TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 797 • Publication 797

Andriy Bazhyna

Image Compression in Digital Cameras

Tampere 2009

Tampereen teknillinen yliopisto. Julkaisu 797
Tampere University of Technology. Publication 797

Andriy Bazhyna

Image Compression in Digital Cameras

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB224, at Tampere University of Technology, on the 20th of March 2009, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2009

ISBN 978-952-15-2127-0 (printed)
ISBN 978-952-15-2150-8 (PDF)
ISSN 1459-2045

Abstract

The usage of digital cameras is continually increasing. The digital camera applications range from entertainment to science. Due to easiness of use, innumerable digital images are produced.

A digital camera is a complex optoelectronic device capable of performing image capturing and reconstruction. Light is transformed to electric form by silicon sensors. The quantized electrical signals acquired by the sensor are called "raw" data. The captured raw data are different from the original scene sensation. Sophisticated reconstruction procedures are used to produce final images from the raw data. Due to limited resources, it is not always possible to implement the best algorithms inside the camera. To overcome this limitation, storing original raw data and subsequent reconstruction on powerful devices are used.

In uncompressed form, both raw and image data, occupy an unreasonably large space. However, both raw and image data have a significant amount of statistical and visual redundancy. Consequently, the used storage space can be efficiently reduced by compression.

In this thesis, image data compression for digital cameras is studied. The problem is divided into two subcategories: raw data compression and compression of reconstructed images.

Compression of raw data is a relatively new research area. In this thesis, we describe peculiarities of raw data and introduce a problem of raw data compression. Similarly to image compression, the methods for raw data compression are categorized into lossless, near-lossless and lossy. We describe the applicability of different compression techniques for raw data. Approaches to performance evaluation of compression methods are presented. In addition, we propose algorithms for lossy, near-lossless and lossless compression. All our algorithms take the peculiarities of raw data into account. Since raw data are always contaminated by noise, special attention is paid to compression of noisy data. We propose a lossy compression method, where the compression errors can be masked by noise present in raw data.

Image compression has been a field of intensive investigation for several decades. In this thesis we argue for the usage of DCT block-based algorithms for image compression in digital cameras. An efficient method for compression of the quantized DCT coefficients is proposed. In addition to high coding efficiency, it allows other services, e.g. progressive coding, spatial and complexity scalability, ROI, etc. The image coder with the proposed encoding method demonstrates compression efficiency similar to JPEG2000. Furthermore, the utilizations of our encoding technique for the additional lossless compression of JPEG images and for the compression of 3D DCT coefficients are proposed.

An efficient method of prediction and compression of sign values of DCT coefficients in block-based image compression is also proposed in this thesis.

Preface

The work presented in this thesis has been carried out in the Department of Signal processing at Tampere University of Technology during the years 2004-2008.

First and foremost, I wish to express my deepest gratitude to my supervisor, Prof. Karen Egiazarian who is leading Transform and Spectral Techniques research group. He has provided me with possibility to work in such a nice place and always supported me with excellent professional guidance and patience throughout the course of this work. I am also very honored to work in one group with recognized experts Dr. Atanas Gotchev, Dr. Alessandro Foi and Prof. Vladimir Katkovnik, who never refuses to share their knowledge and directing my work.

My grateful acknowledgments go to Dr. Rusen Oktem and Prof. Alessandro Neri for reviewing the manuscript and providing constructive comments.

It was a pleasure to me to work closely with Prof. Vladimir Lukin, Dr. Mykola Ponomarenko, Dr. Alexander Totskiy from Kharkov National Aerospace University, Ukraine. I highly appreciate their comments and suggestion that helped me to improve this thesis. I am extend my gratitude for them and Prof. Sanjit K. Mitra from University of California, USA, to be the co-authors of some of my scientific publications.

I express sincere thanks to the all administrative and teaching staff of the Department of Signal Processing. Especially, I would like to mention Prof. Jaakko Astola and Prof. Moncef Gabbouj for creating such a nice research environment. Furthermore, I am warmly thank the secretaries: Virve Larmila, Pirkko Ruotsalainen, Ulla Siltaloppi and Elina Orava for their support in many practical matters.

Many thanks to all colleagues and friends in Department of Signal Processing for the possibility of discussion and creation of nice working atmosphere. Especially, I would like to mention Dmitriy Rusanovskyy, Susanna Minasyan, German Gomez Herrero who have shared an office with me, and Ekaterina Pogosova, Dmitriy Paliy, Atanas Boev, Evgeny Krestyannikov and Robert Bregovic for their friendship.

I would like to thank Tampere Graduate School in Information Science and Engineering (TISE) coordinated by Dr. Pertti Koivisto for providing funds for traveling and sharing knowledge.

The last but not least gratitude I express to my family, mother Nina and father Vladimir, and to my brother Artem. My warmest thanks go to my wife Marina and my daughter Anna for their love, unconditional support, and understanding.

Tampere, 25 January 2009

Andriy Bazhyna

Contents

List of Publications

This thesis consists of an introductory part and collection of papers. It contains some unpublished material, but is mainly based on the following publications. In the text, these publications are referred to as [P1]-[P8].

[P1] A. Bazhyna, A. Gotchev, K. Egiazarian "Lossless compression of Bayer pattern color filter arrays". Proc. of SPIE-IS&T Electronic Imaging 2005, Algorithms and Systems IV, vol. 5672, pp. 378-387, San Jose, CA, USA. January 2005.

[P2] A. Bazhyna, A. Gotchev, K. Egiazarian "Near lossless compression algorithm for Bayer pattern color filter arrays". Proc. of SPIE-IS&T Electronic Imaging 2005, Digital Photography, vol. 5678, pp. 198-209, San Jose, CA, USA. January 2005.

[P3] A. Bazhyna, K. Egiazarian, S. Mitra, C. Koh "A Lossy Compression Algorithm for Bayer Pattern Color Filter Array Data". Proc. of International Symposium on Signals, Circuits and Systems, (ISSCS) 2007, vol. 2, pp. 1-4, Iasi, Romania. July 2007.

[P4] V. Lukin, N. Ponomarenko, A. Bazhyna, K. Egiazarian "Compression of noisy Bayer pattern color filter array images". Proc. of SPIE-IS&T Electronic Imaging 2007, Computational Imaging V, vol. 6498, pp. 64980K, San Jose, CA, USA. January 2007.

[P5] A. Bazhyna, K. Egiazarian "Lossless and Near Lossless Compression of Real Color Filter Array Data". *IEEE Trans. on Electronic Imaging*, vol. 54, iss, 4, pp. 1492-1500, Nov. 2008.

[P6] N. Ponomarenko, A. Bazhyna, K. Egiazarian "Prediction of signs of DCT coefficients in block-based lossy image compression". Proc. of SPIE-IS&T Electronic Imaging 2007, Algorithms and Systems V, vol. 6497, pp. 64970L, San Jose, CA, USA. January 2007.

[P7] A. Bazhyna, N. Ponomarenko, K. Egiazarian ,V. Lukin "Efficient scalable DCT block-based image coder with compression of signs of DCT coefficients", *Journal of Telecommunications and Radio Engineering*, vol. 67, iss. 5, pp. 391-412. 2008

[P8] A. Bazhyna, N. Ponomarenko, K. Egiazarian "Efficient bit-planes based method for compression of 3D-DCT coefficients". Proc. of 26th Picture Coding Symposium (PCS) 2007, pp. 4, Lisbon Portugal. November 2007.

List of Abbreviations

2D, 3D	Two, Three Dimensional
AA	Anti-Aliasing
ASIC	Application-Specific Integrated Circuit
BCFA	Bayer pattern Color Filter Array
bpp	bits per pixel
CFA	Color Filter Array
CR	Compression Ratio
DCT	Discrete Cosine Transform
DRAM	Dynamic Random Access Memory
(D)SLR	(Digital) Single Lens Reflex camera
DSP	Digital Signal Processing
DWT	Discrete Wavelet Transform
ET	Exposure Time
EXIF	Exchangeable image file format
IPP	Image Processing Pipeline
IR	Infrared
JPEG	Joint Photographic Experts Group
OOP	Optimal Operation Point
PSF	Point-Spread Function
(P)SNR	(Peak) Signal to Noise Ratio
RGB	Red, Green, Blue color space
ROI	Region of Interest

ROM	Read Only Memory
SM	Significance Map
UV	Ultraviolet
WB	White Balance

Chapter 1

Introduction

1.1 Motivations and Background

During the past decade digital cameras have rapidly entered into the everyday live. Since the year 2003 more digital cameras have been sold in the US and Europe (year 2004, worldwide) [43] than traditional film cameras. Through the years 2000-2005, the share of the camera market occupied by digital cameras grew from 20 to 80 percent [46]. The growth is still continuing, also in absolute values. Digital cameras are not only separate products but also modules embedded in different consumers and industrial devices (mobile phones, PDA, notebooks, remote sensors, toys, PC peripherals, etc.). Astronomy, medicine and other sciences are also important areas of digital camera applications.

Since digital cameras became affordable, the number of pictures taken and their resolution have been growing very fast. Nowadays the typical resolution ranges from 3-7 megapixels for mobile phone cameras to almost 50 megapixels for the most advanced commercial middle-format Hasselblad H3DII-50 camera [68].

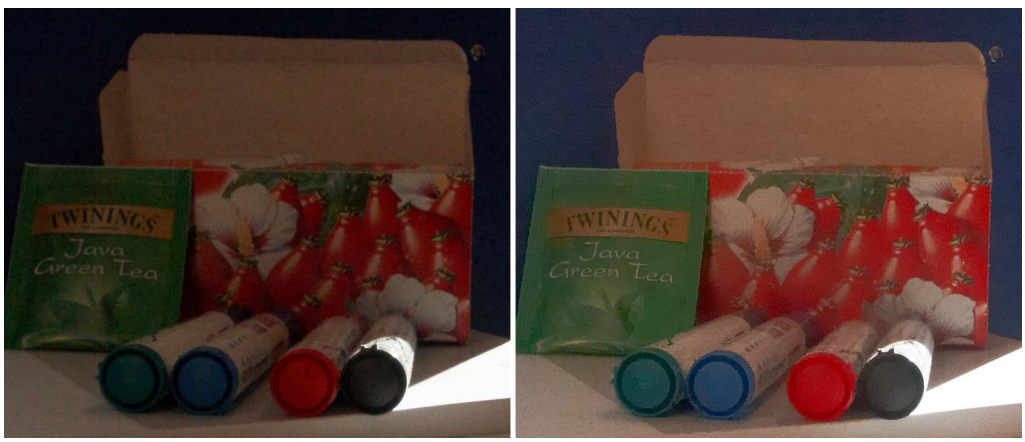

Figure 1.1: Images reconstructed from the same Canon "Powershot G2" raw data by the Canon "Raw Image Converter" released in 2002 (on the left) and in 2007 (on the right).

