X3 image sensors. The sensors are the world's first full color image sensors that capture both red, green and blue color information at every point of the image. This technology results in sharper images and better color reproduction.

A resource demanding operation that is common for cameras with CCD or CMOS sensors, the demosaicing operation, becomes obsolete as the captured image is a fullcolor RGB and need no interpolating. Therefore imaging errors introduced by the demosaicing operation are eliminated.

The Foveon image sensor takes advantage of the fact that light with different wavelength is absorbed at different depths in silicon. The red, green and blue photodetectors are located at different depths within the sensor as in Figure *4*. The blue ones are located near the surface of the sensor, the green ones in the middle, and the red ones in the bottom of the sensor.

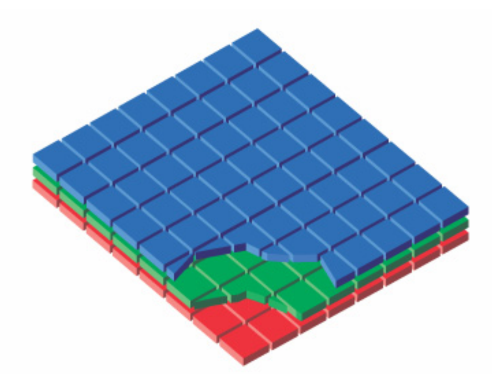

Figure 4 – The fullcolor image sensor

2.3.4 Range of use for CCD and CMOS sensors

There are many differences between the CCD and the CMOS image sensor, and their range of use are distinct.

The CCD sensor is made by special manufacturing processes, while the CMOS sensor is made, as earlier mentioned, by common manufacturing processes. This makes the CCD sensor much more expensive to produce than a CMOS sensor.
Each pixel on a CMOS sensor has several transistors located next to it, making it possible for photons to hit the transistors instead of the photodiode. This makes the CMOS sensor less sensible to light than the CCD sensor.
The CMOS technology, which is known for its low power consumption, makes the CMOS sensor consume as little power as 1/100 of what an equivalent CCD does. For mobile terminals, where low power consumption is very important, the CMOS sensor becomes an obvious choice. When, in addition, the production costs for the CMOS sensor are low compared to the CCD sensor, the CMOS sensor undoubtedly becomes the best option for the mobile terminal.
Based on these properties, we can see that the CCD sensor is best suited for cameras that focus on high quality pictures and where relatively high power consumption is not a big problem. CMOS sensors on the other hand are well suited for hardware where low power consumption is demanded, and where image quality is not so important.

2.4 Pre-processing

2.4.1 General

Image pre-processing includes operations that compensate for errors introduced during image capturing of the CMOS sensor and prepares image data for compression. The figure below outlines the image processing operations found in the pre-processing chain.

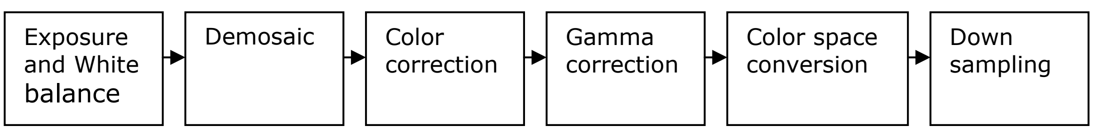

Figure 5 - Image pre-processing operations

As a part of the study of the fundamental image processing chain, the following chapters describe each of the blocks presented in the figure.

2.4.2 Exposure and white balance

In order for a camera device to know the amount of light to be exposed to the light-sensitive photodiode, it is equipped with an *exposure* block. In photographic terms, this is the product of the intensity of light and the time the light is allowed to act on the sensor, or the film. Exposure is typically handled by electronic devices in the camera that simply records the amount of light falling onto the light-sensitive sensor for the shutter-speed period of time. In a true photographic sense, this amount of light is then used to calculate the correct combination of aperture and shutter speed at a given sensitivity.

The purpose of *white balance* is to give the camera a reference to white. White balance is a name given to a system of color correction to deal with differing lighting conditions and is done to aid in overcoming color problems created by adverse lighting conditions. By exposing the light-sensitive sensor to a white motif the camera will record the color temperature and use this to render correct images. The white balance equalizes average pixel luminance among color bands (RGB). By not paying attention to this functionality a real world white motif will not appear white when reproduced.

2.4.3 Demosaicing

The task of the demosaicing operation is to transfer an image from Bayer RGB to RGB by interpolating the three subsampled colors red, green and blue as shown in Figure *6*.

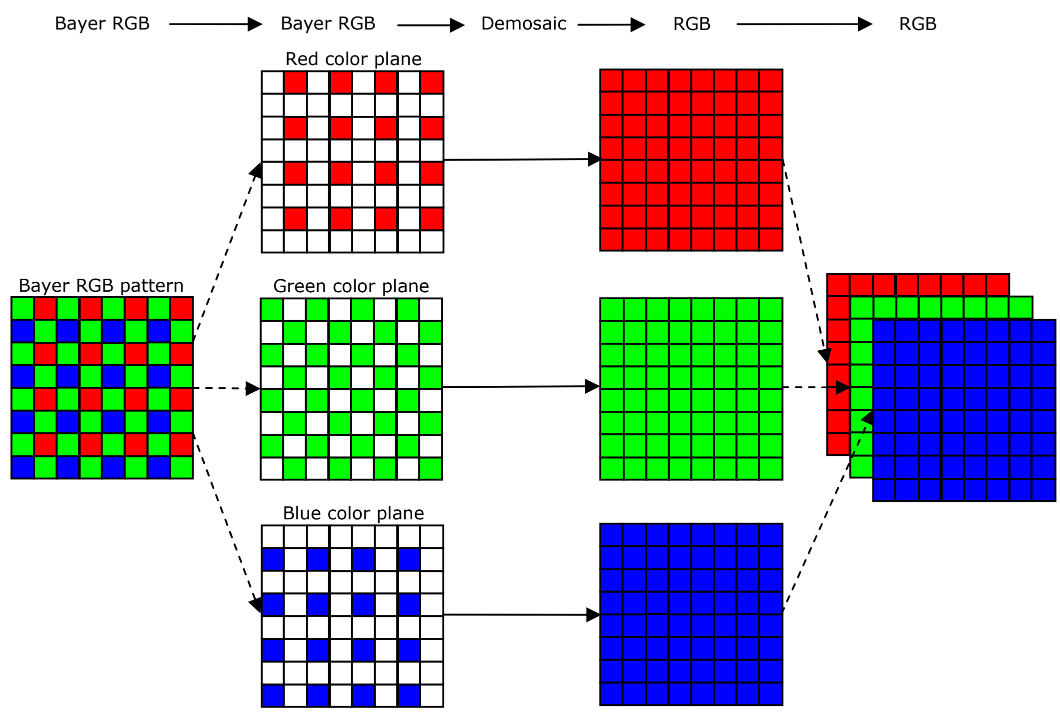

Figure 6 – The demosaicing operation

Referring to Sakamoto [Sakamoto, 1998], interpolation in mathematical terms can be expressed as:

$$c(x,y) = \sum_k \sum_l c(x_k, y_l) h(|x - x_k|) h(|y - y_l|)$$

Equation 2 – Interpolation

where c(x,y) is the interpolated value, h is the interpolation kernel and $c(x_k,y_l)$ is the sampled value at pixel (x_k,y_l). There are several interpolation methods, but in this thesis we only explain three of them: The nearest neighbor, bilinear interpolation and cubic convolution method.

Nearest neighbor interpolation

For nearest neighbor interpolation, each interpolated pixel is assigned the value of the nearest pixel of the input image. If there is more than one pixel that has the same distance to the pixel to be interpolated, one of these is chosen.
The interpolation kernel for such an interpolation is [Sakamoto, 1998]:

$$h(x) = \begin{cases} 1 & 0 \le x \le 0.5 \\ 0 & 0.5 < x \end{cases}$$

Equation 3 – Interpolation kernel for the nearest neighbor interpolation

Nearest neighbor interpolation is a simple interpolation method which requires only small amounts of computational resources. The drawback is the poor quality of the interpolated image.

Bilinerar interpolation

Bilinear interpolation makes use of the four adjacent pixels to calculate the interpolated pixel value. The interpolation kernel is as follows [Sakamoto, 1998]:

$$h(x) = \begin{cases} 1 - x & 0 \le x \le 1 \\ 0 & 1 < x \end{cases}$$

Equation 4 – Interpolation kernel for bilinear interpolation

Bilinear interpolation is a relatively simple interpolation method, and the interpolated image's quality is better than for nearest neighbor intepolation. The interpolation method providing the best image quality, however, is the cubic interpolation.

Cubic interpolation

The cubic convolution use adjacent pixels to determine the value of the interpolated pixel. The number of adjacent pixels used to interpolate is not fixed, but they are weighted to be an approximation of the theoretically optimal sinc function. Its interpolation kernel is described as follows [Pratt, 1991]:

$$h(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \le |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \le |x| < 2 \\ 0 & 2 \le |x| \end{cases}$$

Equation 5 – Interpolation kernel for cubic interpolation

where a is a coefficient that can be used as a tuning parameter to obtain a best visual interpolation. Keys [Keys, 1981] has proposed setting a = -1/2 to obtain an interpolation function that approximates the original unsampled image to as high a degree as possible in the sense of a power series expansion.

From the sections above we see that nearest neighbor is the simplest algorithm, and is also the one which performs the cheapest interpolation when considering quality of the interpolated image. The algorithm use only one adjacent pixel to determine the interpolated value while bilinear interpolation uses 4 adjacent pixels.

The interpolation method that provides highest quality of the interpolated image is the cubic convolution, which also is the most resource demanding interpolation method, using adjacent pixels to estimate the interpolated pixels value.
For image processing, the bilinear interpolation is the most used interpolation method because of its compromise between reconstructed quality and computational effort. Nearest neighbor is a computationally efficient algorithm, but its poor quality of the reproduced images makes it an algorithm not suited for image processing.
The cubic convolution interpolations reproduced images is of high quality, but, as described, it also requires more computational power.

2.4.4 Color correction

In order for the desired colors or real life colors of a motif to be correctly reconstructed, color correction sometimes needs to be applied. Color correction is a pre-processing operation that compensates for the differing color temperatures at different scenes.

Differing color temperatures will affect the way an image capturing device, for instance a CMOS sensor, "sees" the motif and hence affect the color composition of the reconstructed image.

Color correction is a photo technique that is concerned with image acquisition, mainly found in capturing devices like cameras. Because of this, the exact procedure on how color correction is done is not considered to be an issue in this thesis and will hence not be discussed in further detail.

2.4.5 Gamma correction

Real-world image aquiring devices have non-linear characteristics when it comes to the process of capturing the light of a real scene and transforming it into electronic signals. It would be convenient for graphics programmers if all the components of an imaging system were linear. However, many displays, almost all photographic films, and many electronic cameras have nonlinear signal-to-light-intensity or intensity-to-signal characteristics.
Fortunately, all of these nonlinear devices have a transfer function that is approximated fairly well by a single type of mathematical function: a power function. This power function has the general equation

$$output = input^{\gamma}$$

Equation 6 – Non-linear transfer function

Furthermore, the numerical value of the exponent of this power function is trivially known as gamma and denoted by the greek letter gamma. The compensastion for this non-linear relation will be illustrated as *gamma correction* below.

In most still image and video capturing systems, gamma correction is done in the camera or the image acquiring part of the signal processing chain. According to Charles Poynton's [Poynton, 1999] document on gamma this choice is made because it is more cost effective to place the expensive processing in the small number of capture devices (cameras) than in the large number of receivers. Hence, this is the reason why gamma correction is not considered as a part of post-processing in the signal processing chain.

The following figure can be found in the referred document by Poynton. It illustrates the gamma of a CRT display for three different settings of the picture control.

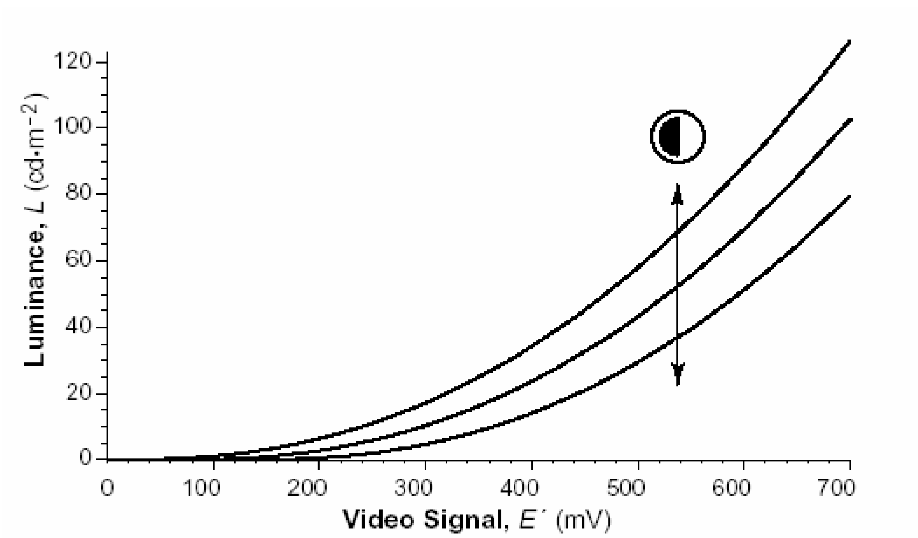

Figure 7 – An actual CRT transfer function

To compensate for the display's transferfunction, gamma correction is applied. In an image acquisition system, luminance of each of the linear-light of the red, green and blue (tristimulus) components is transformed to a non-linear signal by a gamma correction function. This is, as previously mentioned, universally done at the camera. Several transfer functions that take linear light tristimulus values to a non-linear component can be found. The function illustrated in this thesis is called the Rec.709 transfer function and is shown in Figure *8*.