Most of the consumer cameras are able to reproduce full true color images very shortly after a picture is taken. The initial "raw" data captured with the camera is, however, very different from what we see at the output (the "negatives" are a good example from film photography). Digital cameras do all the processing that is necessary to develop raw data into a full color image (like the famous Polaroid cameras do). The optimal image reconstruction using sophisticated algorithms is still a challenging task even for a powerful PC. Therefore, the quality of images reconstructed by the camera or by an embedded camera module may be compromised due to the hardware constrains. Thus, storing of raw data directly and posterior reconstruction on PC under full user control may lead to significant improvement of image quality. Within years, as more sophisticated algorithms become available, a better image can be obtained even from the same raw data (Figure 1.1).

In uncompressed form raw data occupy an unreasonably large space. However, there is a significant amount of redundancy and the used storage space can be efficiently reduced by compression. Therefore, the development of compression algorithms for raw data is an emerging task.

The reconstructed images in most digital cameras are stored in EXIF format [133]. The EXIF is based on the JPEG [140] image compression standard. In addition to image data compressed by JPEG, the EXIF file stores metadata describing the camera and picture-taking conditions.

The JPEG has been widely used image compression standard for more than 15 years. It is based on Discrete Cosine Transform (DCT) [7] with block size 8x8. The JPEG offered acceptable compression gain for a long time. However, new applications, like transmission over broadband (Internet) and narrowband (wireless networks) channels, viewing images on devices with different displaying and computational capabilities impose new required features on the image coding standard. These include progressive coding, spatial, quality and complexity scalability, Region of Interest (ROI), etc.

Recently, image compression using Discrete Wavelet Transform (DWT)[121] gained special attention due to the good decorrelating and localization properties of DWT. Investigations in this direction have resulted in the latest image compression standard JPEG2000[175]. It is widely agreed that multiresolution and, in particular, DWT-based methods are able to provide better quality than DCT-based methods for high compression ratios (CRs) [153, 195]. Together with superior image quality, JPEG2000 has additional capabilities like progressive coding, spatial scalability and ROI coding.

Very high CRs are rarely required in digital cameras. At the same time, the hardware realization of JPEG2000 is more resource-consuming compared to JPEG. This is mainly because DWT, in order to be efficient, should be applied to large tiles of an image or even to the whole image. As a result, JPEG2000 is more demanding of memory and computational resources in contrast to JPEG, which processes the image with relatively small blocks. That is why, although JPEG2000 was standardized more than six year ago, it is not widely used. Moreover, nowadays there are no digital still image cameras that are able to shoot images in JPEG2000 format. However, the research in this direction is still in progress.

Thus, for digital cameras, the development of improved image compression methods that process images in a block-based manner is an important task.

1.2 Objectives and Scope of the Research

The research performed in this thesis is intended to contribute to a better understanding of the role of compression in digital cameras. The ultimate goals are to develop new compression algorithms for digital cameras and/or improve the performance of existing image compression algorithms.

Compression in digital cameras is required for two types of data: raw data and full color images. These are rather different data, although, they may share similar principles for compression. Some methods previously developed for full color image compression can be used for raw data after some necessary modifications, or with certain restrictions.

The research work of this thesis covers two areas (Figure 1.2):

- image capturing and compression of raw data,
- DCT block-based image compression.

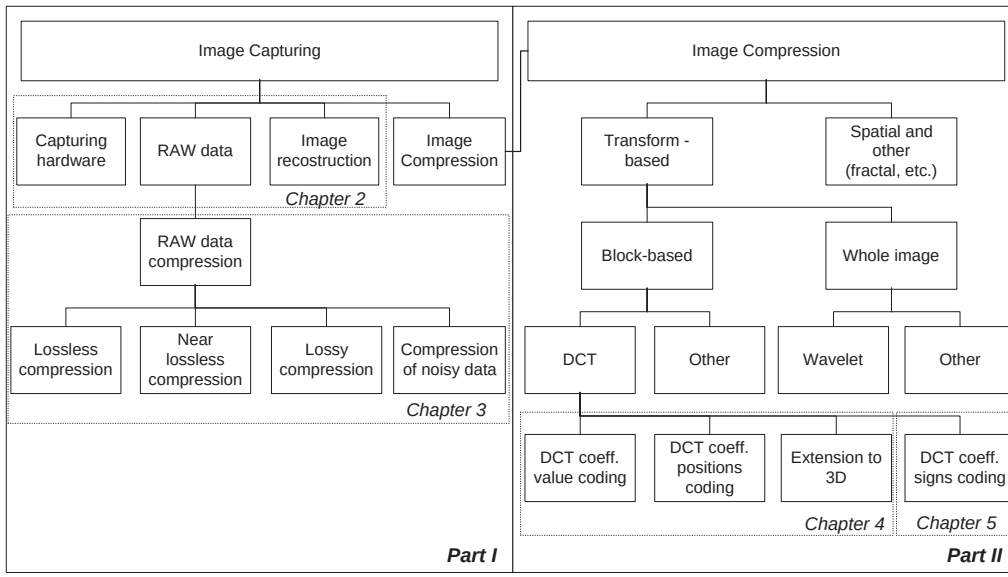

Figure 1.2: Scope of the research and thesis outline

At the start of our research, in 2004, there were only a few publications available in open literature [97, 181, 92] in the area of raw data compression. The benefits, drawbacks and potential problems of raw data compression were not clearly formulated. There was no unified framework or classification of the methods.

Similarly to image compression, methods for raw data compression can be classified into three groups: lossy, lossless and near-lossless. Employing lossy compression of raw data might be sufficient for conventional photography and for mobile device cameras. However, for medical, astronomical, professional photography and other high quality imaging applications the lossless and near-lossless compression might be required.

This thesis illustrates how the raw data is captured and transformed into full color images. A general framework for raw data compression is developed. Methods for

lossless, near-lossless and lossy compression of raw data are presented. The quality of images reconstructed from compressed raw data is investigated.

In contrast to raw data, compression of full color images is the area that attracted attention for several decades. Among the great variety of approaches, this thesis contributes to DCT block-based compression. A highly flexible and scalable method for the compression of quantized DCT coefficients was developed. The method is suitable for the compression of values and positions of DCT coefficients and provides high coding efficiency. The applications of this method for 2D and 3D DCT-based data compression are presented. An additional important application area is the further lossless recompression of a huge amount of existing JPEG images.

When DCT-based compression is employed, the signs of DCT coefficients are considered as random variables. Each coded sign of DCT coefficients usually occupies 1 bit of memory in compressed data. For modern compression methods, coded signs of DCT coefficients occupy up to 20-25% of a compressed bitstream. We developed an effective way of predicting the signs of DCT coefficients. It increases the performance of most DCT block-based methods for image and video compression.

1.3 Thesis Outline

This introductory chapter provides an overview of motivations for this work. The objectives and the scope of the research and the thesis outline are also introduced. The rest of the thesis is logically split into two parts, each consisting of two chapters (Figure 1.2).

The first part is devoted to image capturing technologies and compression of raw data. Capturing of visual information with digital cameras and technologies used for the reconstruction of full color images are presented in Chapter 2. Some parts of publications [P1], [P2] and [P3] relate to this chapter. Chapter 3 focuses on raw image compression. Methods for lossless, near-lossless and lossy compression are presented. The work presented in publications [P1]-[P5] is an essential part of this chapter.

The second part focuses on block-based DCT image compression. In Chapter 4, a bitplane based method for efficient compression of quantized DCT coefficients is presented. Its applications for image/video compression using 2D and 3D DCT blocks are given. Publications [P7], [P8] contribute to this chapter. Chapter 5 is dedicated to the signs coding for the block-based DCT image compression. This chapter is mainly based on publication [P6]. Chapter 6 draws concluding remarks.

Part I

Image capturing and compression of raw data

Chapter 2

Image Acquisition with Digital Camera

A digital camera is a modern complex device capable of performing image/video capturing and reconstruction close to realtime. For this, it uses the latest achievements in optics, hardware, microelectronics, signal and image processing.

In this chapter, an overview of image capturing with digital cameras is given. It is a review of the technologies used to produce digital images. First, the camera hardware components and their interaction are presented. Then, technologies for light to electrical signal conversion and color separation methods for color imaging are described. Different Color Filter Array (CFA) patterns are illustrated. The image processing pipeline (IPP) used to transform raw data captured by the sensor into full-color images is presented. The operations needed for image reconstruction and their up-to-date solutions are reviewed. Finally, image reconstruction using an external computer approach is presented and its advantages are formulated.

2.1 Camera Hardware

The block diagram of the hardware components of a typical digital photo/video camera is presented in Figure 2.1 (Chapter 12 in [163], [149, 73]).

The camera lens focuses incoming light from the capturing scene into a plane where the capturing sensor is located. Most modern cameras are equipped with a variable focal length (zoom) lens. Fix-focal lenses are used for low cost and simple camera modules. Additionally, fix-focal lenses are used in specialized applications, for the highest image quality. The lenses are controlled using zoom and focus motors. The embedded or external flash module can be used to assist image capturing in low-light conditions.

The aperture is the opening that determines the amount of the incoming light reaching the sensor plane. The aperture is controlled by a special element, the diaphragm, placed in the optical path. The shutter is a device that allows light to pass to the sensor for a determined period of time.