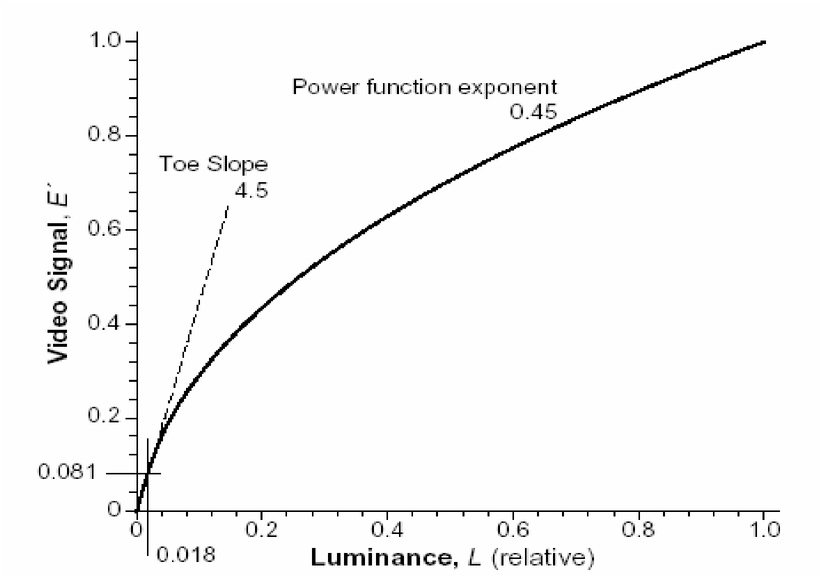

Figure 8 – Rec. 709 gamma correction transfer function

Now that the gamma has been compensated for, an idealized monitor or display will invert the transform. This means that the ratio between the intensities of any two areas in the reproduced image will be the same as it was in the original scene.

2.4.6 Color space conversion

This block converts the color space from RGB to $Y'C_BC_R$, something which makes it possible to separate luminance and chrominance information. This conversion is carried out by simply multiplying the RGB values with the transformation matrix in Equation *1*. According to *chapter 2.2.2*, the result of the conversion will not be YUV but $Y'C_BC_R$. This means that the luminance and chrominance values are only approximately correct. The reason why the colors are divided into chrominance and luminance information is the possibility to compress them differently. Because of the eyes lower sensitivity to chrominance information, chrominance can be compressed more than luminance information without visible losses.

It's important to notice, according to [Poynton, 1998], that the codeword utilization in $Y'C_BC_R$ is very poor. Conventional 8 bit per color RGB coding allows 2^{24} different colors, while for $Y'C_BC_R$ only 1/4th of the codewords represent colors. This introduces more quantization noise, and poorer signal-to-noise ratio than for RGB.

2.4.7 Downsampling

Downsampling is done to reduce the data amount of an image by reducing the color details. Because of the eyes low sensitivity to color details, they can be discarded without visible losses. There are several ways to carry out the downsampling, for JPEG $Y'C_BC_R$ 4:2:0 is used. For $Y'C_BC_R$ 4:2:0 both chrominance components are downsampled by a ratio of 2 in both horizontal and vertical direction as shown in Figure *9*. Nothing is done to the luminance component.

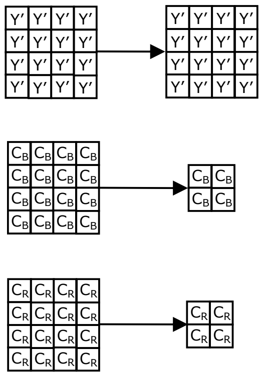

Figure 9 – $Y'C_BC_R$ 4:2:0 downsampling

If subsampling is done by simply dropping or averaging alternate C_B and C_R samples, then filtering artifacts (such as aliasing) will be introduced [Poynton, 1998].
Those problems will however be discarded if the image is lowpass filtered first.

When an image is color space converted, $3/4^{th}$ of the $Y'C_BC_R$ code combinations do not represent colors. The downsampling function then reduces the chrominance information to $1/4^{th}$.
This has become a major disadvantage for CGI (computer-generated imagery). By not converting the image to $Y'C_BC_R$, this problem is avoided. By using either Bayer RGB or classical RGB as input to the compression scheme, the quality loss due to color space conversion and downsampling will be reduced considerably.

3 JPEG compression

3.1 Introduction

The pre-processed image is now color space converted and downsampled to $Y'C_BC_R$ 4:2:0, and is ready to be compressed. JPEG is a general-purpose compression standard that aims to support a wide range of applications for continuous-tone images of varying resolution and size. It is by far one of the most widely recognized standards in existence today. Therefore, and because JPEG provides good compression, JPEG becomes the natural choice of compression algorithm in this thesis.

This chapter describes the various blocks of the JPEG compression scheme. The motivation for this chapter lies within the fact that it is important to know how JPEG behaves. An investigating of JPEGs behaviour enables us to understand the effects of using another color space than YC_BC_R as input to the compression scheme.

3.2 Goals of JPEG

The goals of JPEG have been to develop a method for continuous-tone image compression which meets the following requirements [Wallace, 1992]:

1. Achieve rate and reconstructed image quality “at or near the state of the art” with image fidelity classifiable as “very good” to “excellent”. In addition the encoder should be parameterizable, so that the application can set the desired compression/quality tradeoff.

2. Be useful for compression almost any continuous-tone still-image, including both gray-scale and color, any color space, and most image sizes. Furthermore, it should not have restrictions on image complexity such as range of colors and statistical properties.

3. Have complexity that would allow software implementations on many common platforms and affordable hardware implementations.

4. JPEG should have the following modes of operation:

-Sequential encoding: meaning that each image component is ordered according to a rasterlike scan, left-to-right and top-to-bottom.

-Progressive encoding: using this mode, the image is encoded in multiple scans. This is an advantage for applications in which transmission time is long resulting in the image to build up in multiple coarse-to-clear passes.

-Lossless encoding: this mode guarantees for the exact reconstruction of every source image sample value.

-Hierarchical encoding: this involves image encoding at multiple resolutions so that lower-resolution versions of the image may be accessed without first having to decompress the image at its full resolution.

3.3 Description of JPEG

Image compression and decompression operations are used to reduce the data size of an image. Compression algorithms seek to extract essential information from an image so that the image can be accurately reconstructed. Nonessential information is discarded. If the amount of data necessary to represent an image is reduced, then the amount of storage space and time to transport the image over a link are reduced.

The JPEG image compression scheme was assigned to include two basic compression methods. The first is a DCT(Discrete Cosine Transform)-based method for lossy compression and the second is a predictive method for lossless compression. Lossy image compression preserves some specified level of image quality, while lossless compression means that the exact data content of the original image is preserved. This was done in order for the compression standard to meet the differing needs of many applications.

The JPEG image compression scheme is commonly used in lossy mode. It has several lossy encoding modes, starting with the JPEG feature of a simple lossy technique known as the *Baseline sequential method*, a subset of the other DCT-based modes of operation. The Baseline method is sufficient for a large number of applications and is by far the most implemented JPEG method to date.

It is in the main interest of this thesis to implement a simplified lossy image processing chain and investigate the effects on image reconstruction quality and compression ratio associated with the introduction of the simplified processing algorithm. With this in mind, the Baseline method became a natural choice of JPEG implementation in our test environment. It has all the image processing features necessary for our range of use and is, as previously mentioned a lossy coding technique. The processing steps of the DCT-based Baseline codec are explained in detail in the following.

In Figure *10*, the key processing steps which are at the heart of the DCT-based modes of operation are shown. The figure illustrates a single component image compression (grayscale compression) and can be regarded as compression of a stream of 8 x 8 blocks of image samples. Color image compression can essentially be regarded as compression of multiple single-component compressions. These components are either compressed entirely one at a time, or are compressed by alternately interleaving 8 x 8 sample blocks from each in turn. The implementation of color image compression in this thesis conducts one color entirely at a time.

As illustrated in Figure *10*, the image is partitioned into 8 x 8 blocks of pixels prior to the input of the encoder. These blocks are ordered according to a “rasterlike” left-to-right and top-to-bottom scan. The initial image data is usually converted from normal RGB color space to $Y'C_BC_R$. It is not necessary, however, to change the image’s color space, but the $Y'C_BC_R$ represented color space scheme of the source image makes it easier to identify chrominance and thus eliminate unnecessary image data that cannot be distinguished by the human eye.

Pixel-blocks of the $Y'C_BC_R$ image are then combined into a Minimum Coded Unit (MCU). The image is processed one MCU at a time. Within the MCU, the 8x8 blocks are processed separately, one at a time. However, before the DCT transformation of the $Y'C_BC_R$’s luminance component (Y’) and chrominance components (C_B and C_R) respectively, the dynamic range of the pixel intensity value is converted from 0-255 to
-128 to 127. This is done to uniformly distribute the block and eliminate the DC bias of the signal.

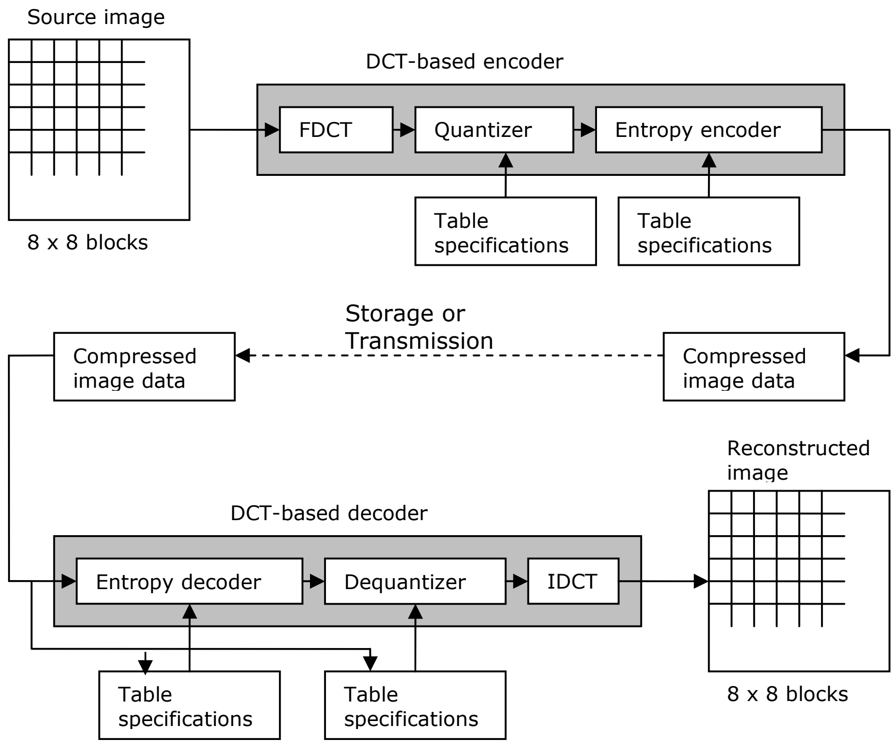

Figure 10 – DCT-based JPEG codec

Further out the compression scheme JPEG uses DCT (Discrete Cosine Transform), quantizing, zigzag ordering, runlength encoding and entropy encoding in order to compress an image. Next, each of these terms will be described.

3.4 The discrete cosine transform (DCT)

The discrete cosine transform (DCT) is related to the Fourier transform and is a technique for converting a signal into elementary frequency components. It is widely used in image compression, a DCT application field pioneered by Chen and Pratt [Chen and Pratt, 1984]. The DCT separates an image into its spatial frequency components in the x and y directions (two-dimensional DCT). For example, a solid color image will have a DCT with low frequencies while a line drawing image will have a DCT with very high frequencies.

The transform of an image has the same number of coefficients as the number of pixel blocks in an image, i.e. an 8 x 8 image will have an 8 x 8 DCT. Pixel blocks are transformed into a linear combination of basis functions, providing the DCT coefficients. From this frequency domain representation of an image, the contribution of each two-dimensional basis function in the transformed image can easily be designated.

DCT image compression causes image degradations that increase proportionally with increased compression ratios. High frequency details can be lost and blocking artifacts can occur in extreme cases. Blocking artifacts are rectangles (blocks) appearing in compressed images. However, DCT compression techniques typically provide gray scale image compression ratios of 10:1 without serious image quality degradations.

The DCT transformation of signals like images requires the use of a two-dimentional DCT transform since images are arranged according to a "rasterlike" left-to-right and top-to-bottom scan. Because of this, before turning the focus on what happens with the $Y'C_BC_R$ input of the JPEG Baseline encoder, a short description of the one- and two-dimensional properties of the DCT transform will be given.

3.4.1 One-dimensional discrete cosine transform

The discrete cosine transform of a list of n real numbers s(x), x=0,…,n-1, is the list of length n given by:

$$S(u) = \sqrt{\frac{2}{n}} \cdot C(u) \cdot \sum_{x=0}^{n-1} s(x) \cdot \cos\frac{(2x+1)u\pi}{2n} \qquad u = 0,\ldots,n$$

Equation 7 – DCT

where C(u) = $2^{-1/2}$ for u=0
= 1 otherwise.

The transformed list S(u) represents the coefficients of the DCT transformed signal. Each element of the transformed list S(u) is the inner product of the input list s(x) and a basis vector. Each basis vector corresponds to a sinusoid of a certain frequency. From this the coefficients may be regarded as the relative amount of each frequency present in the input signal s. The constant factors are chosen so that the basis vectors are orthogonal and normalized.

The DCT transform is reversible, that is, the list s(x) can be recovered from its transform S(u). This is done by applying the inverse cosine transform (IDCT) on S(u) described below.

$$s(x) = \sqrt{\frac{2}{n}} \cdot \sum_{u=0}^{n-1} C(u) \cdot S(u) \cdot \cos\frac{(2x+1)u\pi}{2n} \qquad x = 0,\ldots,n$$

Equation 8 – IDCT

where C(u) = $2^{-1/2}$ for u=0
= 1 otherwise.

This equation expresses s as a linear combination of the basis vectors. The result of this equation represents the input list s(x).

3.4.2 Two-dimensional discrete cosine transform

The transform defined thus far is called one-dimensional since the input data is represented as a function of one variable. One-dimensional transforms have application to speech as well as to other physical signals. However, two-dimensional versions of discrete transforms can also be defined. These transforms have particular importance for naturally occurring two-dimensional entities such as images. The 2D DCT is just a one-dimensional DCT applied twice, once in the x direction and once in the y direction. Thus, the two-dimensional DCT transform is given by:

$$S(u,v) = \frac{2}{\sqrt{n \cdot m}} \cdot C(u) \cdot C(v) \cdot \sum_{y=0}^{m-1} \sum_{x=0}^{n-1} s(x,y) \cdot \cos\frac{(2x+1)u\pi}{2n} \cdot \cos\frac{(2y+1)v\pi}{2m}$$

for u = 0,…,n
v = 0,…,m

Equation 9 – 2D DCT

where C(u) $= 2^{-1/2}$ for u=0
= 1 otherwise

As in the one-dimensional case, each element S(u,v) of the transform is the inner product of the input list s(x,y) and a basis function, but in this case, the basis functions are n x m matrices. Each two-dimensional basis matrix is the outer product of two of the one-dimensional basis vectors. Naturally, as with the one dimensional case of the transform, the 2D DCT is reversible. By applying the inverse 2D DCT on the coefficients found as elements in the list S(u,v), the original input list s(x,y) is obtained.

Now its time to turn the attention back to the $Y'C_BC_R$ input of the encoder. Each $Y'C_BC_R$ 8 x 8 block of source image samples is effectively a 64-point discrete signal which is a function of the two spatial dimensions x and y. The DCT takes such a signal as its input and decomposes it into 64 orthogonal basis signals. Each contains one of the 64 unique two-dimensional "spatial frequencies" which comprise the input signal's spectrum. The output of the DCT is the set of 64 basis signal amplitudes or "DCT coefficients" whose values are uniquely determined by the particular 64-point input signal.

The coefficient with zero frequency in both directions is called the DC-coefficient and the remaining 63 coefficients are called the AC-coefficients.

3.5 Quantization

Quantization is an operation performed on the floating point DCT-coefficients by dividing them by an integer. The result is rounded to the nearest integer. This division is done in conjunction with a 64-element quantization table, which must be specified by the application or user as an input to the encoder. When constructing such a table, there will always be a trade off between compression and image quality. Each element of the quantization table can be any integer from 1 to 255, which specify the step-size or resolution of the quantizer for its corresponding AC coefficient. A small number in the quantization table means smaller step-size of the encoder and better resolution and leads to an increased precision in reconstructing the desired image quality. On the contrary, a large number leads to a larger step-size of the encoder, poorer resolution and may result in the corresponding DCT coefficient to have no influence on retrieving the desired image quality and may thus be neglected. In other words, quantization is the process of reducing the number of possible values of a quantity, thereby reducing the number of bits required to represent it. Stated another way, the goal of this processing step is to discard information which is not visually significant.

The decompressor of the JPEG codec gets its quantization table information from the header of the processed image file where the table information is stored.

3.6 Zigzag ordering and DC coding

After quantization, the DC coefficient is treated separately from the 63 AC coefficients. The DC coefficient represents the average value of the 64 source image samples. Since there usually is a strong correlation between the DC coefficients of adjacent 8 x 8 blocks, the quantized DC coefficient is encoded as the difference between the present and the preceding DC component. In JPEG, this has proved to be an effective way of coding the DC coefficients. This special treatment of the DC coefficients is worthwhile, as they frequently contain a significant fraction of the total image energy.

Finally the DC coefficients and all of the AC coefficients are ordered into a zig-zag sequence illustrated in the figure below. This ordering helps to facilitate entropy coding by placing low-frequency coefficients, which are most likely to be non-zero, before high-frequency coefficients. After quantization, it is not unusual that more than half of the DCT coefficients are equal to zero. Zigzag ordering, or rearrangement of data, is done on the data blocks to consolidate and increase the run length of zeros. JPEG takes advantage of this by runlength coding of the zig-zag ordered DCT blocks, described in the following.

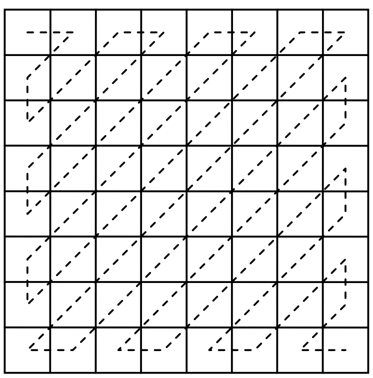

Figure 11 – Zigzagging of the quantized DCT block

3.7 Runlength encoding

Runlength encoding is done on the zigzag ordered data to exploit the property that the result from the zigzagging probably consists of several series of subsequent zeros. The zigzag ordered coefficients are mapped into an intermediate sequence of "symbol-1" and "symbol-2" pairs. Symbol-1 consists of (RUNLENGTH, SIZE), where RUNLENGTH is the length of zero values preceding the next nonzero AC-coefficient and SIZE is the number of bits needed to code the next nonzero coefficients amplitude, which is symbol-2. Symbol-2 (AMPLITUDE) is as stated the value of the AC coefficient. RUNLENGTH takes the values 0 to 15, and a symbol-1 pair of (15,0) represents a runlength of 16 zero-valued AC coefficients. Since there can be long runs of zero-valued coefficients, up to three consecutive (15,0) pairs followed by a symbol-1 that completes the runlength and a single symbol-2 are allowed.

If the last run of zeros includes the last AC coefficient in the block, no symbol-2 is sent after the final run of zeros, but instead, the (0,0) end of block (EOB) symbol is sent that denotes the end of the 8x8 block [Gibson et.al., 1998]

3.8 Entropy coding

After decomposition and quantization of a signal in the JPEG standard, entropy coding is used for final image compression. The two entropy encoding schemes allowed by the JPEG standard are Huffman and arithmetic encoding. Even though codecs with both methods are specified for all modes of operation, the Baseline sequential codec uses Huffman coding only. Although arithmetic encoding often results in a smaller file size, it is not commonly used since the variant of arithmetic encoding that is optionally applied in JPEG is patented by IBM, AT&T and Mitsubishi.

Huffman coding achieves additional compression losslessly by encoding the runlength encoded quantized DCT coefficients more compactly based on their statistical characteristics.

For the Baseline sequential codec the entropy encoding is accomplished in two steps. First, the coefficients are converted into an intermediate sequence of symbols, previously described, and then these symbols are coded using Huffman code. Huffman coding is dependent on existing Huffman table sets specific to the application on hand. The configuration and choice of Huffman tables to apply is considered to be outside the scope of this thesis and will not be investigated in further detail. Rather, the description of entropy coding in this work assumes that the correct choice of tables has been undertaken. The JPEG proposal includes an example set of Huffman tables in its information annex, but because they are application specific, it specifies none for required use. Furthermore, since the DC value of an 8 x 8 block is differently encoded, intermediate symbol sequence generation is slightly different for the AC coefficients than for the differenced DC coefficients. Although Huffman coding is used in both cases, the Huffman coding of AC and DC coefficients use different Huffman tables. In the following, a description of Huffman coding is given. The following description of Huffman coding is not specific to either AC or DC coefficient coding, but outlines the general mathematical principles on which Huffman coding is based.

Once the coefficient data of an 8 x 8 block is represented in the intermediate symbol sequence, each symbol-1, for both AC and DC coefficients, is encoded with distinctive Huffman codes.

Huffman codes are of variable length and in order to make decoding unique it is important that no Huffman-code shall start with a code similar to the code of another symbol. The uniqueness of the codes is to avoid misinterpretation at the decoder where the Huffman encoding operation is reversed. This is taken care of by the applied Huffman table set. According to the Huffman tables used, symbols with a high probability of occurrence are represented with few bits and the other way around. Furthermore, Huffman encoding also requires that the two last code-words shall be of the same length with the only difference lying in the least significant bit. Finally, the entropy encoded information is written to the output file. The Huffman tables used to compress an image are also needed to decompose it. As a result, the Huffman tables are represented in the output stream. On the receiver side, the decompressor reverses the above process.

In order to elaborate further on the generation of Huffman codes an illustration is given on the next page in terms of an example. The following example on how Huffman coding is conducted can be found on the web site by [Aas]. The site focuses on digital image processing techniques like image compression and has an educational perspective concerning multimedia.

The following table represents a list of symbols to be Huffman encoded together with their corresponding probability of occurrence and Huffman code.

Symbol	Probability	Huffman code
E	0,400	0
F	0,200	100
D	0,100	110
A	0,075	1010
G	0,075	1110
C	0,060	1111
B	0,050	10110
H	0,040	10111

Table 1 – Example of Huffman codes

As an example, a received code "10011101010" is unambiguously interpreted into the sent sequence "FGA".

4 The simplified image processing chain

4.1 Introduction

In the preceding chapters, JPEG image compression is described in detail from capturing the image through pre-processing to the compressed file.
A question that arises is: why transform the image from Bayer RGB to YC_BC_R and then back again? Is it really necessary?
By using Bayer RGB directly as input to the compression scheme, the mentioned transformations become obsolete. Mobile terminals often suffer from high power consumption, and by using the output from the image sensor directly as input to the compression scheme several resource demanding operations can be omitted. Hence the power consumption is also lowered.

In this thesis we therefore have investigated two different approaches to the problem of using Bayer RGB as input.

The first solution, presented in *chapter 4.3*, use Bayer RGB as it appears as input to the compression scheme. This means Bayer RGB represented as one array as in *Figure 12*. Here, the image information is treated as it was JPEG luminance information by using JPEGs standard luminance quantization tables.

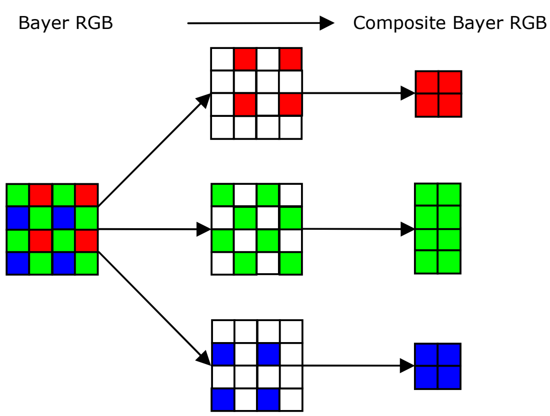

Figure 12 – Bayer RGB and composite Bayer RGB

For the second solution, presented in *chapter 4.4*, Bayer RGB is divided into the three color arrays red, green and blue as shown in *Figure 12*. Green is processed as it was JPEG luminance information, while red and blue is treated as chrominance information.

4.2 General

As earlier explained the output from the image sensor is a Bayer RGB pattern. For classical image processing, the Bayer RGB is interpolated into classical RGB and then transformed to the $Y'C_BC_R$ color space. See *Figure 13*. Next, the $Y'C_BC_R$ chrominance information is downsampled by 2 in both horizontal and vertical direction. After downsampling, the Y'C_BC_R image is dct-transformed from the spatial domain to the frequency domain and the picture is quantized and then entropy encoded.

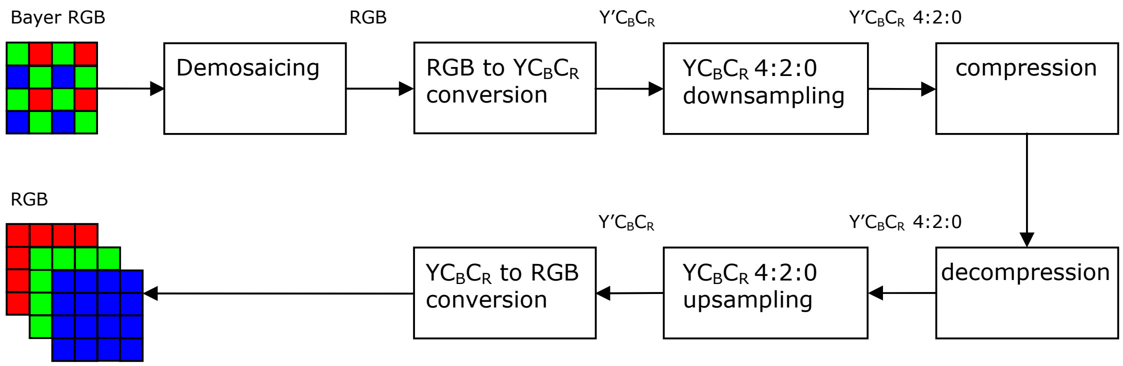

Figure 13 – Block diagram, classical JPEG

When demosaicing the Bayer RGB image, we actually upsample the tree color arrays red, green and blue. Later in the image processing chain, however, the color information is downsampled (Y'C_BC_R 4:2:0 downsampling). This is a waste of computational resources, and in addition color information is lost due to the color space conversion. Is it really necessary to do this resource demanding color space conversion?
Another question that arises is: why demosaic and make the amount of information to be compressed larger when it is possible to move the demosaicing operation to the receiver side and reduce the data amount to be processed?

By using Bayer RGB directly as input to the compression scheme instead of Y'C_BC_R 4:2:0, the demosaicing, color space conversion and downsampling operations can be omitted. It is possible to do this two different ways; either use Bayer RGB as it appears as input as in *Figure 14*, or extract the red, green and blue information into three arrays and use this as input to the compression scheme (*Figure 15*). Note that the demosaicing operation actually is moved from the compression side of the image processing chain to the decompression side, it is not totally left out.

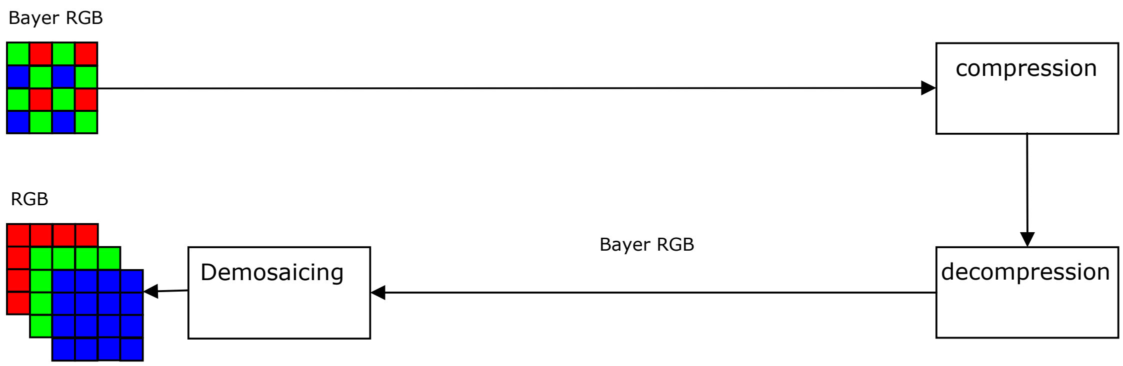

Figure 14 – Block diagram, Bayer RGB as input to the compression scheme

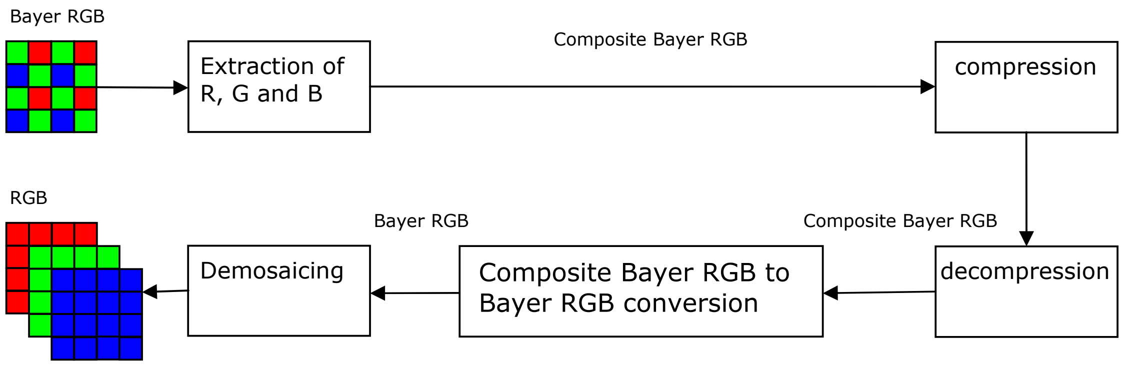

Figure 15 – Block diagram, composite Bayer RGB as input to the compression scheme

A Bayer pattern image contains twice as much green color information as both red and blue color information. However, in comparison with RGB a Bayer RGB image will contain less color information as Bayer RGB is subsampled RGB color information.
It's important to notice that the term color information in this case refers to the amount of storage needed to represent the image, not how much information about the image it contains. From this, because a RGB image is an interpolated version of the Bayer RGB image, we can say that the amount of storage needed to represent the images is not the same, while the visual image information stored is actually the same.
Hence, using Bayer RGB as input to the compression scheme obviously leads to less amounts of data to process and is expected to result in computational savings.