The infrared (IR) and ultraviolet (UV) are nonvisible components of light. Sensors, however, are sensitive to these components. Therefore, IR and UV are filtered-out by

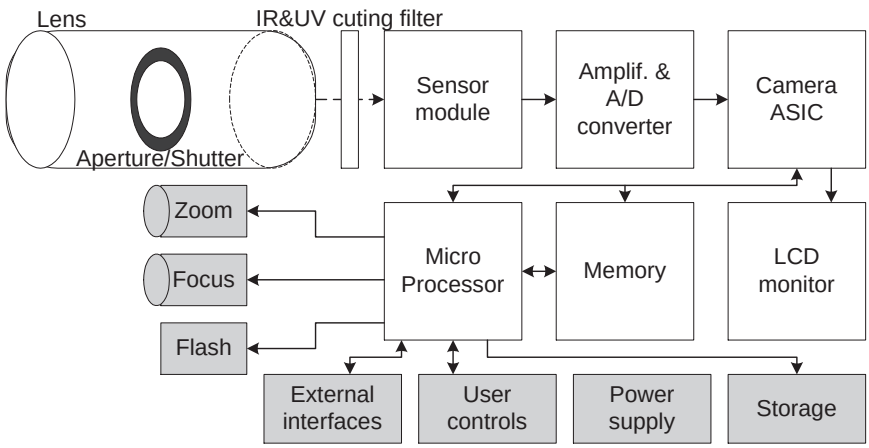

Figure 2.1: Block diagram of the hardware components of a digital camera.

the corresponding optical filters.

The analog signal from the sensor is amplified and converted into digital form using an analog-to-digital (A/D) converter [79]. The resolution of modern A/D converters for digital cameras ranges from 8 to 16 bits.

The digital data are processed by an application-specific integrated circuit (ASIC), which implements IPP. Alternatively, these tasks also can be performed by a high-performance microprocessor with an embedded digital signal processor (DSP). The microprocessor and the ASIC are often incorporated into a single integration circuit. The microprocessor is also responsible for controlling zoom and focus motors, flash, camera controls, interaction with user, power management and interfaces (e.g. PC, TV, and so on).

The memory used by a digital camera is composed of two types. The fast dynamic random access memory (DRAM) buffer for several unprocessed images allows series shooting. The programmed instructions for the microprocessor (firmware) are stored in erasable programmed read only memory (EPROM).

The reconstructed images are stored in memory cards (SecureDigital (SD), CompactFlash (CF), etc.), hard drive disks or other embedded or removable media.

The images and camera operational parameters can be monitored on LCD or TFT display. In many cameras data captured by a sensor can be viewed close to real-time.

2.2 Sensor Module

The sensor module is composed of one or several light-sensitive sensors and optical elements. Additional optical elements and/or several sensors are required for color imaging.

2.2.1 Light Sensors

For the needs of digital imaging, light is converted into an electrical charge using the photoelectric effect [32, 160]. The photon can move the electron into a conducted band

if the energy of the photon (E_{ph}) is greater than or equal to the band gap energy of the material (E_g). The photon energy is $E_{ph} = h\upsilon = hc/\lambda$, where h is a Planck's constant, υ is the frequency, λ is the wavelength and c is the speed of light.

The special aspect of the photoelectric effect is a critical wavelength $\lambda_c = hc/E_g$. Only photon with wavelength $\lambda < \lambda_c$ have sufficient energy to generate an electron-hole pair. For the silicon (main material for light sensors) $E_g = 1.12eV$ and $\lambda_c = 1.11\mu m$ lies in the IR region.

In practice, not all photons that reach the silicon surface with $E_{ph} > E_g$ are capable of generating electron-hole pairs. Quantum efficiency (η) is one of the most important parameters of the sensor, which reflects the percentage of photons P converted to electrons E. Thus,

$$E = \eta P. \tag{2.1}$$

The reasons for which η being below 100% are absorption in optical insensitive structures on top of a sensor, reflection from the silicon, etc. All these vary significantly depending on manufacturing technology as well as on sensor design. Additionally, there is a complicated dependency $\eta(\lambda)$, which is called the sensor spectral response. The modern sensors for digital imaging are designed so that they have η close to 90-100% for middle of visible wavelength 400-700nm [31, 32, 119].

The number of photons that strike the silicon surface while the shutter is open is discrete. This process is well approximated by a Poisson distribution [89, 151]. This distribution is associated with a number of random independent events appearing within a fixed time interval. According to Poisson statistics, if N is an average number of events expected to occur in a certain time interval, the standard deviation is $\sigma_p = \sqrt{N}$.

These fluctuations of number of photons (and thus electrons) are considered as a noise, called photon-counting [18] or "shot" [119] noise. Therefore, image data, registered by a sensor, are contaminated by a signal-dependent noise, which originates from the capturing process itself. The signal-to-noise (SNR) ratio is

$$SNR = P^2/\sigma_p^2 = P, \tag{2.2}$$

where P is the average number of photons collected. For a large P the SNR is large as well. However, it is not rare that modern sensors operate with only several tens of photons, or even less. In modern light sensors, the dominating source of noise is often the shot noise.

The photogenerated electrons (charge) are accumulated in two types of structures: photodiodes and photogates (capacitors) (see Figure 2.2) [177].

Photogates use metaloxidesemiconductor (MOS) capacitors to create a voltage-induced potential well to store photoelectrons. With photogates, practicably 100% of the sensor surface is photosensitive (high fill factor). Thus, more photons can be captured, transformed to electrons and stored. The tradeoff is lower sensitivity due to the gate on top of the sensor. Photodiodes accumulate photogenerated electrons in the $p - n$ junction. Although photodiodes provide higher sensitivity, they trade it for a lower fill factor. Both technologies are in use and their characteristics are constantly being improved.

The spatial sampling of the image projection formed by a lens is performed by a 2D array of photoelements. Each photosite in the sensor corresponds to one pixel (picture element) - the smallest structural element of the image.

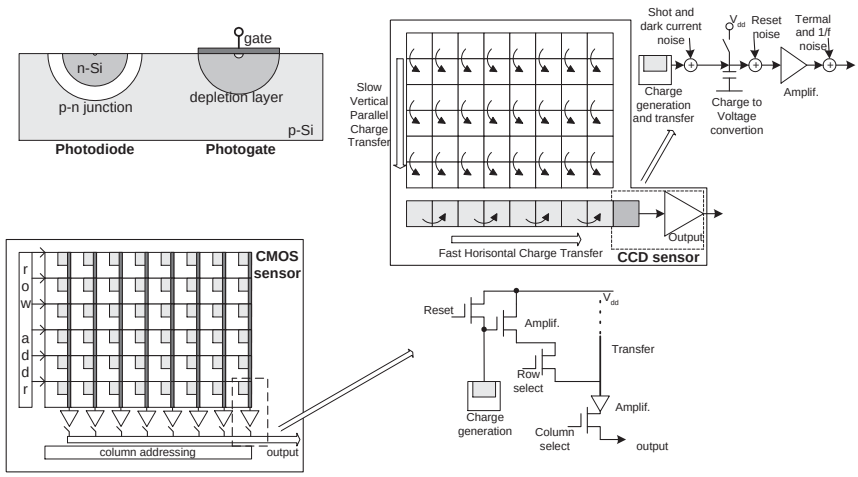

Figure 2.2: Sensor elements: photodiode and photogate. Two sensor technologies: CCD and CMOS.

Once a charge is accumulated while the shutter is open, it needs to be measured and transferred to the output. For measuring, the charge is converted into voltage using a capacitor. There are two sensor technologies that differ in the way the charge is converted into a voltage and how it is transferred from the sensor. These are the charge-coupled device (CCD) and the complementary metaloxidesemiconductor (CMOS) [119].

A CCD (Figure 2.2) is an analog shift register, enabling electric charges, as analog signals, to be transported through successive stages and controlled by a clock signal. The charge - to - voltage conversion is performed in one or a small number of convertors at the sensor output. This allows high uniformity of the output signals.

The main difference of the CMOS (Figure 2.2) is integration inside the pixel, the transistors for charge - to - voltage conversion, buffering and row/column addressing (3T - three transistors CMOS).

The CMOS was invented (DALSA, Dr. Savvas Chamberlain) at nearly the same time as the CCD (Bell Labs, Willard Boyle) in the early 1970s. However, due to limitation of semiconductor lithography processes their performance was very poor. Nowadays, both technologies are comparable and with their own strengths and weaknesses. Some of these are listed below [29, 78, 106].

- *Fill factor.* High (up to 100%) for CCD. Moderate for CMOS, as part of the sensor is covered by transistors and readout circuits.

- *Dynamic range.* High for CCD. Moderate for CMOS due to smaller fill factor.

- *Power consumption.* Much smaller for CMOS compared to CCD due to using low-power, low-voltage in-pixel amplifiers. CMOS usually operates with a single voltage. CCD require several high-voltage and powerful sources for charge transfer.

- *Uniformity.* High for CCD, because it uses one or several amplifiers. Moderate for CMOS, because of the large number of amplifiers in every pixel.

- *Complexity.* For CCD, the design and manufacturing of the sensor is cheap. The rest of the system can be changed without touching the sensor. The overall system, however, is more complicated. CMOS sensors are more complicated in design and fabrication. However, CMOS sensors are fabricated using the same processes as for other digital and analog circuits. Thus, overall system complexity can become much smaller. The most challenging ability is to realize the whole camera on a single chip [55].

- *Electronic Shuttering.* This is the ability to start and stop exposure arbitrarily. It can be easily realized for CCD with little fill-factor compromise. For CMOS realization requires additional in-pixel transistors.

- *Speed.* Comparable for both technologies. However, CMOS has an advantage because the whole system can be placed into one crystal and have smaller distances, capacitance and delays.

- *Antiblooming* is the ability to drain overexposed areas without affecting the rest of the image. Not relevant for CMOS because of in-pixel charge - to - voltage conversion. Could be problematic for CCD.

- *Windowing.* CMOS provides extensive windowing capabilities due to memory-like addressing of every pixel. CCD has limited windowing ability.

Both technologies are continually improving their features. The CCDs are constantly decreasing power consumption, and increasing antiblooming and windowing functions. The 4/5T CMOS, on the other hand, provides shuttering and dark current noise reduction capabilities. The noise reduction is performed by double sampling of each pixel in dark and after exposure and subtracting the signals [177].