However, the main contribution to computational savings by using Bayer RGB as input to the compression scheme is that resource demanding image processing blocks are left out, not that there is less data to process.

The JPEG compression standard makes use of $Y'C_BC_R$ 4:2:0 as input (Figure *16*). This is not mandatory, however, but is done to achieve better overall image compression by dividing the picture into luminance and chrominance planes which are differently compressed.

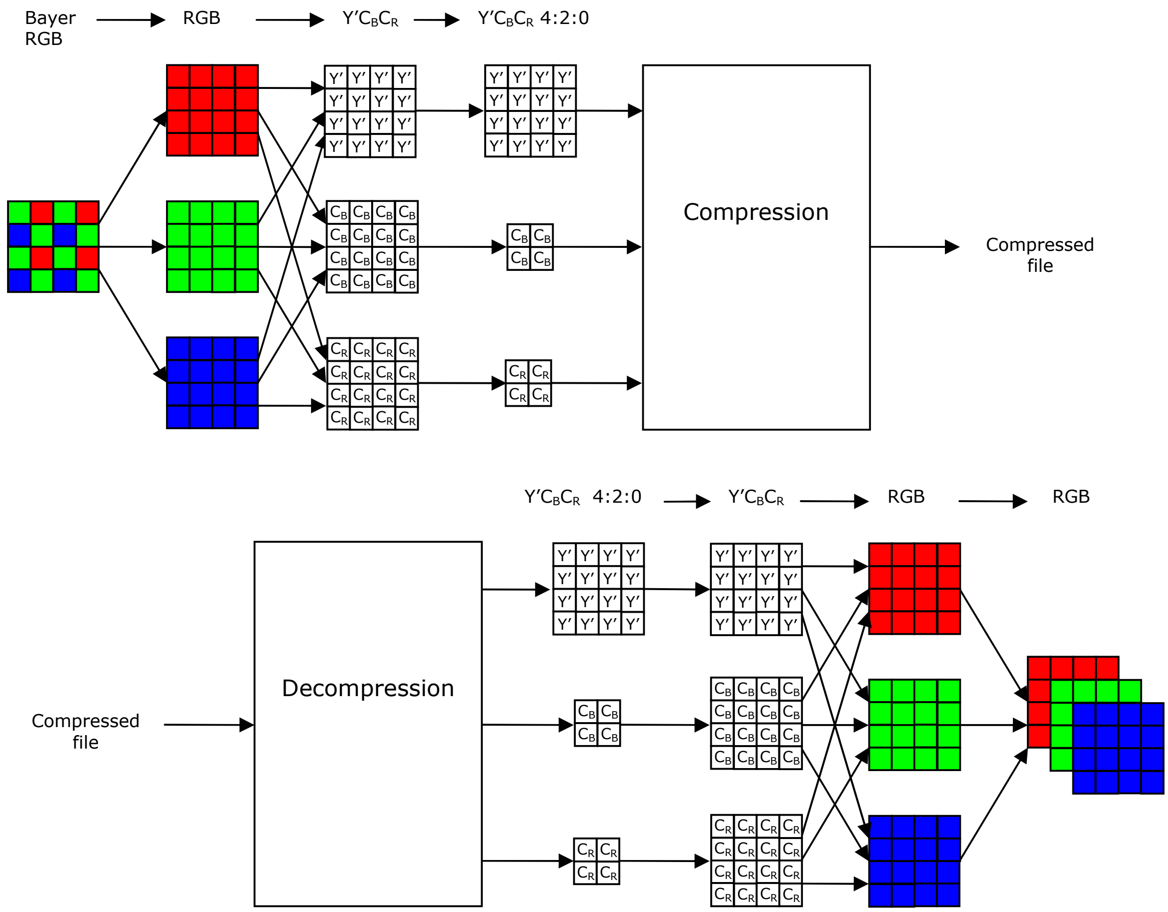

Figure 16 – Classical JPEG compression chain, with Y'CBCR 4:2:0 as input

From the figure we see that the 16 x 3 = 48 RGB pixel values has become 16 + 4 + 4 = 24 $Y'C_BC_R$ 4:2:0 pixel values. This implies that an $Y'C_BC_R$ 4:2:0 image needs only half the storage space that a RGB image needs to be represented. We also see from the figure that RGB uses three times as much storage space as Bayer RGB to represent an image.

Bayer RGB is not expected to cause complications to the compression scheme one of the goals of JPEG is "to be useful for almost any image color space." [Wallace, 1992]

4.3 Bayer RGB as input to the compression scheme

When Bayer RGB as one single array is used as input to the compression scheme, the amount of data to be compressed is reduced to 2/3rd compared to classical JPEG where $Y'C_BC_R$ 4:2:0 is used.
As explained in the chapter above, the data amount required to represent an image as $Y'C_BC_R$ 4:2:0 is half the data amount needed to represent the same image as a conventional RGB image.
A Bayer RGB representation of an image needs 1/3rd of the data amount of what classical RGB does to represent the same image. This means that Bayer RGB only needs 2/3rd of the data amount that $Y'C_BC_R$ 4:2:0 needs to represent an image.

Instead of interpolating each color in the Bayer RGB pattern into new color arrays, we use Bayer RGB as one array as input. This is possible due to the nature of Bayer RGB, which let three colors be represented in one single array, illustrated in Figure *17*.

An obvious disadvantage by putting the Bayer RGB straight into the compression blocks is that JPEG utilizes the possibility of converting images from RGB into $Y'C_BC_R$ format. (See Figure *16*)
This makes it possible to compress the image a lot further because luminance (Y') and chrominance (C_B and C_R) information can be treated differently and therefore be compressed at different ratios by using different quantization matrices.

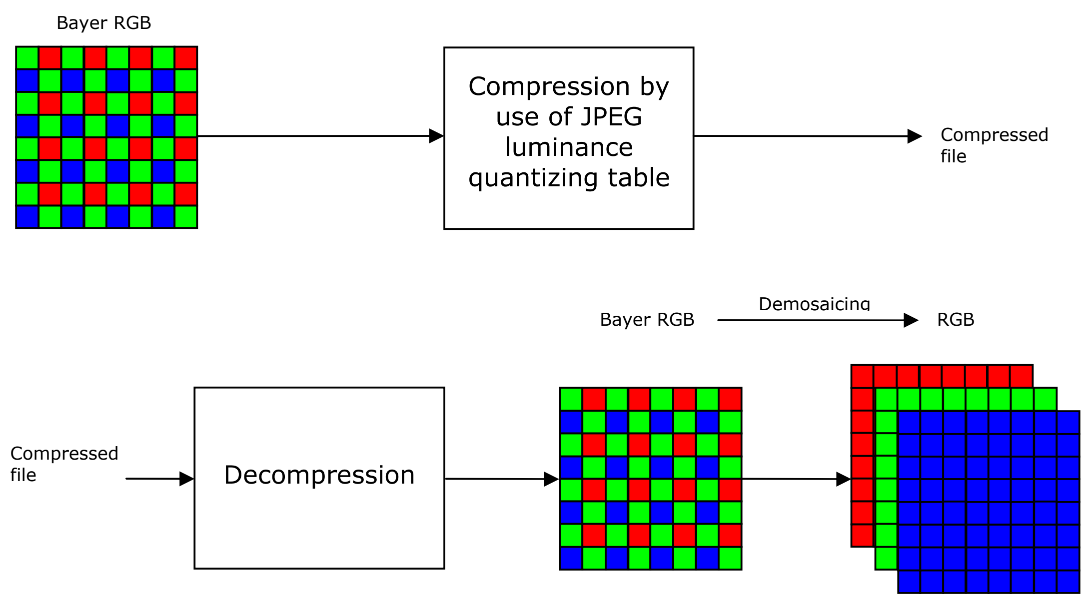

Figure 17 – Bayer RGB as input to the compression scheme

The biggest advantage by doing the image processing this way is the computational savings. When using Bayer RGB as input to the compression scheme, the demosaicing operation and RGB to $Y'C_BC_R$ conversion will become redundant (see *Figure 14* and Figure *17*). Demosaicing is a resource demanding operation, and therefore this improved image processing chain demands less computational power than the classical JPEG. The fact that the data amount to be processed is reduced by 2/3rd compared to classical JPEG reduces the computational requirements additionally.

Another aspect of having the images converted into $Y'C_BC_R$ color space in classical JPEG is $Y'C_BC_R$ 4:2:0 downsampling. This is a property that leads to quality loss as the chrominance information is reduced by a ratio of 4:1. The advantage by doing this is that the file size is reduced considerably.
However, none of these operations are feasible when using Bayer RGB as input to the compression scheme since the image is represented as one array, not three. Bayer RGB can therefore not be compressed at the same degree as JPEG chrominance (C_B and C_R), this will compress the image too much and important luminance information (Y') will be lost. As the human eye is more sensitive to luminance than chrominance, the most

obvious solution is to threat the Bayer RGB as luminance information; this will prevent the image from losing important information.

Another drawback by using Bayer RGB as input to the compression scheme is that high frequency components will be introduced when compressing color images. Those frequencies arise because of Bayer RGBs nature; here's an example:

When acquiring an image which is completely red (all red pixels have the value 255, the other colors have the value 0), the first line of the Bayer RGB pattern will be like this: 0 255 0 255 0 255 0 255…(G R G R G R G R). Now, when transforming this pattern, a maximum frequency component of the image will appear. This will be the only frequency information representing the image and will prevent us from using JPEGs chrominance quantization table since it removes any high-frequency information. With this in mind the critical frequency information will be lost, leaving the reconstruction of the red color Bayer RGB image as an impossible task. The image will be totally distorted and blocking artifacts will occur. For JPEG this image will not experience any problems since, in the example, the red array will be interpolated prior to the discrete cosine transformation of the image. The result after the discrete cosine transform will therefore consist of low frequency information only. This information is easy to compress, contributing to a greater compression ratio.

On the other hand, if the acquired image is greyscale (black/white) or consists of small amounts of color information, Bayer RGB as input will experience a compression ratio like classical JPEG. The reason for this is that R, G and B all have the same value in one specific image sample since transitions between neighboring pixels will be smooth. There will of course not necessarily be smooth transitions between neighbour pixels; this depends on how the actual image looks like.

Using Bayer RGB as input to the compression scheme is also expected to result in additional image compression when it comes to greyscale pictures because occurring high-frequency components as in the example above will not be present. When these frequencies are not present, the probability of the coefficients resulting from the quantization to being zero increases. This implies a smaller file size.

The computational savings by using Bayer RGB is an important advantage. It's hard to determine exactly how much this will affect the computational effort, but it's obvious that the computational effort will be reduced. Here's an example to illustrate this:

An image with a size of 640 x 480 pixels is acquired from a standard CMOS-sensor. If the color depth is 8 bit, 1 byte is needed to represent the red, green or blue value at a given pixel. A Bayer RGB image will then be 640 x 480 bytes, while a classical RGB image will be 640 x 480 x 3 bytes. To compress the image, JPEG will carry out these operations:

- demosaicing: from 640 x 480 bytes to 640 x 480 x 3 bytes

- RGB to $Y'C_BC_R$ conversion: 640 x 480 x 3 bytes

- $Y'C_BC_R$ downsampling: 640 x 480 x 3 bytes to 640 x 480 (Y') + 320 x 240 x 2 (C_B and C_R) bytes

- discrete cosine transform: 640 x 480 + 320 x 240 x 2 bytes

- quantizing: 640 x 480 + 320 x 240 x 2 bytes

- zigzagging: 640 x 480 + 320 x 240 x 2 bytes

- runlengthencoding: 640 x 480 + 320 x 240 x 2 bytes to be processed. The output of this function depends on which quantization table that is used. The table affects the number of zeros to be runlengthencoded. Accordingly, it also affects the amount of data resulting from the runlengthencoding. How the visual image looks like is also an important factor.

- huffmancoding: The data amount to process depends on the distribution of several occurring values. Many appearances of specific values lead to higher compression, but if the number of values is spread the compression ratio will be lower.

For Bayer RGB, these operations are performed for the same image:

- discrete cosine transform: 640 x 480 bytes

- quantizing: 640 x 480 bytes

- zigzagging: 640 x 480 bytes

- runlengthencoding: data amount to process varies as for conventional JPEG

- huffmancoding: data amount to process varies as for conventional JPEG

We see that JPEG performs the resource demanding demosaicing operation, RGB to $Y'C_BC_R$ conversion and $Y'C_BC_R$ downsampling. These operations become redundant when using Bayer RGB as input. We also notice that the resource demanding DCT and the quantization operation for JPEG processes twice as much data compared to Bayer RGB.

Operations like zigzagging and runlengthencoding are not as resource demanding as DCT and demosaicing. Zigzagging is only a relocation of bytes, while runlengthencoding only counts the number of consecutive zeros and inserts values that are not zero.
For DCT and demosaicing, several mathematical operations have to be performed for each pixel as described in *chapter 3*.