2.2.2 Color Separation

The information about color is described by the tristimulus values of the amounts of the three primary colors in a three-component additive color model. The three-component additive color model is motivated by the fact that the human eye has three types of receptors (cones) that are sensitive to short, middle, and long wavelengths [105, 183]. The three primary colors are Red (R), Green (G) and Blue (B). The CIE1931 [25, 169] is the first color space standard defining the CIE RGB color space with monochromatic primaries and the abstract CIE XYZ color space, which is used for conversion between color spaces.

Therefore, for color imaging, the sensor must be sensitive to three primary colors. Silicon-based sensors are achromatic in nature. They have no ability to determine information about color from the light. There are three technologies used to extract color information from the scene [163, 31].

- *Color Sequential* (Figure 2.3A). The color image is produced by three successive exposures while switching in optical filters. A three times longer exposure is required which is acceptable only for stationary scenes, or with a very strong light source (reasonably short exposure). Filter switching adds additional mechanical complexity. This technology is used in Digital Micromirror Device (DMD) projectors.

- *Three-Sensor* (Figure 2.3B). This method uses a beam splitter (dichroic prism) to separate light into color components and directs beams to three sensors. It provides great color fidelity, high spatial resolution and easy color image reconstruction. The disadvantages are: high cost (the sensors and optical elements are often the most expensive parts [4]), high assembly accuracy and relatively big size. This method is used in a small number of expensive specialized cameras and in high quality consumer video cameras.

- *CFA* (Figure 2.3C). In this method each individual photodetector is made sensitive to only part of the light spectrum. This is done by placing an optical color filter (mask) into each photodetector. This mask is placed in a special order, called the CFA pattern. Technologically, it is the simplest, smallest and cheapest method to date. The disadvantage is lower spatial resolution and additional processing for full color image reconstruction.

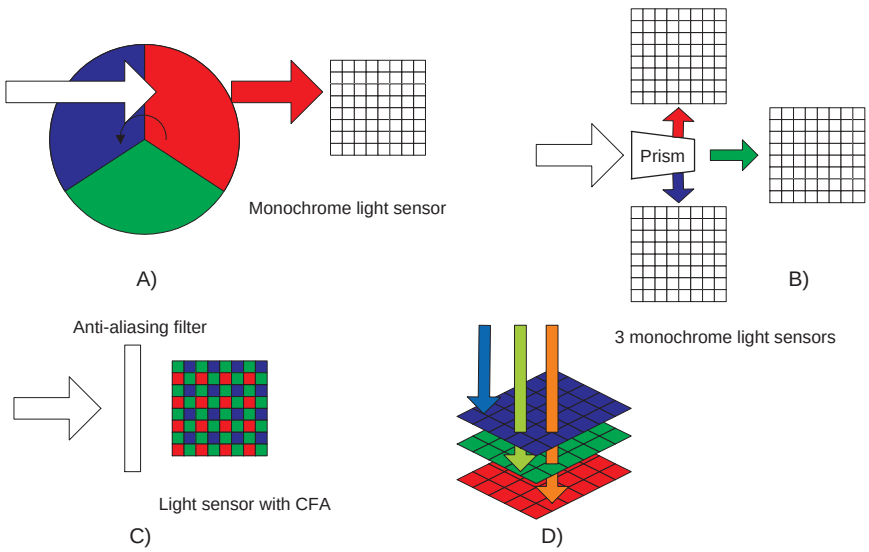

Figure 2.3: Color separation methods for color imaging: Color Sequentis(A), Multi-Sensor (B), CFA (C), Foveon® technology(D).

One more technology for color capturing is present on the market, called Foveon® (Figure 2.3D) proposed by the company of the same name [30]. The Foveon sensor uses a single 2D array of photoelements, each of which consists of three vertically stacked photodiodes. Each of the stacked photodiodes responds to different wavelengths of light. This is due to the fact that different wavelengths of light penetrate silicon to different depths. The signals from the three photodiodes are then processed in order to reconstruct the three primary colors [59].

The sensor provides excellent spatial resolution and color accuracy. No color artifacts are introduced by the reconstruction algorithm. The sensor is able to capture more photons than a sensor with CFA. However, reproducing primary color requires additional processing, which increases noise level, thus causing relatively poor low light selectivity.

There is also a certain technological problem. The most advanced Foveon X3® sensor has 13.1 megapixels overall, but is able to produce only a 4.7 megapixels non-interpolated full-color image. The Foveon sensor is used in only eight digital cameras and five are currently available on the market.

Overall, the CFA is the dominant technology nowadays. It is used in up to 99% of all digital still image and video cameras. The situation is not expected to change in the near future.

One issue related to CFA is aliasing [4]. This phenomenon occurs when trying to capture spatial frequency finer than the spacing between the pixels. It becomes apparent as low-frequency moire patterns in the high frequency region. Since, in CFA, colors are sub-sampled, additional actions should be taken to prevent aliasing. This problem is overcomed by placing an optical anti-aliasing (AA) filter in front of a sensor. The purpose of the AA filter is to exclude high frequency components by blurring. Thus, the digital images suffer from blurring because of the nature of capturing and, therefore, an increase of sharpness is required during the reconstruction stage.

Two types of AA filters are the most common ones: polarization-based [62] and phase delay [132, 131]. In the polarization filter, one or two pieces of birefringent material (quartz or calcite) are oriented to split incoming light into two or several outgoing beams, directing each beam to a different pixel. The drawback is the high price of optical quality birefringent material. In the phase-delay filters a near-random pattern is etched on a single slice of optical material. As a result, light coming from different depth of material suffers different phase delays. By adjusting the delays it is possible to suppress the higher spatial frequencies. This is a very inexpensive method, but, in some cases, it can act as a diffraction grating and completely destroy the image. Both technologies are in use.

2.3 Color Filter Arrays

The choice of CFA is a crucial point in digital camera design. The CFA pattern affects resolution (luminance and color), camera sensitivity, SNR performance, color reconstruction quality, image artifacts, and subsequent image processing steps.

There are two issues in CFA pattern design: which colors to use and how they should be placed [156, 50]. Mainly primary colors (R, G, B) or their complementary Cyan $(C = G + B)$, Magenta $(M = R + G)$ and Yellow $(Y = R + G)$ colors are used in practice. Sometimes, White (W) or colorless elements are added (Figure 2.4).

Primary and complementary color CFAs were compared in [139]. It was shown that a sensor with a CMY pattern (Figure 2.4B) provides about 6dB better luminance SNR at low-light levels. In contrast, for sufficient lighting conditions, the RGB pattern is about 2.6dB superior in luminance SNR. However, the main problem of CMY sensors is poor color reproduction, preventing their widespread use [129]. In an attempt to improve color accuracy, patterns that use combinations of primary, complementary and white colors have been proposed [128] (CYGW), (Figure 2.4C (CMY+G)) (see Chapter 12 in [163]). However, color reproduction accuracy provided by the use of primary colors was not achieved.

The pioneering CFA pattern was proposed by B. Bayer ((Figure 2.4A [16]) in 1976. It uses 50% of G, and 25% of R and B pixels. The author called the G photosensors as luminance-sensitive elements and the red and blue ones as chrominance-sensitive el-

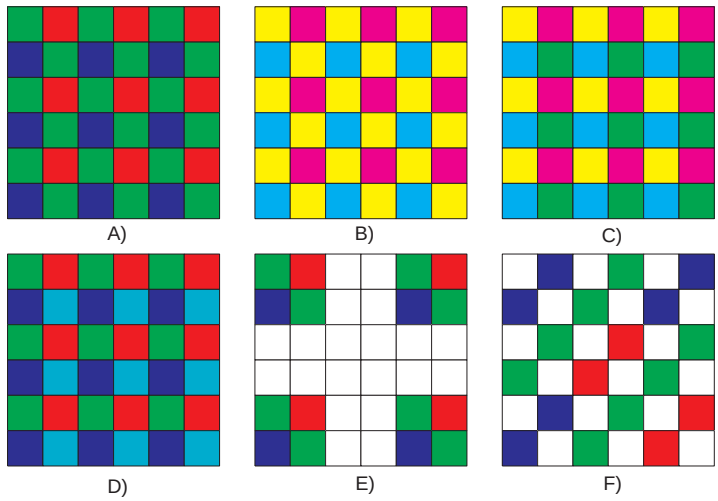

Figure 2.4: Examples of CFAs: Bayer pattern (A), CMY (B), CMY+G (C), RGB+emerald (D), pattern with transparent cites (E), new Kodak CFA (F).

ements. The reason is that the human Visual System (HVS) is most sensitive to the green part of the spectrum. The G pixels are placed in a check-border (quincunx) grid. The empty places are equally filled by R and B pixels.

Other proposed technologies use repeating coloring stripes of different sizes and direction [74, 50], checkboard patterns [110, 185], pseudo-random color [110, 156] and HVS-based [138] placing. These variations were motivated to improve some characteristics of the CFA pattern for some specific applications (e.g. interleaved video recording).

The latest advances in CFA design are to include colorless pixels (Figure 2.4E-F [115, 34, 116]). This is mainly done to improve low-light selectivity of small sensors with a large number of pixels. These sensors have tiny photocites and are capable of capturing a small number of photons during reasonably short exposure times. This results in a poor SNR of captured images. The patterns with achromatic cites have better light sensitivity at the expense of less accurate color reproduction and lower spatial resolution for colors.

Despite the great variation in CFA patterns, only few of them have been used in the mass market devices so far. They are mostly limited to those presented in Figure 2.4A-D. The dominant CFA pattern is the Bayer pattern (Figure 2.4A). Practically any digital camera nowadays uses Bayer pattern CFA (BCFA) for capturing color images. Multiple experiments were carried out in [110, 112] for performance evaluation of the different CFA patterns. They show high efficiency of the Bayer CFA pattern for color reproduction, for the purpose of digital image capturing.

2.4 Image Formation

2.4.1 Image Processing Pipeline

The raw data captured by a sensor need to be extensively processed before a full color image is obtained. For this it is passed through an IPP. The typical IPP of a digital camera with a CFA sensor [82, 203, 149, 186] is presented in Figure 2.5.