By summarizing the advantages and disadvantages of using Bayer RGB as input instead of classical RGB, we see that the size of the compressed file will be approximately the same for both methods (with the same image quality for both). This will naturally depend a little bit on how the image to be processed looks like: Images consisting of large areas of low frequency information will be better compressed when using classical JPEG instead of Bayer RGB as input because unnatural high frequencies will arise when Bayer RGB is used as input. This restrains the compression ratio.

The main advantage by using Bayer RGB as input is that the computational effort is reduced considerably compared to classical JPEG. For mobile terminals this is very important in order to keep the power consumption as low as possible and not use resources that other functionality of the mobile terminal might make use of.

4.4 Composite Bayer RGB as input to the compression scheme

In this approach the Bayer RGB image is split into the respective color planes red, green and blue. The composite version of the Bayer RGB image is sent directly to the input of the JPEG encoder. An illustration of the segmentation of the Bayer RGB pattern and what forms the input of the JPEG encoder, is given in Figure *18*.

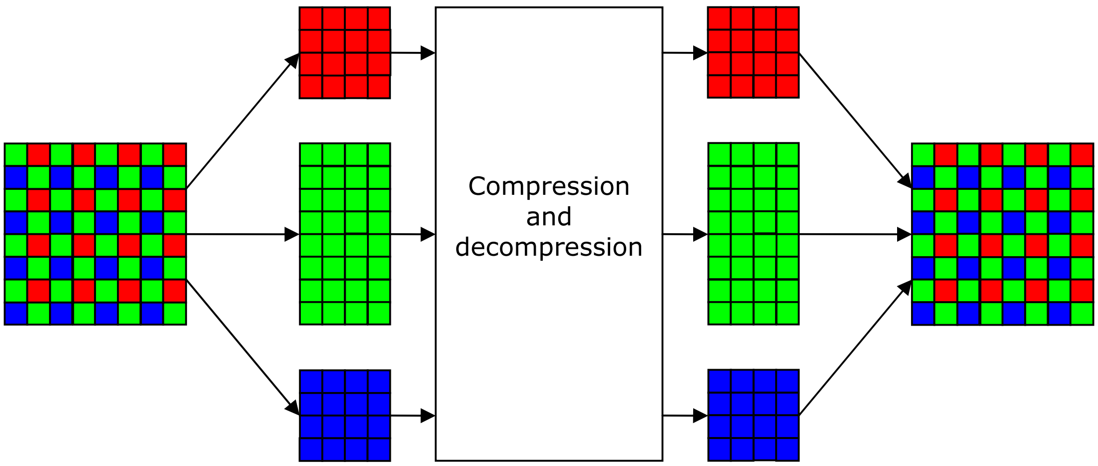

Figure 18 – Composite Bayer RGB as input to the compression scheme

Due to the eyes higher sensitivity to green and because green contains most luminance information, the Bayer RGB pattern consists of twice as many green pixels as red and blue pixels to pick up as much luminance information as possible and still retain chrominance information.
The advantage of composite Bayer RGB as input to the compression scheme is that a composite Bayer RGB image, with regard to luminance and chrominance information, can be seen as a parallel to $Y'C_BC_R$ representation of images containing less data. This parallel could not be drawn when using Bayer RGB as input since the image was kept in its original form and not separated into the three additive color planes.

Having the Bayer RGB image split into luminance (green) and chrominance (red and blue) information will result in the use of both the luminance and chrominance quantization tables of the JPEG compression scheme, making it possible to compress the color information further without affecting the luminance information. Compared to Bayer RGB as input, this is likely to help achieving a better compression ratio. Visual reconstruction errors can however occur as a result of the contraction of each color array. By contracting the arrays, high-frequency components will appear. When assembling red, blue and green pixels from a Bayer RGB image into three new color segments, new frequencies will occur in each of them. This is a result of the fact that neighbour pixels in the resulting image are located with a certain distance between each other in the original image. This will cause smooth variations in the original picture to appear as relatively big differences between neighbour pixels in the three composite color segments. By quantizing the three color segments at different degrees, the introduced high frequency components will also be affected differently for each color. It's not easy to predict how this will affect the image visually, but by removing high frequency components due to this phenomenon, naturally occurring high frequency components in the original image will disappear too. This results in the loss of certain frequency features in the reconstructed image making it looking blurred.

In the example in *chapter 4.3* where a completely red image was acquired and then compressed, high-frequency components were introduced for Bayer RGB because of its nature. For composite Bayer RGB on the other hand, images with monochrome areas will not be introduced to high-frequency components because neighbouring pixels of the same color does not differ much from each other. This leads to a higher compression for composite Bayer RGB when the acquired picture consists of monochrome areas.

Considering the relative amount of green color information compared to red and blue, the question arises on whether it is possible to make use of one of the standard luminance quantization tables used for JPEG or if it is necessary to redefine or make adjustments to the existing tables.

Having made these assumptions, the degree of luminance quantization appears to be the most challenging hurdle to take. Further throughout the JPEG compression algorithm, the effect of using composite Bayer RGB as input is not expected to introduce problems. This means that, apart from the adjustments to the luminance quantization table in JPEG, the JPEG compression scheme can be used as it appears without introducing peculiarities or changes to the compression algorithm.

The computational effort when using composite Bayer RGB as input to the compression chain is assumed to be of same proportions as for regular Bayer RGB. Operations to be performed are the same as for Bayer RGB, but for three different color segments. However, the total number of pixels in the three segments is the same as for one single Bayer RGB array, leading to the same computational effort as for Bayer RGB as input to the compression scheme.

5 Problem solving Environment

5.1 Introduction

To confirm the hypotheses presented in the previous chapters, a problem solving environment is needed. This chapter presents the test environment and its functionality in addition to simplifications that is done.

The test environment in this thesis is built by use of MATLAB 6.0. This powerful simulation program is primary used for digital signal processing, including image processing.

One great advantage by using MATLAB instead of using a standard C/C++ compilator is the built-in feature 'Image Processing Toolbox'. This is a toolbox containing functions adapted to image processing. Several functions such as the two dimensional DCT transform, functions that load and save images and functions to view images are included. This results in considerable savings when it comes to implementation of code. If our code should be carried out through C/C++ it would probably be necessary to write many times as much code as if it were implemented in MATLAB.

A disadvantage by using MATLAB as we see it is that it is a resource demanding program and requires a relatively fast computer with a respectable amount of memory.

Another disadvantage is that it is pretty difficult to measure the computational cost of the operations we perform. In earlier versions of MATLAB, there was an operation called 'flops' which counted the number of floating-point operations.
In MATLAB 6.0, matrix computations are done by LAPACK. LAPACK is a "multiauthor Fortran subroutine library for numerical algebra" according to the MATLAB documentation. "LAPACK extends MATLAB's matrix computation capabilities and increases its speed on larger problems."
By introducing LAPACK to MATLAB, it is no longer possible to count floating-point operations.

However, since it is not the scope of this thesis to measure the computational costs for our operations, computational efficiency has not been focus when writing the functions. They are therefore not optimized for computational cost and processing large images can take a while.

The test environment is constructed by use of classical pre-processing and JPEG blocks such as demosaicing, $Y'C_BC_R$ to RGB conversion, DCT, quantization and runlength coding as explained in *chapter 3*. Preprocessing blocks like white balance, color correction and gamma correction are however skipped as they are not important for the test environment. The reason for this is that all the three image processing chains that have been implemented, investigated and compared in our work, all use the result of the demosaicing operation; the interpolated RGB image as reference as shown in *Figure 19* on the next page.

Both the conventional JPEG compression chain and the improved processing chains with Bayer RGB as input use the same interpolated image as reference. As demosaicing operation we have chosen to implement the cubic interpolation kernel because this is the interpolator which provides the highest quality of the interpolated image. However, this is also the most resource demanding interpolator, but as it has no affection on the test results it's obvious that this is the most appropriate solution.

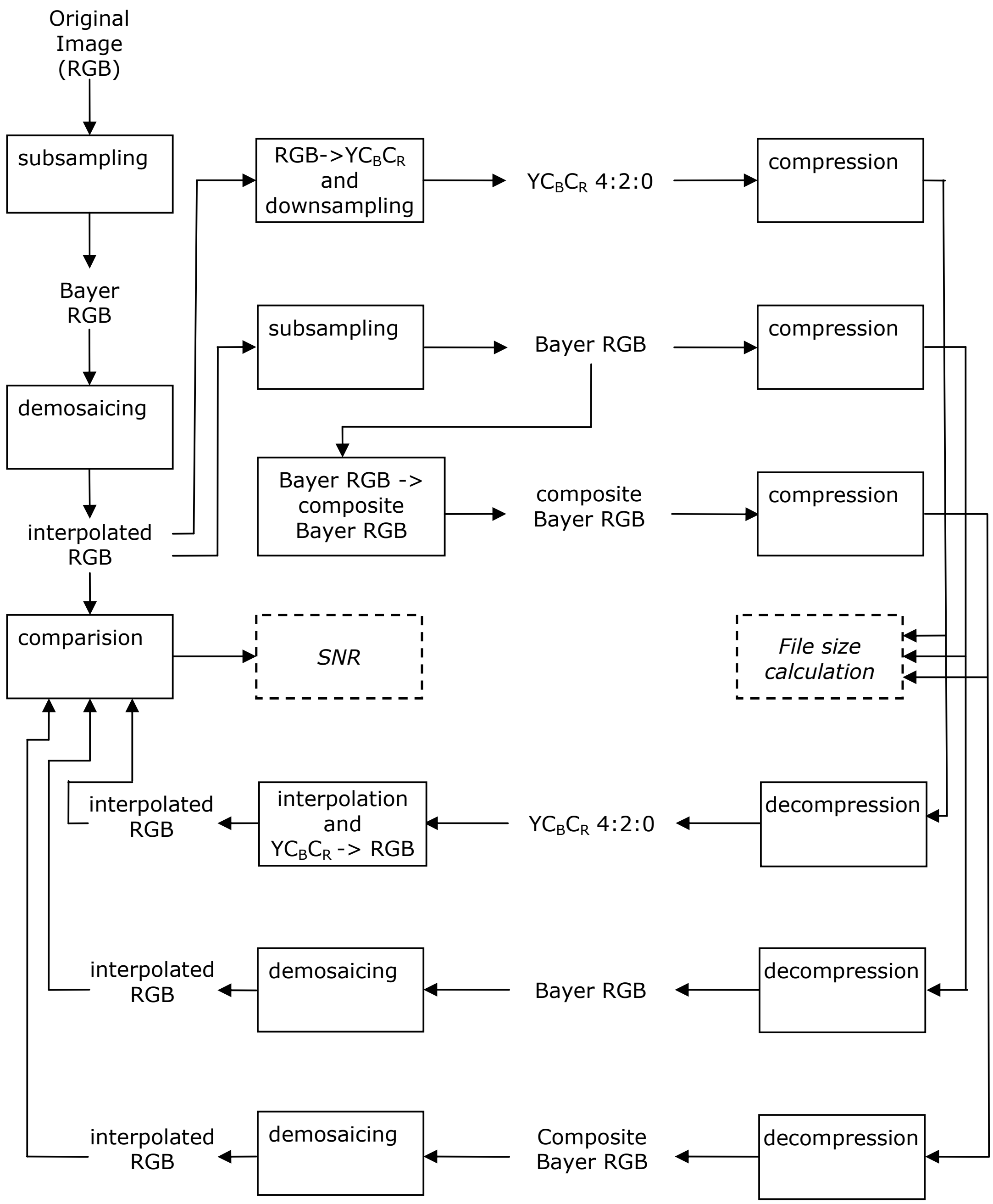

Figure 19 – Test environment

The blocks located subsequent to the demosaicing block in the preprocessing chain (*Figure 5*), color correction and gamma correction, become unnecessary because the test pictures we use already are gamma corrected, white balance adjusted and color corrected.

One important block in the image compression chain, the Huffman coding block, has also been left out as it, through literature understandings, has proved insignificant to this specific thesis. The block provides additional image compression, but is complicated to implement as it is a dynamic compression algorithm. It is of course possible to use a predetermined Huffman tree, but that would imply an erroneous difference in the compression ratio for different images. The reason for this is that each Huffman tree is generated by means of properties of the picture to be compressed.

5.2 Short descriptions of files

All of the following file descriptions have their corresponding file implementations listed in *Appendix B*. Please apply to this appendix for more details on the files.

bayer2red.m / bayer2green.m / bayer2blue.m:

The task of these files is to extract red, green and blue pixels respectively from the Bayer RGB pattern into the new color arrays red, green and blue like in Figure *18.*

bayer2rgb.m:

This file separates a Bayer RGB image into three color arrays, making it possible to view the Bayer RGB image as a color picture.

dct.m:

DCT-transforms an image from the spatial domain to the frequency domain.

demosaic.m:

Interpolates a Bayer RGB image by interpolating the red, green and blue arrays into a RGB image.

downsample.m:

Downsamples an image by 2 in both horizontal and vertical direction.

error_bayer_image.m:

Calculates the SNR of a Bayer RGB image, resulting in a new image.

error_image.m:

Calculates the SNR of a RGB image, resulting in a new image.

idct.m:

This file performs the inverse DCT transform on a picture.

irunlength.m:

This file decodes a runlengthencoded data stream.

quantize.m:

This function quantizes an 8 x 8 blocks of data by means of a quantization matrix.

redgreenblue2bayer.m:

Transforms a composite Bayer RGB image into an ordinary Bayer RGB image.

rgb2bayer.m:

This file converts a RGB image to a Bayer RGB image by fetching the appropriate pixels into a new array.

rgb2yuv.m:

Converts an image from RGB format to $Y'C_BC_R$ (YUV) format.

runlength.m:

Runlengthencodes an image.

snr.m:

Calculates the SNR of a RGB picture.

snr_bayer.m:

Calculates the SNR of a Bayer RGB picture.

unquantize.m:

Unquantizes an image by means of the quantizing matrix.

unzigzag.m:

Unzigzags an 8 x 8 block of pixels.

upsample.m:

This function upsamples a picture by 2 in both horizontal and vertical direction.

yuv2rgb.m:

Converts a picture from $Y'C_BC_R$ (YUV) color space to the RGB color space.

zigzag.m:

Zigzags an 8 x 8 block of pixels.

test.m:

This is the 'main' file which loads an image and processes the image by classical JPEG, Bayer RGB, and composite Bayer RGB coding and decoding as in *Figure 19*. SNR is also calculated for the different image processing chains. Here's a brief description of what the file does in chronological order:

- loads an RGB image from a specified file.
- converts the RGB image to a Bayer RGB image.
- interpolates the Bayer RGB image to a RGB image. The interpolated RGB image is the first common image format for the three image processing chains, and is therefore used as reference for our quality measurements.

- START of classical JPEG compression chain.
- transforms the interpolated RGB image to the $Y'C_BC_R$ plane.
- downsamples C_B and C_R by 2 ($Y'C_BC_R$ 4:2:0).
- DCT transforms the 3 'images' Y', C_B and C_R.
- quantizes the DCT transformed images. One quantization matrix for Y', and one for C_B and C_R.
- zigzags the quantized images.
- runlengthencodes the zigzagged images.
- calculates the size of the compressed images (Y', C_B and C_R).
- runlengthdecodes the compressed images, resulting in zigzagged images.
- unzigzags the zigzagged images, resulting in quantized images.
- unquantizes the quantized images, resulting in DCT transformed images.
- inverse DCT transforms the DCT transformed images, resulting in Y', C_B and C_R.
- upsamples C_B and C_R by 2.
- transforms the image from $Y'C_BC_R$ to RGB.
- transforms the RGB image to Bayer RGB.
- performs quality measurements. SNR of the reconstructed RGB and the reconstructed Bayer RGB are calculated. Error images are also printed.
- END of classical JPEG compression chain.

- START of Bayer RGB compression chain.
- DCT transforms the Bayer RGB image.
- quantizes the DCT transformed image.
- zigzags the quantized image.
- runlengthencodes the zigzagged image.
- calculates the size of the compressed image.
- runlengthdecodes the compressed image, resulting in a zigzagged image.
- unzigzags the zigzagged image, resulting in a quantized image.
- unquantizes the quantized image, resulting in a DCT transformed image.
- inverse DCT transforms the DCT transformed image, resulting in a Bayer RGB image.
- interpolates the Bayer RGB image into a RGB image.
- performs quality measurements. SNR of the reconstructed RGB and the reconstructed Bayer RGB is calculated. Error images are also printed.
- END of Bayer RGB compression chain.

- START of composite Bayer RGB compression chain.
- transforms the Bayer RGB image into the three ’images’ red, green and blue.
- DCT transforms the images.
- quantizes the DCT transformed images. One quantization matrix for green, another one for red and blue.
- zigzags the quantized images.
- runlengthencodes the zigzagged images.
- calculates the size of the compressed images.
- runlengthdecodes the compressed images, resulting in zigzagged images.
- unzigzags the zigzagged images, resulting in quantized images.
- unquantizes the quantized images, resulting in DCT transformed images.
- inverse DCT transforms DCT transformed images, resulting in a composite Bayer RGB image.
- converting the composite Bayer RGB image to a Bayer RGB image.
- interpolates the Bayer RGB image into a RGB image.
- performs quality measurements. SNR of the reconstructed RGB and the reconstructed Bayer RGB is calculated. Error images are printed.
- END of composite Bayer RGB compression chain.

6 Performance and quality validation

6.1 Introduction

The motivation for trying to find a suitable way of performing quality validation of processed images is to preserve validity for the test results generated in the test environment of this thesis. Literature describes several quality metrics with appropriate use for validating image quality, ranging from objective mathematical methods to complex perceptive implementations of the human visual system. The model suitable for validating test results put forward in this report is able to prove mathematical correlation between images and to point out changes introduced by the image processing system.

There are three major types of quality measurements [Woolley et.al., 2001]: objective, subjective and perceptual measurement. In the following a brief description of the three methods is given.

- Objective quality metrics provide mathematical deviations between original and compressed images. A number of mathematically defined metrics have been used in the literature, including signal-to-noise ratio (SNR), peak signal-to-noise ratio (PSNR), mean absolute error (MAE), mean squared error (MSE) and local mean squared error (LMSE).

- Subjective quality measures provide numerical values that quantify viewers' satisfaction with reconstructed images. This type of quality measurement, however, as previously mentioned, involves time-consuming experiments as different viewers have to observe images and observer responses can vary significantly.

- Perceptual quality measures are based on models of human visual perception. It incorporates the aspect that the model itself should able to predict threshold visibility levels of distortion based on psychophysical and physiological evidence. Even though a perceptual aspect of validating image quality results in a more reliable outcome compared to pure mathematical validation models, in the sense of not only considering "correlation quality" or "local image quality", this quality measure also involves a very crude implementation complexity in the effort of modelling the human visual system as a series of linear and non-linear stages. Without going into further details, this model includes a variety of techniques like contrast sensitivity function (CSF), decomposition into multiple frequency bands, luminance adaptation, contrast masking, distance computation and summation of errors across frequency bands and space.

One of the problems associated with validation of image quality metrics is that the "gold standard" is a human observer. This means that the accuracy and robustness of a perceptual metric is closely tied to the psychophysical experiments used to validate the metric. Unfortunately, there seems to be as many psychophysical techniques to validate metrics as there are metrics.

Even though objective quality validation methods are easy to calculate and perform well when using images with constraints on the image content, they are poor predictors of distortion visibility since their implementation do not consider human visual sensitivities. However, in the quality validation of test results put forward in this report, due to the time consumption of subjective quality metrics and the model complexity of perceptual quality measures, objective quality metrics will be the validation tool implemented. This quality metric is described in further detail in the following chapter.

6.2 Signal-to-noise ratio

Even though extensive evaluation of SNR has shown that it does not work well across images which contain significantly different content, the primary advantage of this quality metric is its ease of use. As with other mathematical quality metrics it does not require any information about viewing conditions, does not adapt to local image content and the computations are simple. The fact that a metric is defined without consideration for visual factors does not make visual factors insignificant; rather, it simply ignores them and relies on luck to provide correlation with perceived quality. With this in mind, the conclusions drawn in this report will not solely depend on the numerical values provided for by the SNR. Error images will form an additional foundation when considering the quality of a processed image. This is done to possibly point out features that the SNR numerical values cannot reveal when discussing the satisfaction with reconstructed images.

In the test environment of this report, the formula of SNR [Sakamoto et.al., 1998] below is implemented in order to quantify mathematical reconstruction quality of processed images. The results are given in *chapter 7*.

$$SNR = -10\log_{10}\left[\frac{\frac{1}{KL}\sum_{k=0}^{K-l}\sum_{l=0}^{L-1}\{f(k,l)-f'(k,l)\}^2}{255^2}\right] \quad [\mathrm{dB}]$$

Equation 10 – Signal to Noise Ratio (SNR)

In this equation, the product KL refers to the spatial resolution of the image. For a 640 x 480 pixels image, K is 640 and L is 480 or vice versa. f(k,l) represents the original image and f'(k,l) represents the reproduced image. The numerator represents the squared average pixel value for the image, while the denominator represents the squared color resolution.
Because the color depth is 8 bit, 255 different colors can be described and the denominator will be 255^2.

7 Results

7.1 Introduction

The results from simulations in the test environment in *chapter 5* with respect to performance and quality measurements described in *chapter 6* are summarized in this chapter. SNR, compression ratio and file size for the test sets are presented in this chapter, while reconstructed images from the simulation are presented in *Appendix A*.

7.2 Simulation results

In the tables in the following chapter, the simulation results are presented. There are three test sets, with different images as input. For each test set three different inputs are used; classical JPEG, Bayer RGB and composite Bayer RGB.
For each of the inputs, three setups of quantization matrices are used, denoted by 'Setup 1, 'Setup 2', and 'Setup 3'. The matrices are specified in *Appendix A* and are taken from ITU-T's standard recommandations.
For each of the setups the file size of the compressed image, the signal-to-noise ratio (SNR) and the compression ratio are measured. By using the 'setup 1' file size of classical JPEG as reference, the relative compression is calculated. The SNR is calculated as in *Figure 19*, using the interpolated RGB image as reference for the reconstructed RGB image.

For visual quality validation, reconstructed images and error images are presented in *Appendix A*.

Due to the implementation architecture of using Bayer RGB as input to the compression scheme and as described in *chapter 4.3* the tests on Bayer RGB require only one quantization matrix. For Bayer RGB the given luminance matrix in *Appendix A* is therefore used as quantization matrix.

7.2.1 Test images

In terms of performance, for easier to observe the main differences between classical JPEG, Bayer RGB and composite Bayer RGB as input to the compression scheme, we have used three test pictures with different properties adapted to emphasize qualities of the different image processing chains.

Test set 1:
For test set 1 in *Appendix A*, we have used a 'circular zone plate' as test image. See *figure 1* in *Appendix A*. This is a black/white image where the lowest frequency is located in the corner down to the left. The longer away from the corner we are, the higher the frequency is.

Test set 2:
The test picture used in test set 2 is a picture of a car, and consists of many colors and high-frequency components. See *figure 21* in *Appendix A* for the picture 'car'.

Test set 3:
For this test set we have chosen the image 'winter', which is an image consisting of large monochrome areas and some details. *Figure 41* in *Appendix A* shows the image 'winter'.

The picture 'czp' is a .bmp file, while the pictures 'car' and 'winter' are originally .bmp files, but they are resized and stored as lossless .jpg files.

Test set 1:

File: czp.bmp
Dimensions: 320 x 240 pixels
Uncompressed file: 230400

Classical JPEG:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	156762	33,9	1:1	1:1
Setup 2	56432	31,9	2,78:1	2,78:1
Setup 3	36534	30,0	4,29:1	4,29:1

Table 2 - Test results for classical JPEG, czp.bmp

Bayer RGB:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	116864	57,6	1:1	1,34:1
Setup 2	53190	44,7	2,20:1	2,95:1
Setup 3	35454	37,5	3,30:1	4,42:1

Table 3 - Test results for Bayer RBG as input, czp.bmp

Composite Bayer RGB:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	135342	57,6	1:1	1,16:1
Setup 2	54378	33,9	2,49:1	2,88:1
Setup 3	38756	28,9	3,49:1	4,04:1

Table 4 - Test results for composite Bayer RGB as input, czp.bmp

Test set 2:

File: car.jpg
Dimensions: 256 x 256 pixels
Uncompressed file: 196608

Classical JPEG:

Compression	File size	SNR[dB]	Compresison	Relative compression
Setup 1	149690	33,0	1:1	1:1
Setup 2	44570	30,1	3,35:1	3,35:1
Setup 3	24590	28,7	6,10:1	6,10:1

Table 5- Test results for classical JPEG, car.jpg

Bayer RGB:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	116310	58,1	1:1	1,29:1
Setup 2	56064	40,0	2,07:1	2,67:1
Setup 3	28568	32,3	4,10:1	5,24:1

Table 6 - Test results for Bayer RGB as input, car.jpg

Composite Bayer RGB:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	121534	58,0	1:1	1,23:1
Setup 2	42392	33,8	2,87:1	3,53:1
Setup 3	25656	29,6	4,74:1	5,83:1

Table 7 - Test results for composite Bayer RGB as input, car.jpg

Test set 3:

File: winter.jpg
Dimensions: 320 x 240 pixels
Uncompressed file: 196608

Classical JPEG:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	150152	37,4	1:1	1:1
Setup 2	32130	34,6	4,67:1	4,67:1
Setup 3	16830	32,9	8,92:1	8,92:1

Table 8 - Test results for classical JPEG, winter.jpg

Bayer RGB:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	134450	57,5	1:1	1,12:1
Setup 2	48692	40,4	2,76:1	3,08:1
Setup 3	25502	34,1	5,27:1	5,89:1

Table 9 - Test results for Bayer RGB as input, winter.jpg

Composite Bayer RGB:

Compression	File size	SNR[dB]	Compression	Relative compression
Setup 1	128318	57,3	1:1	1,17:1
Setup 2	25644	38,0	5,00:1	5,86:1
Setup 3	13728	34,9	9,35:1	10,94:1

Table 10 - Test results for composite Bayer RGB as input, winter.jpg

8 Discussion

8.1 Introduction

The motivation for this chapter is to discuss several issues that have arisen throughout the work. In order of appearance these issues are:

- How the test results correspond to literature reviews
- The validity of test results
- Bayer RGB versus composite Bayer RGB
- Classical RGB versus simplified image processing chain
- JPEG 2000 – a future framework for image processing in mobile terminals?

Concerning the test results put forward in this thesis it is interesting to see whether they correspond to literature reviews done in the preliminary study of this thesis. In addition, the validity of the results put forward is an important issue to discuss as it proves consistency of the work presented. Next, the behaviour and performance of Bayer RGB and composite Bayer RGB compared to conventional JPEG is discussed. Towards the end of this chapter, a brief presentation of the newly released image processing standard JPEG 2000 is given. JPEG 2000 introduces a new technique and, accordingly, new processing capabilities to the field of image processing. Even though a detailed study of this standard is beyond the scope of this thesis, a short discussion on how JPEG 2000 performs compared to the current standard is given. The motivation for this lies within the promising future aspects concerning image processing associated with the standard.

Furthermore, this chapter will be held as basis for the next chapter; the conclusion.

8.2 Test results

Test image 'czp'

Starting with the CZP image from test set 1 in *Appendix A* we see from series 1 in *chapter 7.2* that for setup 1 where the quantizing matrices are set to 1, Bayer RGB as input experiences the highest compression and also the highest SNR. In relation to JPEG, Bayer RGB experiences a file size reduction of 25% and the SNR is almost twice as high.

Notice that for Bayer RGB as input both the compression ratio and the SNR is higher than for conventional JPEG compression when using CZP as input. The reason for this is what is described in *chapter 4.3* regarding grayscale pictures.

From *figure 14* in *Appendix A*, which is the error image for 'czp' when using composite Bayer RGB as input, it's obvious that high-frequency components in the vertical direction are better conserved than high-frequency components in the horizontal direction. This is a result of the division of Bayer RGB into the 3 color arrays red, green and blue. The green color array's resolution is twice as large in vertical direction as in horizontal direction (see Figure *18*), and this affects the compression: High frequencies in vertical direction experience a higher SNR than those in horizontal direction.