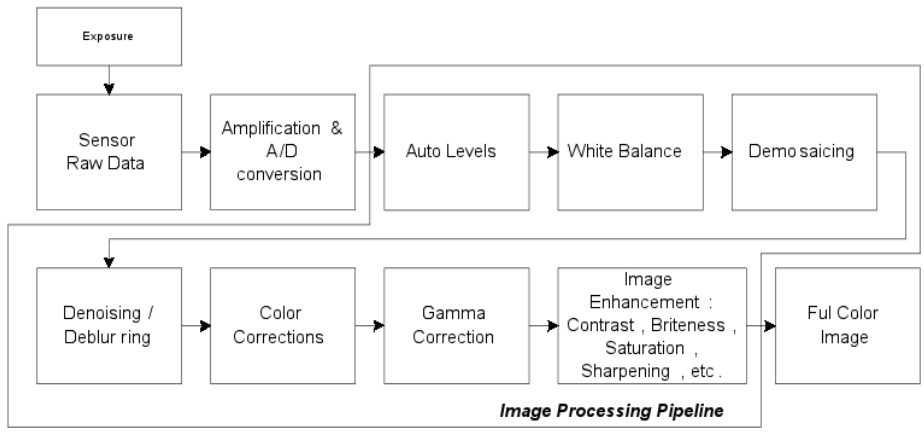

Figure 2.5: Typical IPP of a digital camera.

The following considerations are underlined in the IPP design [82]. White balance (WB) should be performed at an early stage because the later processing will rely on the correct ratio between the color channels. If WB adjustment is applied at later stages after nonlinear processing, the colors tend to become incorrect.

The later processing typically requires three-stimulus values of pixels. These are generated by interpolating from the raw data also applied at an early stage. Moreover, demosaicing often causes artifacts, and thus it should be performed before image enhancement stages.

Color correction, sharpening and deblurring also increase the noise. Hence, noise filtering should be performed before these operations.

Gamma correction (GC) and image enhancement are often nonlinear operations. Preferably, they should be applied at the later stages.

There are may be other algorithmic and technological considerations for a particular camera design, that modify order, add or remove additional blocks.

2.4.2 Exposure determination

The exposure is the total amount of light that falls on the sensor during the exposure time (ET). Most natural scenes contain a wide range of luminance values. The acceptable image is obtained only if the most important areas fall into the middle of the dynamic range of the sensor (limited by noise level and saturation values).

The exposure depends on the following factors:

-*Lens F-number or relative aperture.* The wider the opening, the more light coming to the sensor. There is a theoretical limit on the maximum F-number for glass lenses.

In practice, it is limited technologically. Moreover, at the widest aperture the lens parameters degrade.

-*ET*. Longer time provides more light, but may cause blurring due to camera and object movement.

-*Illumination*. Scene illumination level and scene reflectance can vary within wide range.

-*Gain parameter*. The more the signal from the matrix is magnified prior to A/D conversion, the shorter the ET required. This is a costless parameter, but the higher magnification increases the noisiness of the data.

In order to standardize the exposure calculation, methods for measuring overall camera sensitivity in terms of ISO speed were developed [76, 33]. The cameras selectivity depends on sensor quantum efficiency, the size of the individual photocite and a gain parameter. Different cameras use different combinations of these parameters to achieve the same ISO speed [26].

There are many methods for scene illumination level determination: average (the simplest, but not good for high and low illuminated areas), spot (allows correct illumination for a selected small region by sacrificing the rest of the image), and matrix (measures average luminance in relatively large (100-1000) number blocks covering the whole image).

2.4.3 Levels Adjustment

The signals after A/D conversion are presented with 8-14 bits per pixel (bpp) accuracy. The camera DSP processor is usually integer-valued and uses a wider range than the A/D converter. In order to avoid accumulating rounding errors during the processing stage the signal from the A/D converter should be normalized.

Additionally, due to exposure errors or scene illuminations, a signal may occupy only a part of the dynamic range of the sensor. Even with the right exposure the dynamic range of the scene may be narrower or wider than the dynamic range of the scene.

The dark level after A/D conversion does not necessarily correspond to the image dark level. The offset is caused by the dark current of the sensor and flare resulting from lens imperfection, etc. The level correction is typically done by histogram analysis and adjustment [82].

2.4.4 White Balance

The scene captured by the camera may be illuminated by different types of light sources. The amount of R, G and B components varies due to different spectrum of light sources. Thus, color characteristics of the scene depend on which light source it is illuminating.

The HVS is very adaptive and automatically corrects color sensations to reconstruct colors in the scene. In contrast, the camera sensor has a fixed spectral response resulting from the particular sensor design. Thus, the correct color reproduction for different types of light sources is problematic.

The first stage to correct color reproduction is to set the correct WB - relative relations of the R, G and B light that correspond to a particular light source.

This adjustment can be performed in the following ways (Chapter 12 [163]): a) using an optical correction filter to equalize sensor exposure levels for different color channels,

b) adjusting analog amplifier gain when the signal is read from the sensor, c) adjusting the digital codes of the captured image.

The first approach usually cannot be automated and requires additional user actions. The first and second approaches fail if different areas of the scene are illuminated by different sources of light (e.g. daylight from window and tungsten bulb in the room). Thus, most digital cameras use the third method.

Plenty of methods for automatic WB have been proposed so far [57, 11, 188]. Some rely on heuristic ideas, e.g., gray world assumption or Retinex theory [94]. However, such assumptions may not hold for a particular scene. Others use separate sensors directed to the light source to measure its R, G and B color relations [67]. This, however, fails for several light sources. Better results could be obtained by analyzing color relations in different segments of the image sensor data [107, 126]. One more approach is to calibrate the sensor characteristics and create correlation tables or gamut maps for different illuminants [51]. This method relies on the huge image database, and each picture being annotated with its illuminant conditions.

The challenging part of WB is to determine a point of the scene that has neutral color (colorless). Often the brightest point is used for this. However, sometimes the brightest point may have some color tonality, and thus, produce an incorrect reference. Additionally, the brightest point is often overexposured in one or several color channels, because the dynamic range of the scene is wider than that of the sensor. The WB determination from overexposure areas would be non-correct.

The most correct WB could be achieved by putting a reference white/gray cart in the scene. The user can then point to this card as a reference for WB calculations. The above is true if only one illuminant is lighting the scene.

2.4.5 Demosaicing

The sensor with Bayer CFA samples full color scene in three-component RGB color space X_{RGB} with dimensions MxN:

$$Bayer(m,n) = \begin{cases} X_{RGB}(m,n,r), & \text{for (odd } m, \text{even } n) \\ X_{RGB}(m,n,b), & \text{for (even } m, \text{odd } n) \\ X_{RGB}(m,n,g), & \text{otherwise} \end{cases} \tag{2.3}$$

where, $m = 1, 2...M$, $n = 1, 2...N$ are Bayer pattern dimensions and r, g, b stands for R, G, B component of the scene. Demosaicing attempts to estimate X_{RGB} having only $Bayer$.

The demosaicing can be considered as the interpolating of full color planes from their decimations. Earlier demosaicing solutions employed independent interpolation of R, G and B color planes. This led to heavy distortions e.g. zipper effect, blurring, false colors, etc.

Modern demosaicing algorithms utilize several factors to produce high quality results. The first is that the G plane is less decimated. Thus, more details are preserved in this channel and it is interpolated first [64]. The second factor is the high correlation between R, G and B channels for natural images. Thus, there is a high probability that all channels share the same edge and texture locations. This correlation is usually utilized either via the color difference [96, 65] or the color ratio [90, 108] rule. The third factor is

that interpolation along edges produces fewer artifacts than across edges. Thus, many algorithms employ edge-directed interpolation. The edges are found by calculating the gradients in different color channels [65, 90, 108].

It has been observed that obtained interpolated values in one channel than could be utilized to refine interpolation in other channels. This procedure could be performed iteratively. The iterative algorithms typically provide higher quality for the price of additional computations [64, 90, 108, 69, 100].

Recently, the demosaicing was reformulated as a denoising problem [100, 199]. The difference between edge-directed interpolated color channels is considered as non-stationary "demosaicing noise". This noise is later removed by appropriate denoising method. This approach allows utilizing a great variety of denoising methods. Additionally, simultaneously with color "demosaicing noise" suppression, other noises that affect the image can be eliminated [137]. This is the so-called joint demosaicing and denoising [69].

Other demosaicing techniques include: demosaicing in frequency [45] or wavelet [44] domain, pattern recognition [28], restoration algorithms [127], local polynomial approximations [136], etc.

The demosaicing from other than the Bayer CFA (especially with non-regular grids) pattern may be problematic due to lack of high quality interpolation algorithms. A universal demosaicing algorithm appropriate for imaging pipelines employing RGB CFAs with an arbitrary grid was introduced in [111]. The performance, however, is poorer than with methods that are specially designed for specific patterns.

The Bayer CFA pattern benefits greatly from the existence of a great variety of demosaicing algorithms that could meet practically any requirements. Although invented more than 30 years ago, it still remains dominant among the used patterns.

2.4.6 Color Correction

The color processing in the camera goes according to the scenario shown in Figure 2.6 [163, 149]. Every digital camera accomplishes these operations in a direct or indirect way.

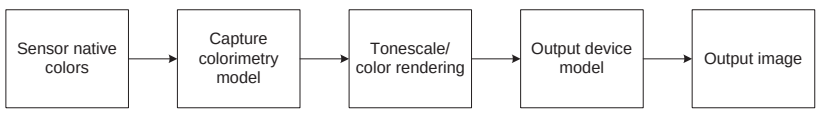

Figure 2.6: Color processing in digital cameras.

The data captured by a sensor is in the color space of a camera (camera native colors). Generally, camera color space is a function of spectral characteristics of the optics, lighting conditions, CFA filters, etc. Therefore, the color vision of the camera is far from the original scene colors and from that perceived by HVS [91, 83]

The output of the capture colorimetry model (Figure 2.6) is an estimate of the true scene colors. This estimate is represented using device-independent color spaces: CIE XYZ [169], CIE L*a*b* [75] or RIMM RGB [10].

The WB is the procedure preceding conversion from the camera colors to the intermediate device-independent color space. The WB compensates for different lighting

conditions and provides an accurate representation of a bright neutral object. The colorful parts need to be corrected. The color correction can be performed by a 3x3 matrix operation if the sensor spectral sensitivities are expressible as linear combinations of HVS color matching function. Although this is never the case in practice, this solution works sufficiently well [163].