After having simulated the image processing of the circular zone plate we discovered undesired artifacts (rings) in the error images as in *Figure 20*:

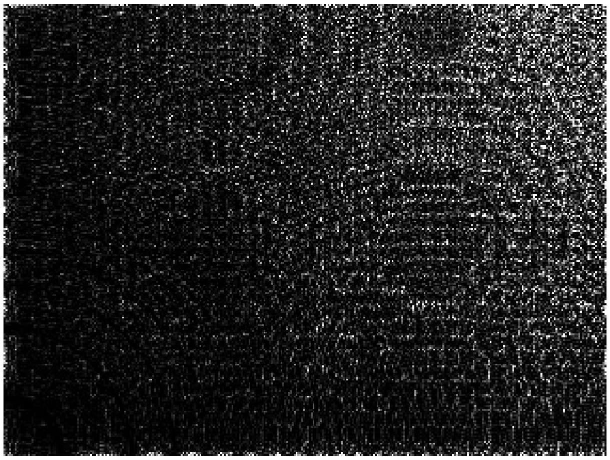

Figure 20 – Error image for CZP with aliasing artifacts

By comparing the error images of the three test results in set 1, setup 3 in *Appendix A* it is apparent that these effects only appear in compression using classical JPEG. These rings are aliasing occurring as a result of decimation, or RGB to Bayer RGB conversion, without first having lowpass filtered the image. Lowpass filtering of an image previous to processing can be done to ensure that the sampling frequency satisfies the Nyquist criterion.

Test image ‘winter’

For the ‘winter’ image, a large amount of it is blue shaded. For Bayer RGB this is not beneficial, because new frequencies will be introduced. This is in accordance to *chapter 4.3*, where a single colored Bayer RGB compressed image is discussed. The result of this is a lower compression ratio.
We see from the simulation results in *chapter 7.2* that composite Bayer RGB is being compressed almost twice as much as Bayer RGB for the ‘winter’ image. However, for the images ‘czp’ and ‘car’, the compression ratio for Bayer RGB and composite Bayer RGB is almost equivalent. This is because those images don’t consist of large areas of the same color, like the ‘winter’ image does.

From the 'winter' image, we can also see that the reconstructed composite Bayer RGB image (test set 3, setup 3) is blurred in detailed areas. This is shown in *Figure 21* below.

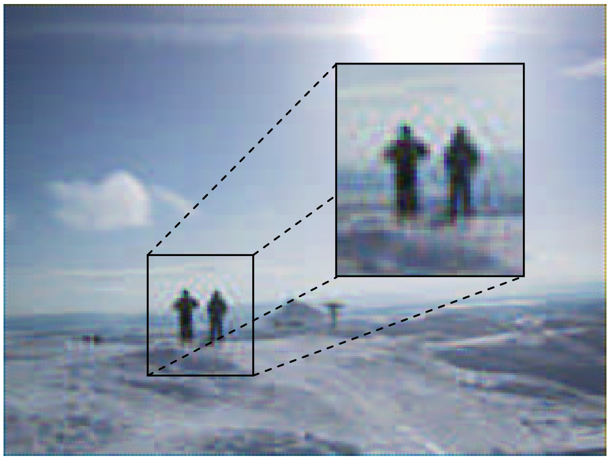

Figure 21 – Blurred composite Bayer RGB compressed 'winter' image

When the image is compressed by using Bayer RGB as input to the compression scheme, those artifacts are not present (as shown in *Figure 22*). By looking at the snow in this image, it can easily be seen that the composite Bayer RGB image is smoother than the Bayer RGB image. The reason for this is that introduced high-frequency components are suppressed by the JPEG adapted quantizing matrices, as explained in *chapter 4.4*. When high-frequency components of the compressed composite Bayer RGB image are suppressed the reconstructed image will lose high-frequency information, such as those mentioned above.

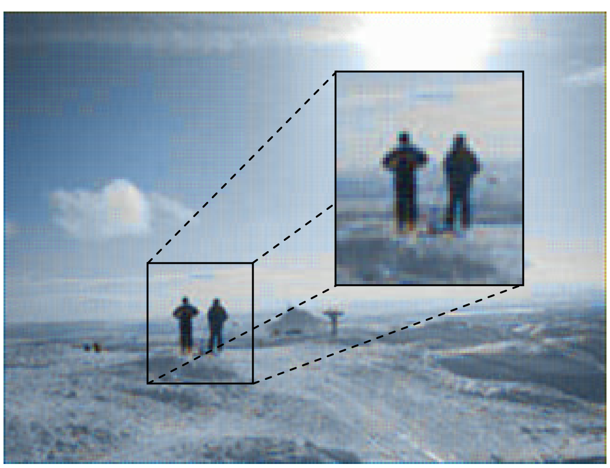

Figure 22 – Bayer RGB compressed 'winter' image

Both *figure 21* and *figure 22* are taken from test set 3, setup 3. The quantization matrices we have used are taken from CCITT's recommendations and are adapted to fit JPEG. High-frequencies occurring in Bayer RGB and composite Bayer RGB are as a result of this discarded more or less, and imaging artifacts are introduced.

Test image 'car'

The image 'car's properties are a mixture of the properties of the 'czp' and 'winter' images, and is in addition built up by many different colors.
For this image, composite Bayer RGB yields the lowest quality of the reconstructed image. That is a result of the properties of composite Bayer RGB which makes high-frequency components blurred. The image 'car' consists of much high-frequency information, and as result the picture will be blurred.

Notice that for the composite Bayer RGB compressed image the SNR is higher than for JPEG, while the visual reconstructed image's quality is poorer. The reason for JPEGs relatively low SNR is that its reproduction capabilities when it comes to colors are bad, according to *chapter 2.4.6* and *2.4.7*, and this lowers the SNR for the reconstructed JPEG image considerably. The poor visual reconstruction quality of the composite Bayer RGB processed image is a result of composite Bayer RGBs bad performance when it comes to high-frequency information, something which lowers the SNR.

Bayer RGB, on the other hand, achieves a better image quality than classical JPEG. For setup 2 Bayer RGB's SNR is 40 while it is only 30.1 for JPEG. It's important to take into account that the JPEG image is compressed at a ratio of 3.35:1, while the Bayer RGB image is compressed at a ratio of 2.67:1. From the error images, it's obvious that the visual quality of the Bayer RGB image is far better than the quality of the JPEG compressed image.

Summary

For all the three test sets we see that Bayer RGB and composite Bayer RGB experience a much higher SNR than classical JPEG when the quantizing matrices are set to 1.
According to *chapter 2.4.6* and *2.4.7*, JPEGs relatively low SNR is a result of quality loss due to RGB to YUV conversion and downsampling.
However, when increasing the compression, the SNR for Bayer RGB and composite Bayer RGB decreases drastically. This is a result of the nature of the quantizing matrices, which in this case are adapted to JPEG. High frequency components are introduced when using Bayer RGB or composite Bayer RGB as input to the compression scheme, and those components will be discarded more or less by the JPEG adapted quantizing matrices.

8.3 Test result validity

Since the pure mathematical nature of SNR does not work well across images containing significantly different content, the test results put forward in this thesis also feature a subjective image quality validation in terms of perceived image reconstruction quality for processed images. By comparing reconstructed images with their corresponding error images many aspects on how the compression scheme performs become apparent. This additional evaluation of performance is based on the processing of images containing different characteristics. The choice of images is done with respect to important image characteristics like varying frequency content, motifs with large monochrome areas and greyscale properties of motives. This way, the performance of the simplified image processing chain can be evaluated for a relatively wide spectrum of inputs.

However, despite the fact that SNR as a quality measure in some contexts has been criticized to add little or no substance at all to the validity of test results, it has in this thesis proved good conformity between actual results, discussion and literature reviews.

Here, it is important to notice that the main scope of this thesis is to investigate and discuss the functionality and operability of the simplified image processing chain. In this sense, the SNR is a sufficient quality measure for this field of operation.

Furthermore, when it comes to compression ratio, this is adopted as a quality measure in the work in order to reflect the actual compression capabilities of the simplified image processing chain. The image file size is calculated and compared in each setup with the file size of the processed image where no compression is applied. In order to preserve test result validity towards conventional JPEG, relative compression ratio is also calculated. Relative compression ratio is calculated the same way as compression ratio. The only difference here is that the calculated file sizes are being compared with the file size of the processed image with JPEG as input and no compression applied.
It is important to notice that these two measures are calculated on basis of pure image data. That is, no header information is applied to the image files processed in the work. The insertion of such header information, however, is not expected to cause changes to the compression ratio as the header is very small compared to the image data contained in the files.

8.4 Bayer RGB vs composite Bayer RGB

When it comes to computational effort, the two compression schemes are approximated to be equal. This approximation is based on the fact that both compression algorithms have the same amount of image information to process. Furthermore, both image processing chains implement the same processes. The only difference is that composite Bayer RGB is split into three different color arrays. This, however, has no effect on the total number of image samples to process.

Test image 'winter'

The comparison of reconstructed images due to different setups discussed in the previous chapter states some interresting results. For the test image 'winter' no visual errors are introduced in the case of lossless compression (setup 1), that is, both quantization matrices are set to 1 leading to a round-off of coefficients to the nearest integer. For the image 'winter', which mostly can be regarded as a monochrome image, the reconstructed images using composite Bayer RGB as input prove an overall better compression ratio (almost double in some cases) compared to Bayer RGB with the corresponding SNR being left with only small variations. However, despite of this being an uplifting result it must be emphasized on the fact that the corresponding error images prove additional aspects of reconstruction that are essential for the perceived quality of the reconstructed images. These aspects are, as discussed in the previous chapter, poorer visual reconstruction quality for details when the ability to compress and recreate monochrome areas in a picture are good and the other way around. Despite of this, considering the error images, it becomes obvious that Bayer RGB eventually results in the best reconstructed quality. On the other hand the reconstructed file size of composite Bayer RGB is only half the size of Bayer RGB's file size. This severe reduction of the file size, as discussed earlier in *chapter 8*, is due to composite Bayer's additional use of the chrominance quantization matrix.

Test image 'czp'

The image ‘czp’ is included in the simulations because of its sweeping frequency nature. As with the test image ‘winter’ the image ‘czp’ does not introduce visual errors during simulation without any compression. Naturally, this is also obvious through a good SNR. The interesting part is rather how the change of motif affects the compression ratio. Compared to images with small and slow variations in motifs the ‘czp’ has more high frequency content making the obtained compression ratios for monochrome images impossible to achieve for the ‘czp’. When it comes to processing of the increasing frequency content of ‘czp’ the compression ratios do not differ considerably using either of the two inputs Bayer- or composite Bayer RGB. Rather, the SNR appears to be overall lower for the case of composite Bayer due to this method using both luminance and chrominance quantization matrices during image processing. Using chrominance quantization matrices will result in higher compression of the red and blue arrays, and hereby lower the SNR compared to Bayer RGB.
Another aspect of processing an image with such nature is that high-frequency components in the vertical direction are better conserved than high-frequency components in the horizontal direction. This is, as previously discussed due to the way composite Bayer distinguishes the colors before compression and is very obvious from the error images.
The files, processed one or the other way, are comparatively equal when it comes to file size. The file size therefore becomes irrelevant in process of determining which compression scheme is better. It is eventually the visual quality of the reconstructed images that proves a better performance of Bayer RGB.

Test image ‘car’

Except for classical JPEG as input, the test image ‘car’ does not result in any visual errors when no compression is applied. Classical JPEG seems to experience tiny errors in certain high-frequency areas of the image due to the round-off of coefficients during quantization. The result of this is small, barely noticeable points of error in the corresponding error image whereas no errors occur for either Bayer RGB or composite Bayer RGB as input.
The main intention of using ‘car’ as one of the test images is to introduce a wider range of colors in the image content to the processing operations. It is therefore interesting to see how the two methods of processing the image perform, especially since composite Bayer RGB has a dedicated quantization matrix for color compression compared to Bayer RGB. It is obvious from the compression ratios that composite Bayer RGB manages to achieve a better compression, in fact an improvement of 39% for setup 1 compared to Bayer RGB. Furthermore, this also explains the reduced file size of composite Bayer RGB.

However, these two aspects prove to be at the expense of visual quality and reduced SNR. As a result the composite Bayer RGB does not seem to be capable of reconstructing high-frequency details without blurring effects appearing in the contours of detailed objects in the image. Bayer RGB on the other hand performs well in this case. It does not achieve the compression ratio and not quite the same reduction of file size as for composite Bayer, but results in a better visual quality of the reconstructed images. Considering the error images, this aspect is very apparent.

Summary

To summarize the results above, even though composite Bayer RGB prove an overall better compression ratio compared to Bayer RGB when it comes to images of monochrome motives, it becomes obvious that Bayer RGB eventually results in the best reconstructed quality. In terms of grayscale images with a wide range of frequency content, Bayer RGB also in this case proves the best visual reconstruction quality of images without this being at the expense of either compression ratio or SNR. Considering images with a wide range of frequency and color content, composite Bayer RGB tend to achieve the best compression ratio and the smallest reconstructed file size. However, composite Bayer RGB does not seem to be capable of good reconstruction quality in these cases. As a result, despite of Bayer RGB not being able to achieve the compression ratio as of composite Bayer RGB, Bayer RGB proves an overall good quality of reconstruction.

8.5 Improved image processing chain vs classical JPEG

When considering these two image processing chains, computational effort is most certainly an issue. However, due to the limited scope of this thesis, computational cost is not considered to be of main interest for investigation since we already know that the improved image processing chain omits the operations demosaic and colorspace conversion and hence reduces on the computational effort. From this, the interesting issue is how the two chains compare in aspects of performance and visual reconstruction quality.

Considering the file sizes of the reconstructed images the dilemma occurs on what file size can be tolerated for Bayer RGB with the compression scheme still being justified by its computational savings. Since computational effort in terms of processing time and power is not considered to be of importance to this thesis, rather the reconstruction quality of images processed with the proposed simplified image processing scheme, the file sizes will not constitute the foundation of conclusions put forward. It is expected that if this improved image processing chain proves to compete as an image processing algorithm, the sometimes comparatively large file sizes will be justified by the introduced computational savings of Bayer RGB.

Test image 'winter'

As discussed in *chapter 8.2* large monochrome areas as in the image 'winter' is not beneficial for Bayer RGB as this introduces high-frequency components. However, even though Bayer RGB in all cases can refer to a poorer compression ratio compared to classical JPEG, Bayer RGB performs better compared to classical JPEG when it comes to SNR and visual quality. Based on this, Bayer RGB therefore seems to appear as a fair comparison to classical JPEG. Indeed, the Bayer RGB does in some cases leave a raster-like pattern in monochrome areas of reconstructed images, but the important thing worth noticing here is the computational savings Bayer RGB involves. In fact, the reconstructed quality of Bayer RGB images do not suffer from errors that make these two compression schemes not worth comparing. The introduction of small errors will easily be justified by the computational savings Bayer RGB involves.