The correction is typically carried out after transformation to device-independent color space. The complexity of this transformation may vary from simple matrix-based models to complex 3D look-up tables.

The next operation (Figure 2.6) is color rendering. For the purpose of natural scene color reproduction, this step should be skipped. In practice, color rendering is required to account for psychological (humans prefer more colorful objects than they are) and technological (dynamic and color range of the scene is higher than can fit into the rendered image) factors. A drawback of this step, is that the noise level is typically increased and unwanted artifacts may be produced [182].

The last step is conversion to the output model (Figure 2.6). Ideally, it should be a target device (monitor, printer) model. In practice, a universal model suitable for the most of devices is used such as sRGB [71] or AdobeRGB [6].

While most advanced cameras directly follow these color processing steps, the simple cameras could use a simplified transformation from sensor native colors to output model. All color corrections are then performed in the output color model domain [66, 166].

2.4.7 Gamma correction

The sensor response is linearly proportional to the number of photons and thus to the light intensity (2.1). The input voltage-to-emitted light intensity response of most displaying devices (such as cathode-ray tube - CRT and LCD) is nonlinear. In many cases, it can be described as the exponential function:

$$luminance = voltage^{\gamma}, \tag{2.4}$$

where γ is the system-dependent parameter. The GC can be explained as a process of compensation for this nonlinearity in order to achieve the correct reproduction of the intensity:

$$output = input^{1/\gamma}. \tag{2.5}$$

In most capturing systems, GC is carried out once in the acquisition part of the IPP. Image data stored in most formats are gamma pre-corrected. This solution causes a problem in the case of transferring an image from one platform to another [P2].

In digital imaging, GC is used not only for compensating nonlinearity of voltage-to-luminance response of displaying device. It is also used to improve perceptual coding efficiency [146]. On average, the HVS resolves two different intensities in ideal conditions if the difference between them exceed certain, intensity-dependent, threshold. The eye is more sensitive to the difference between two intensities, when the intensity level is small, than when the intensity level is high, since the relative difference between two neighboring codes is higher for the lower code values. The GC allows the readjusting of codes in such a way that lower luminance levels are quantized more accurately and high luminance values are quantized more roughly (Figure 5 in [P2]).

Hence, light representation by codes after GC becomes nonlinear. However, it becomes linear from the HVS point of view, that is, a linear increase in code number will linearly increase lightness sensation [146].

For coding GC data, the encoding error may be independent of the base level. However, it is not acceptable for the encoding error to be independent of the base level for non-GC data. This can be illustrated by a real life example (Figure 2.7) [P5]. As can be seen, the step-like and blocking artifacts are clearly visible on dark and gray parts of the compressed images because they have been increased by GC.

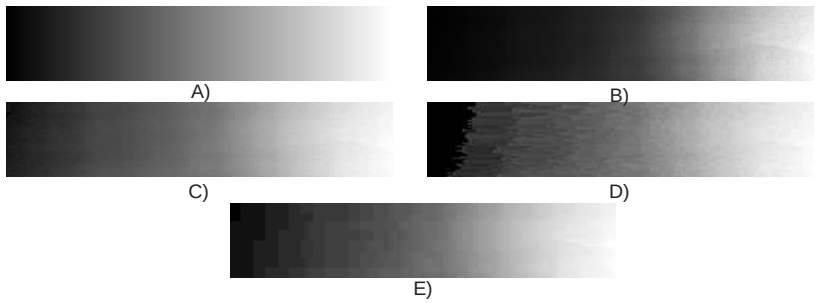

Figure 2.7: Illustrative example. Picture with intensity linearly changed from black to white (A). Data captured by the camera (B). Picture reconstructed from non-compressed data (C). Picture reconstructed from data lossy compressed by: JPEG-LS (D), standard JPEG (E).

2.4.8 Denoising, Deblurring and Image Enhancement

The sensor data are contaminated by noises from different sources [176]. First, to arrive from the image-capturing process is signal-dependent shot noise. The rest are additive noises: dark-current, fixed-pattern (non-uniformity), reset noise (circuits commutations), thermal noise (analog signal magnification) and quantization noise (A/D conversion). In modern cameras, in most cases, noises could significantly degrade image quality and special care should be taken to prevent this happening.

Normally, the denoising starts from the noise model formulation. The most widely spread is the noisy model:

$$y(m,n) = x(m,n) + n(m,n), \tag{2.6}$$

where, $x(m,n)$ is an unknown true signal, $n(m,n)$ is noise, $y(m,n)$ is the observed signal and (m,n) is defined as in (2.3). Noise can be represented in the form $n(m,n) = \sigma(m,n)\eta(m,n)$, where $\eta(m,n)$ is an independent zero-mean noise with unit variance and $\sigma(m,n)$ is the standard deviation of $y(m,n)$. The standard deviation of nonstationary Gaussian noise with a signal-dependent component can be expressed as [69]:

$$\sigma(m,n) = a + bx(m,n). \tag{2.7}$$

For the data captured from the sensor: a corresponds to the Gaussian part, which models signal-independent noises, and b corresponds to the Poissonian component, which

represents the signal-dependent photon counting process. Recent methods for removing such noises are presented in [69, 53].

More advanced noisy modeling that accounts for the effects of under- and oversaturation is presented in [54]. A method for noise model parameter estimation from a single raw image is proposed. The practical use of this modeling can be found in [52].

The blur distortions arise from relative camera and object motions, AA filter, lens and focus imperfection. The convolution is a common method of blur modeling [152]:

$$y(m,n) = (x(m,n) \circledast \nu(m,n)) + n(m,n), \tag{2.8}$$

here, $\circledast$ denotes convolution operation, $\nu(m,n)$ is the shift invariant point-spread function (PSF), the rest are the same as in (2.6). The deconvolution tries to invert (2.8) and reconstruct $x(m,n)$ from a noisy and blurred observation $y(m,n)$. The unknown PSF leads to blind deconvolution problem. A good review of deconvolution methods can be found in [135].

Advanced algorithms for denoising and deblurring are, typically, computationally expensive estimation processes that are hard to implement in camera hardware.

Other image enhancement operations include but are not limited to: [176]

-*Sharpening.* Can be viewed as a simplified form of deblurring using convolution with a small (3x3, 5x5) high-pass filter kernel. The price is the increase of a noise.

-*Dead pixels.* Due to limitations of technological processes some photocites on the sensor show abnormal behavior: do not respond to light ("dead" pixels) or their charge does not correspond to captured photons ("hot" pixels). Such a process can be modeled by salt-and-pepper noise. These pixels are excluded from subsequent image formation by replacing them with estimates from surrounding pixels.

-*Blooming.* If it is not removed by sensor design, special care should be taken during the image processing stage.

-*Lens distortions.* Zoom lens often have geometry distortion. Also different types of aberration and vignetting affect picture quality. These artifacts should be corrected.

-*Brightness and contrast adjustment.* The optimal combination of brightness and contrast is highly user-dependent.

2.4.9 Compression

The reconstructed image in most digital cameras is stored in EXIF [133] format which is based on the JPEG [140] image compression standard. In addition to the compressed image, the EXIF file stores metadata describing the camera and picture-taking conditions.

Another option provided by some cameras is the TIFF/EP format [77], which is based on the TIFF standard [5]. This allows lossless image storing. It can store images that are not fully processed, together with camera color profiles and picture-taken conditions.

The JPEG2000 [175] based image compression could be an additional option in future cameras. This would provide more flexibility and additional services. The JPEG2000 is more demanding to memory and computational resources. Although JPEG2000 was standardized more than six year ago, it is not widely used. There are only few consumer digital still image cameras that are able to shoot images in JPEG2000 format.

Other alternatives for compression of reconstructed images are presented in Chapter 4.

2.5 Alternative Image Reconstruction

The image formation from raw data in most digital cameras is performed by the camera itself (Figure 2.8). This approach is capable of delivering a full-color image very fast after image capturing. In this way, however, original raw data captured by the sensor (also called "digital negative") are discarded after being used once for image reconstruction.

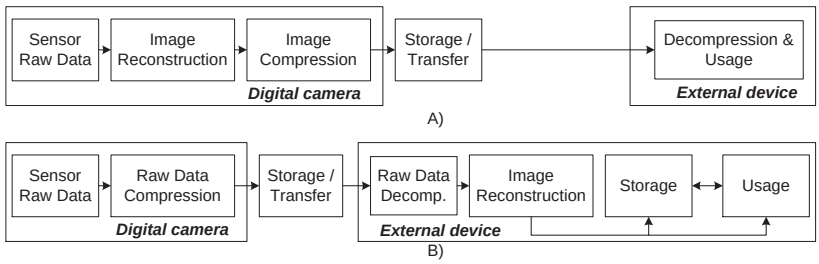

Figure 2.8: Conventional (A) and alternative (B) image formation.

An alternative to the conventional approach can be proposed (Figure 2.8) [P1]. In this method, camera functionality is limited only to raw data compression, while the image reconstruction is performed on an external device.

There are a number of advantages in the alternative image formation chain compared to the conventional one. Some of these are summarized below [P1]:

- *Image quality.* Due to the limited recourses it is challenging to apply optimal reconstruction algorithms (computationally expensive by rule) in the camera. Thus, image quality is always compromised due to hardware constraints. The transferring of raw data to an external powerful PC provides access to advanced algorithms providing the highest image quality.

- *Camera characteristics.* Digital cameras very often have limited computational and power recourses, so removing some processing steps will decrease time between shoots and prolong battery life.

- *Data redundancy.* The data redundancy is generated by demosaicing which generates three full-size color planes from a single raw plane. This redundancy is reduced by the compression in the last stage. This is a sub-optimal solution, since demosaicing does not provide any new information. The direct compression of raw data can potentially provide higher a CR or, alternatively, better image quality for a fixed CR.

- *System flexibility and scalability.* The external device may use reconstruction algorithms that are appropriate to its displaying and computational resources. For example, mobile phones and PDA may use very simple reconstruction algorithms, while for powerful PCs the highest quality can be achieved by advanced algorithms. Thus, the overall system may be more flexible and scalable.

As the price of the memory falls and processing power increases, the combining of both approaches is preferable. The camera produces reconstructed image using fast

algorithms and stores compressed raw data from the sensor. The reconstructed image is used immediately and the raw one is used later to reconstruct with the best quality if needed.

Recently, some algorithms for image processing in the CFA domain have been proposed. These include digital zooming [109], denosing [17, 80], deblurring [180], source camera identification [124, 23], sharpening [81], lens artifact correction etc. In general, algorithm performance is better, when applied prior to demosaicing. Also, the computational cost is reduced, since algorithms need to process less data.

The methodologies described above for image formation raise the problem of efficient CFA compression methods. The next Chapter is dedicated to this problem.

Chapter 3

Compression of raw sensor data

This chapter is dedicated to compression of sensor raw data with BCFA. The Bayer pattern is considered to be the most widely used in practice. First, the problem of raw data compression is addressed and directions for solving this problem are summarized. The different experiment setups for quality evaluation are presented. In the following parts, algorithms for lossless, near-lossless and lossy compression are introduced. Publications [P1]-[P5] contribute to this Chapter.

3.1 Problem Formulation

The BCFA data have the dimensions of the corresponding CCD or CMOS sensor and an accuracy of 10-16 bpp. Such data can be stored in several ways. Some primitive methods for archiving raw data, utilized in earlier cameras, are described in [P1]. The required storage space can be further decreased by applying compression. There are many peculiarities that make the raw data compression problem different from, but probably closest to it, the image compression area.

The first peculiarity is the specific structure of BCFA (Figure 2.4A). The CFA image is a combination of pixels from three color planes. Although these color planes should be highly correlated for natural scenes, the pixels from different planes will most probably have very different levels. This is due to the non uniform spectrum of the light source and the non uniform light spectrum sensitivity of the camera sensor. When pixels from such color planes are mixed together, they create an array with high frequencies that does not allow achieving of the high CRs.

Other peculiarities arise from the nature of the data. These are the linearity and the noisiness of the data. Because of raw data linearity, it is not acceptable for the encoding error to be independent of the base level (Section 2.4.7). In practice, this limits the number of methods that can be used for compression with losses. The noisiness arises from the capturing process (Section 2.2.1). Because of the nature of the light capturing process, the raw data are contaminated by signal-dependent noise. There are other noise sources as well (Figure 2.2, Section 2.4.8). For the lossless compression, the

presence of noise will simply decrease the compression efficiency of any method because the correlation between neighboring pixels is decreased. Lossy compression of noisy data could give interesting effects, such as, in the case of rather small CRs, the quality of the compressed image can be even better in comparison to the quality of the original noisy image [142]. This is due to the fact that under certain conditions, the main "loss" in lossy compression of a noisy image is related to the noise.

The generalized framework for BCFA compression is proposed in Figure 3.1. It separates preprocessing and compression operations. Color space transform is used to decorrelate color planes. Color planes de-interleaving is needed, in order to pack the pixels of the same color into a structure appropriate for the subsequent compression. For the R and B pixels this transformation is straightforward. These pixels can be directly packed into a compact rectangular form. For the G pixels, located on a quincunx grid, there are several possibilities. A pre-filtering may be used in one of the stages. The role of pre-filtering is to smooth out aliasing artifacts that occur during color planes de-interleaving and to facilitate compression. At the last stage, some standard compression algorithm may be used, such as that used for compression of full color images when image reconstruction is performed on a camera board.

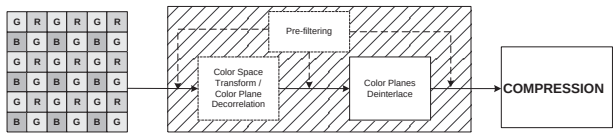

Figure 3.1: Generalized framework for Bayer CFA compression.

Most raw data compression methods can be described using this generalized framework. Algorithm developers either solve each stage of this framework separately or propose a combined solution. Different algorithms have been considered for lossless and lossy compression.

For lossy compression, a modified RGB to YCbCr transform suitable for BCFA has been proposed [97] and has become widely used. For lossless compression in RGB color space usually good decorrelation can be obtained by coding color differences: R-G, G, B-G. There are several ways to calculate these differences from the BCFA [201, 40],[P1]. Efficient color transforms from BCFA to 4:2:2 or 4:2:0 YCbCr color space which is suitable for image and video compression, were proposed in [173]. The drawback, however, is the reduced resolution.

Color plane de-interleaving is needed when transforming from a quincunx to a rectangular grid. A large variety of methods has been proposed. Some of them are non-reversible and thus could not be used for lossless compression.

A pre-filtering is used prior to or as a part, of the color planes de-interleaving process. Pre-filtering is often a nonreversible procedure.

The CMOS technology allows integration of digital and analog circuits on a sensor chip and their fabrication using same the lithography processes. This advantage is utilized for designing the CMOS chips that contain circuits facilitating raw data compression. One example is the CMOS sensor with integrated analog image compression [72].

Charge-prediction circuits for pixel-level predictive coding are proposed in [99]. More

sophisticated on-sensor pixel predictive coding using adaptive quantization circuits is presented in [165]. The architecture of a sensor with block-based differential pulse code modulation (DPCM) with reduced compression artifacts is demonstrated in [200].

There are also sensors that support transform-based image encoding. Integration of circuits for analog 2D DCT computation was reported in [86]. Later, the A/D convertor directly quantizes DCT coefficient values instead of pixel values. There is also a sensor with 2D DCT calculation via arithmetic Fourier transform [172]. There are CMOS sensor designs with integrated wavelet transform [117] and SPIHT [104] algorithm. All use simplified methods with 8x8 block-size support.

A CCD sensor has been proposed [88], which outputs differential pixel values in a special manner that supports hierarchial lossless compression and transmission.

Recently, a number of methods for compression of videos captured using BCFA-based sensors were proposed [58, 85, 39, 42]. For video, reducing the amount of data to be processed is even more important. The most computationally expensive procedure - Motion Estimation - is significantly accelerated, while working with reduced data.

3.2 Quality Evaluation for Raw Compression

Let us consider the sequence of operations: raw data generation, compression, decompression and image reconstruction. Depending on compression type and goals to be achieved, different experimental setups are feasible (Figure 3.2). Under Quality Metric (QM), we assume the metric used for assessment differences of between source and target images. This can be an absolute pixel difference, MSE, PSNR, Structural SIMilarity (SSIM)[184] and other objective and subjective methods. Compression Metric (CM) stands for CR, bitrate or file size.

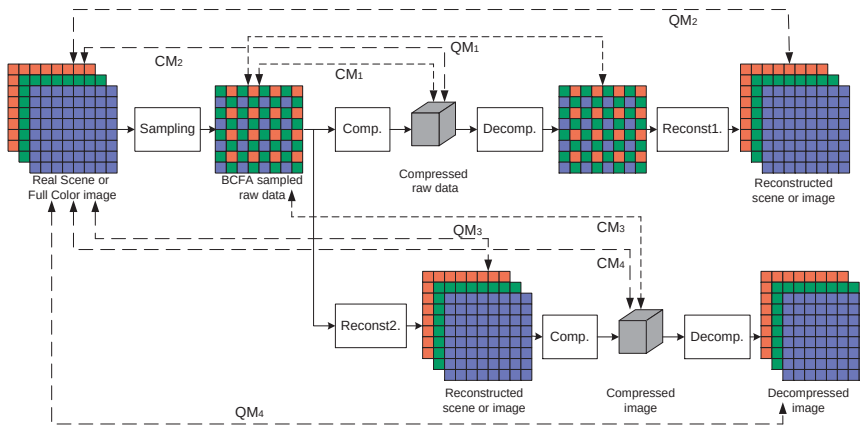

Figure 3.2: Quality evaluation for BCFA raw data compression

Raw data used for experiments could be captured either directly by the digital camera or obtained from a full color image by sampling according to the BCFA pattern. In the later approach, the fact that raw data are heavily processed before obtaining full color images is not taken into account. However, some assumptions that may be true for

reconstructed images may not hold for real raw data. Thus, the compression efficiency of different methods may vary. A common limitation of many publications on raw data compression is that evaluation of the compression methods is done only for artificial raw data generated from reconstructed images.

For lossless and near-lossless methods, the only changing factor is a CR, while distortions are not allowed or are bounded. For these methods it is reasonable to evaluate their performance directly on real CFA data obtained from digital cameras. Algorithms could be compared only with CM_1 for the same raw data and the maximum allowed error (in case of near-lossless compression). This approach is utilized in Sections 3.3, 3.4.

The lossy compression methods have two changing factors: CR and reconstruction error. To compare two compression methods, QM_1 as a function of CM_1 is valuable. The same metrics could be used for lossy algorithm evaluation on raw data, since the original scene is not available. This method is used in Section 3.6. The problem, however, is that there is no linear dependency between the quality of decompressed raw data and the quality of the full color image reconstructed from it.

Often one needs to evaluate the performance of compression method in combination with some reconstruction algorithm. This evaluation requires original non-disturbed scenery data. For this, the usage of artificial raw data is reasonable. Because it is generated from a fully reconstructed image, the only required image reconstruction operation in this case is demosaicing. The dependency of QM_2 on CM_1 or CM_2 could be used for evaluation.

The conventional IPP could be modeled by first applying reconstruction to BCFA data, followed by compression. The QM_3 in this case could be used to determine the maximum reconstruction quality in the conventional chain. This quality starts to degrade because of lossy compression of the reconstructed image. The dependency of QM_4 on CM_3 or CM_4 is used for evaluation of the conventional chain with compression.

Comparison of conventional and alternative IPP is possible by comparing QM_4 on CM_3 and QM_2 from CM_2. The *Reconst*1 may differ from *Reconst*2, because external host typically have more resources and are able to provide more sophisticated reconstruction algorithms.

The approach described above for simplified evaluation of lossy compression is used in Section 3.5.

3.3 Lossless Compression of Raw Data

In this section, lossless compression using a generalized framework (Figure 3.1) is presented. The performance of different de-interleave methods, as well as color plane decorrelation using color differences is tested for real data. A comparison with other methods for lossless compression of CFA data is carried out.