Test image 'czp'

Chapter 4.3 states that a compression scheme using Bayer RGB as input is expected to perform well on black/white or greyscale images. This assumption turned out to be correct. During the processing of the image 'czp' Bayer RGB does indeed result in an overall slightly poorer compression ratio compared to classical JPEG. However, this can be justified by the fact that all the corresponding SNR levels are considerably better for

Bayer RGB with the most dramatic outcome being the result of setup 2, which represents an increase of SNR of more than 40%! Although it is difficult to see the effects of this from the reconstructed images, it is clearly revealed from the error images that errors are being introduced in details with increasing frequency. Therefore, in this case, the Bayer RGB compression scheme outperforms classical JPEG.

Test image 'car'

The test image 'car' represents images containing different features like a wide range of colors and details.

As previously discussed, Bayer RGB proves an overall satisfying performance also in cases where images contain a combination of features. Compared to classical JPEG, Bayer RGB manages to reconstruct images containing less error. Bayer RGB seems to have reconstruction skills exceeding those of classical JPEG when it comes to details. Furthermore, Bayer RGB does not, on the basis of the subjective quality evaluation performed in this thesis, seem to experience lower quality in the appearing monochrome areas of 'car'. In addition the SNR is also considerably better for Bayer RGB. However, despite of these uplifting results, Bayer RGB cannot refer to the same ratios of compression in this case; at worst 62% poorer compression ratio.

Summary

Summarizing the comparison of the improved image processing chain and conventional JPEG, Bayer RGB performs best for images with large monochrome areas when it comes to SNR and visual quality. Considering grayscale images with a wide range of frequency content, Bayer RGB tend to result in the best reconstruction quality and SNR. This, however, appears to be at the expense of an overall slightly poorer compression ratio compared to classical JPEG. In terms of images with a wide range of frequency and color content, Bayer RGB compared to traditional JPEG also in this case performs overall better with the only drawback being a poorer reduction in the reconstructed file size.

8.6 JPEG 2000 – a future framework?

With the explosive emerge of web based services, mobile multimedia and Internet applications the requirements of the image processing technologies grow accordingly. As a result, the JPEG committee has recently released its new image coding standard, JPEG 2000, which will serve as a supplement for the original JPEG standard introduced in 1992. The JPEG 2000 standard provides a set of features that are of importance to many high-end and emerging applications by taking advantage of new technologies. In the new image coding standard the JPEG committee has adopted an implementation of an entirely new way of compressing images. Rather than incrementally improving the original standard using the discrete cosine transform (DCT), the new coding standard is based on the state-of-the-art wavelet transform. It addresses areas where current standards fail to produce the best quality or performance and provides capabilities to markets that currently do not use compression.

The wavelet transform is based on the time-frequency domain understanding of signal analysis. This is advantageous compared to current transforms where only one of the domains is available at any given time. Having both domains represented at a time is preferable to applications that strive to achieve signal compression since all redundancy of a signal cannot be detected in one domain alone. For signals of a non-stationary nature, such as transient impulses, it is often beneficial to be able to acquire a correlation between the time and frequency domains of the signal. Afterall, when it comes to practical purposes a lot of the naturally occurring signals are non-stationary. Many current transforms suffer from shortcomings on this topic.

The wavelet transform in JPEG 2000 allows exceptional localisation in both the time and the frequency domain. Without going in further detail on the wavelet transform, it is worthwhile mentioning that it is a mechanism used to decompose a signal into its set of basis functions (wavelets), analogous to the use of sines and cosines in the transformation used in conventional JPEG to represent other functions. A wavelet is a "small wave", which has its energy concentrated in time to give a tool for the analysis of transient, non-stationary, or time-varying phenomena [Valens, 1999]. It has the oscillating wavelike characteristic but also the ability to allow simultaneous time and frequency analysis.

One of the most important properties of the JPEG 2000 is concerned with performance. The *superior low bit-rate performance* property states that the performance at low bit rates should be achieved without sacrificing performance on the rest of the rate distortion spectrum [Skodras et. al, 2001]. The offered performance should be superior to the current standard at low bit rates. For instance, 0.25 bpp should be sufficient for highly detailed gray-scale images. This is an important feature for the standard when concerning network image transmission.
By comparing the different image compression algorithms JPEG and JPEG 2000, with the aspect, in this context, being the relationship between the rate (bits per pixel - bpp) and the distortion (Peak SNR - PSNR), JPEG 2000's superior compression capabilities become obvious. [Chai and Bouzerdoum, 2001] examine the coding performance as the JPEG 2000 encoder is put to the test against the original JPEG encoder. From this paper it can clearly be seen that the JPEG 2000 encoder outperformes its predecessor at every tested bitrate. The same test image is applied to both JPEG and JPEG 2000 encoders and coded at various bit rates, and the PSNR values, which quantify image quality, of the corresponding reconstructed images were obtained. Both bit-rate and PSNR is plotted to produce the rate-distortion curves in the figure on the next page. According to the results, JPEG 2000 proves a significantly better signal-to-noise ratio at every tested bit-rate. Furthermore, the PSNR value of JPEG encoded images decline more rapidly than the value of JPEG 2000 encoded images as the bit-rate approaches zero.

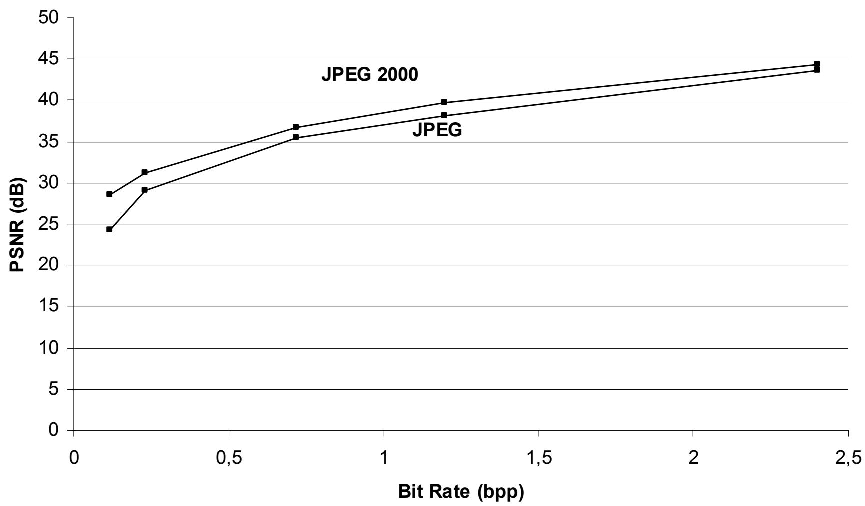

Figure 23 – Rate distortion curves for JPEG 2000 and classical JPEG

In addition, a subjective evaluation of reconstructed images processed with JPEG 2000 confirms that differences in image quality in general are more apparent at the lower end of the bit rate spectrum. Blocking artifacts appear in JPEG images due to the high level of quantization. This is a major inherent drawback of block based DCT coding scheme used by the previous JPEG standard. In contrast, the JPEG 2000 images experience a blurring effect at low bit rates. However, this does not depreciate the results of the wavelet based compression standard, as this effect is perceived less annoying by most viewers.

The superior performance of the JPEG 2000 is, however, at the expense of increased computational complexity of the algorithm. Furthermore, in the design of most practical systems and services, like mobile multimedia, complexity is often an important concern. Therefore, in the context of image processing in mobile terminals, JPEG 2000 is most likely to result in a tradeoff between computational complexity and reconstruction quality of images.

Apart from the expected increase of computational complexity the introduction of JPEG 2000 involves, an implementation of the JPEG 2000 as the compression scheme in this thesis would be of great interest. According to the description of the codec [Huh, 2001], the algorithm performs component transformation if the input image has multiple components (as of an RGB image). This component transformation is a procedure that provides decorrelation among image components and hence improves image compression efficiency. Based on this, JPEG 2000 should be capable of having Bayer RGB as input. This would make JPEG 2000 applicable as a compression scheme in the context of this thesis. Moreover, having JPEG 2000 applied as the compression scheme instead of conventional JPEG in the simplified image processing chain proposed in this thesis, not only computational savings will introduced, additional improvements of image reconstruction quality are expected. This would be an interesting topic to investiagate.

As a concluding remark, keeping in mind the continuing emerge of applications with increasing needs for compression techniques, the JPEG 2000, with the state-of-the-art wavelet technology has been developed to better address compression capability and quality. In addition, the JPEG 2000's functionalities and features such as lossy and lossless compression, spatial and SNR scalability, ROI (Region of Interest) encoding and good error resilience make the image compression standard a promising framework for future applications.

9 Conclusion

This thesis presents and investigates a simplified image processing chain for modern mobile terminals. It focuses mainly on the reconstructed quality of images processed by the simplified image processing chain. The work has proved that the suggested chain can be implemented two different ways: Bayer RGB and composite Bayer RGB as input to the compression scheme, both resulting in computational savings compared to conventional image processing.

Both implementations of the simplified image processing chain are simulated in Matlab. The quality of reconstructed images are evaluated according to suitable measures like SNR, compression ratio and perceived visual quality.

Bayer RGB performs as expected and according to literature reviews. It does not perform well for images containing large monochrome areas. Monochrome motifs are reconstructed with an evident raster-like pattern. On the other hand, Bayer RGB shows a very good reconstruction quality when it comes to greyscale images. Considering such images, classical JPEG is outperformed by Bayer RGB. Furthermore, taken into account the computational savings of the simplified image processing chain, Bayer RGB also performs well for images containing detailed motifs, with only small reductions on compression ratio compared to classical JPEG.

Composite Bayer RGB also proves consistency between theory and testing. Composite Bayer RGB proves good reconstruction capabilities of images containing monochrome motifs. The most advantageous feature of composite Bayer RGB is that it can be treated like JPEG and hence not introduce necessary changes to the existing compression scheme. When it comes to high-frequency details composite Bayer RGB tends to result in poorer reconstruction quality than both Bayer RGB and classical JPEG. In addition, the fact that composite Bayer RGB does not perform a symmetric image processing results in the appearance of excessive horizontal errors compared to the vertical errors.

Irrespective of the properties of the image to process, test results have proved that both the implementations of the simplified image processing chain reveal overall better results for SNR compared to classical JPEG when no compression is applied. Furthermore, both implementations are capable of better color reproduction than JPEG.

The perspective of this work has been to investigate the effects on reconstruction quality when reducing on the image processing operations a mobile terminal has to perform in order to process an image. The quantization matrices used by Bayer RGB and composite Bayer RGB in this thesis are based on existing tables adapted to JPEG. Further research on this area should therefore be to improve the tables and make them more suitable to Bayer RGB and composite Bayer RGB.

Another aspect of future work in the context of the improved image processing chain should be the implementation of the new image coding standard, JPEG 2000 featuring the wavelet transform. The effects on image processing performance should be investigation and compared with the existing JPEG standard in terms of computational complexity and reconstruction quality.

Furthermore, a full-color sensor with a new image capturing technology has recently been introduced. The effects this sensor, capable of capturing true RGB in each image pixel, has on image processing complexity needs to be investigated.

10 Abbreviations

AC	Alternating Current
CCD	Charged Coupled Device
CFA	Color Filter Array
CIE	Commision Internationale de L'Éclairage
CMOS	Complementary Metal Oxide Semiconductor
CRT	Cathode Ray Tube
CSF	Contrast Sensitivity Function
CZP	Circular Zone Plate
DC	Direct Current
DCT	Discrete Cosine Transform
GPRS	General Packet Ratio Services
GSM	Global System for Mobile communacation
IDCT	Inverse Discrete Cosine Transform
JPEG	Joint Photographic Experts Group
LMSE	Local Mean Squared Error
MAE	Mean Absolute Error
MCU	Minimum Coded Unit
MSE	Mean Squared Error
MMS	Multimedia Message Services
PSNR	Peak Signal to Noise Ratio
RGB	Red, Green and Blue
ROI	Region Of Interest
SNR	Signal to Noise Ratio
UMTS	Universal Mobile Telecommunications System
2D DCT	2 Dimensional Discrete Cosine Transform

11 References

[Aas]	Aas, O.,T., "DAT2430 Multimedieteknikk", http://fag.grm.hia.no/dat2430/
[Chai and Bouzerdoum]	Chai, D. and Bouzerdoum, A., "JPEG 2000 image compression: an overview"
[Chen and Pratt, 1984]	Chen, W.H., Pratt, W.K., "Scene adaptive coder", IEEE Transactions on Communications COM-32: 225-232
[Gibson et.al, 1998]	Gibson, J.D., Berger, T., Lookabaugh, T., Lindbergh, D., Baker, R.L., "Digital compression for multimedia, principles and standards", Morgan Kaufmann publishers, 1998
[Huh, 2001]	Huh, Y., "Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing"
[Keys, 1981]	Keys, R.G., "Cubic convolution interpolation for digital image processing", IEEE Transactions on aqoustics, speech and signal processing, AASP-29, No 6, December 1981, p. 1153-1160
[Mikkelsen, 2001]	Mikkelsen, T., "Camera functionality in modern mobile terminals", Agder University College, May 2001
[Polikar, 2001]	Polikar, R., "The Wavelet Tutorial – Fundamental Concepts & An Overview of the Wavelet Theory", 2001. Web: "http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html"
[Poynton, 1997]	Poynton, C., "A guided tour of color space", New foundations of video technology, February 1995, p. 167-180
[Poynton, 1999]	Poynton, C., "Frequently asked questions about gamma", http://www.inforamp.net/~poynton, December 1999
[Pratt, 1991]	Pratt, W.K., "Digital image processing, second edition", John Wiley & Sons 1991, ISBN 0471857661
[Sakamoto et.al., 1998]	Sakamoto, T., Nakanishi, C., Hase, T., "Software pixel interpolation for digital still cameras suitable for a 32-bit MCU", IEEE Transactions on consumer electronics, VOl. 44, No. 4, November 1998, p. 1342-1352
[Skodras et.al, 2001]	Skodras, A., Christopoulos, C. and Ebrahimi, T., "The JPEG2000 Still Image Compression Standard", September 2001.
[Valens, 1999]	Valens, C., "A really friendly guide to wavelets"

[Wallace, 1992]	Wallace, Gregory K. "The JPEG still image compression standard"; Wallace IEEE Transactions on consumer electronics, vol.38, February 1992
[Woolley et.al., 2001]	Woolley, S.I., Oh, J., Arvantis, T.,N., Townend, J.N., "A multistage perceptual quality assessment for compressed digital angiogram images", IEEE Transactions on medical imaging, vol.20, no.12, December 2001