After color planes have been de-interleaved, R and B planes are naturally transformed by removing blank rows and columns into a compact rectangular form. It is common first to encode the G plane, since it has more pixels and thus higher inter-pixel correlation and can be used later on to improve performance of R/B plane coding. The G pixels need to be transformed from a quincunx to a rectangular grid before the standard compression algorithm can be applied.

3.3.1 Green plane compression

In pioneering work on raw image compression [181] it was proposed to use interpolation to fill the missing G pixels. This results in a G plane with two times more pixel than the original, which does not facilitate compression. Later authors tend to retain the number of G pixels unchanged.

A comparison of different methods for transforming from a quincunx to a rectangular grid was carried out in [P1] and [P3]. Among many existing approaches only few are fully reversible and suitable for lossless compression (Figure 3.3)[P3]:

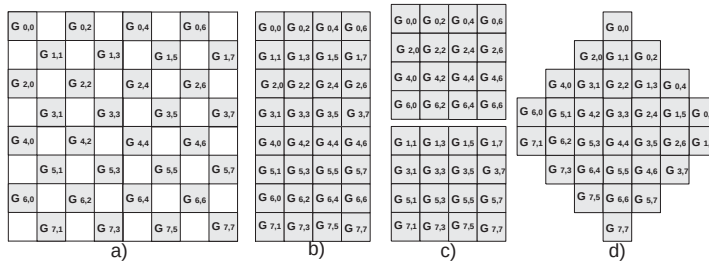

Figure 3.3: Methods for transforming from quincunx to rectangular grid. Original quincunx grid (a). Merge (b). Separation (c). Rotation (d).

- *Merge.* The G pixels are grouped together in a raster scan order, in a horizontal or vertical direction (Figure 3.3b). After such a transformation the horizontal or vertical edges may introduce artificial high frequencies to the data, causing non-optimal compression. To overcome this problem directional (horizontal or vertical) low-pass filter prior compression may be used [92].

- *Separation.* The G pixels from a quincunx plane are split into two rectangular planes [92], [P1]. The first includes odd indexed pixels (odd rows and odd columns). The second includes even indexed pixels (Figure 3.3c). These are often named G_r and G_b.

- *Rotation.* The transformation from the quincunx grid into the two-dimensional array is done by rotating data 45 degrees clockwise and removing the blank columns and rows [97, 35] (Figure 3.3d). This is accomplished by the following transform:

$$\begin{bmatrix} i \\ j \end{bmatrix} = \frac{1}{2}\left(\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} -1 \\ Z-1 \end{bmatrix}\right),$$

where, $i + j = odd$ are coordinates of the pixel position in CFA and Z is the width (height) of the CFA data. Despite the simplicity and naturality of such a transform, it is not easy to compress the rotated plane by a standard compression method. It is relatively easy to modify a method that processes images line by line (JPEG-LS) or by small blocks (JPEG). However, it is difficult to do the same with JPEG2000.

- *Prediction.* This method is based on a separation method [P3]. It utilizes the fact that pixels of an odd-indexed sub-plane are located between even-indexed pixels. Thus, pixels from both sub-planes should be highly correlated. The current pixel from the odd-indexed plane is predicted using pixels located in the northwest, northeast, southeast and southwest of the even-indexed sub-plane:

$$\hat{G}_{i,j} = F(G_{i-1,j-1}, G_{i-1,j+1}, G_{i+1,j-1}, G_{i+1,j+1}) \tag{3.1}$$

  where i, j are odd numbers, F is a prediction function, which can be mean, median, etc. The prediction errors $e_{i,j} = G_{i,j} - \hat{G}_{i,j}$, are reduced modulo α to be within the interval $-\lfloor \alpha/2 \rfloor$ and $\lfloor \alpha/2 \rfloor - 1$, where α is an alphabet size. The resulting sub-plane of prediction errors is encoded by a similar compression algorithm as other sub-planes.

We have tested the performance of different methods in a database composed of artificial CFA data generated from full color images and from real CFA data from different cameras capable of shooting in raw mode (Figure 3.4)[P5]. After transforming from a quincunx to a rectangular grid, the G sub-planes are compressed by a standard JPEG-LS algorithm (The JPEG2000 performs worse on separated planes, as was shown in [201].)

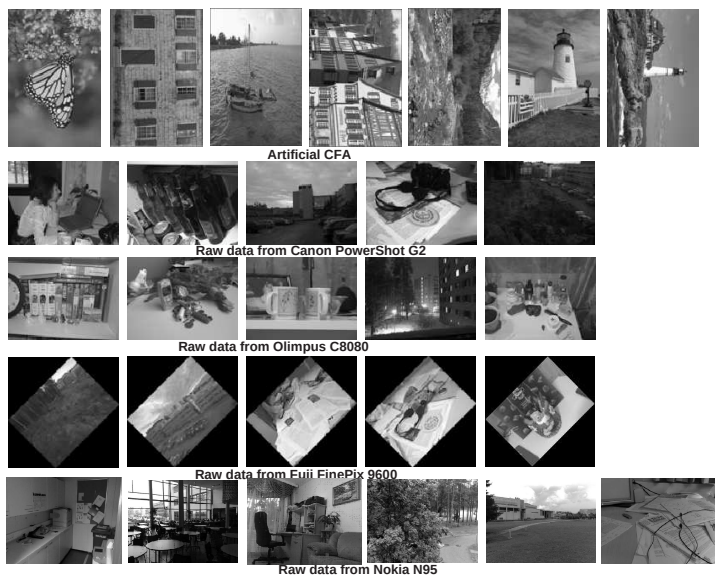

Figure 3.4: Thumbnails of test images. (Fuji FinePix9600 has an image sensor rotated by 45 degrees).

The average bitrates of lossless compression by JPEG-LS of the outputs of different de-interleave methods for the images from our database are presented in Figure 3.5. As can be seen, it is hard to select a best performing method. This is in contrast to lossy compression, where the selection of an appropriate de-interleave method significantly affects the performance of the compression algorithm [P3], [92, 191]. The rotation method

shows slightly better performance (1-1.5% on average) for the real CFA data. The minor gain in bitrate allowed by the rotation method is, however, neglected by the fact that the compression algorithm could not work in the standard mode and should be modified.

The prediction method shows its efficiency in lossy compression [P3] performed at the same level as other methods. Additionally, it should be noted that with the Bayer pattern, the G_r and G_b are normally read out from the sensor via different circuits. Due to imperfection in the sensor process creation, it might be that the G_r and G_b are unbalanced. In this case, the minimum bitrate could be achieved by a simple separation method.

It can be concluded that any of the reviewed methods for G plane de-interleave can be used for lossless compression without a significant difference in performance.

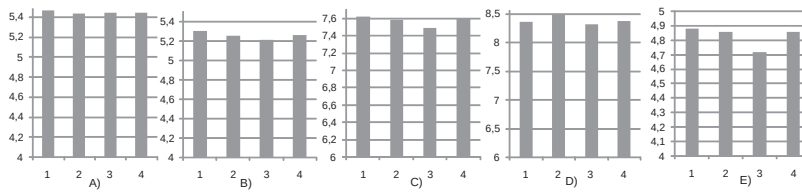

Figure 3.5: Bitrates of green channel for different quincunx to rectangular transforms: 1-separation, 2-merge, 3-rotation, 4-prediction; Artificial CFA (A). Canon G2(B). Olympus C8080 (C). Fuji FinePix (D). Nokia N95 (E).

3.3.2 Red and blue planes compression

Once the G plane is encoded and available for the decoder part, it could be used to facilitate compression of R/B color planes. In BCFA the values of a G pixel at the R/B pixel position are unknown and should be estimated. The nearest G pixels from the same row or column [P1] can be used. More efficient way is to use mean, median [P1], bilinear, bi-cubic spline [201] interpolation or low-pass filtering [40]. It was shown in [201] that a simple bilinear interpolation provides adequate results.

The estimate of the G pixel at the R/B pixel position is done by equation (3.1), where i is odd and j is even for R pixel positions, and vice versa for B positions, F is a mean or median. The difference planes are obtained as follows:

$$Rd_{i,j} = R_{i,j} - \hat{G}_{i,j}; Bd_{i,j} = B_{i,j} - \hat{G}_{i,j}$$

The differences are reduced modulo α to be within the interval $-\lfloor \alpha/2 \rfloor$ and $\lfloor \alpha/2 \rfloor - 1$, where α is an alphabet size.

The bitrates for lossless compression of original R/B planes and color difference planes with JPEG-LS for the images from our database are given in Figure 3.6.

Encoding color differences instead of original planes decreases the bitrate by approximately 6-7% for the artificial CFA data, which is significant for lossless compression. In contrast, for real CFA data, encoding of color differences, on average, does not decrease the bitrate or even increase it. This illustrates the importance of compression method evaluation on real raw data instead of artificial ones.

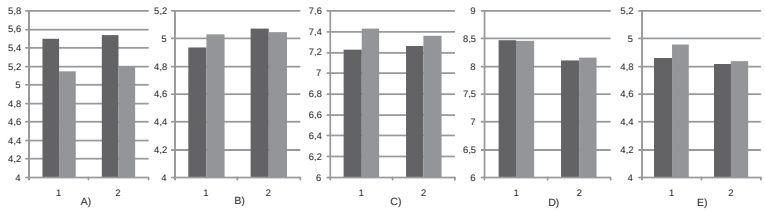

Figure 3.6: Bitrates for 1 - R and 2 - B channels for intra (dark gray) and differences (light gray); Artificial CFA (A). Canon G2(B). Olympus C8080 (C). Fuji FinePix (D). Nokia N95 (E).

Two factors that decrease the performance of the color de-correlation technique are described in [P5]. First is the noise, which is always present in real raw data. As a result, SNR in the difference plane is higher, and this degrades the performance of a compression algorithm. The second factor is a very big difference in intensity levels of signals in different channels. In most reconstructed images, not only signals in channels are highly correlated, but also their intensity levels are close. For real CFA data, signals in channels are still highly correlated but absolute values can differ very significantly from one color channel to another. The compensation for these factors is done by white balancing and color correction procedures during image reconstruction process. These are complex and sensor-dependent procedures (Sections 2.4.4,2.4.6). For high quality imaging, at which lossless correction is targeted, it is preferable to set a correct WB and correct colors manually during the image reconstruction process. The correction of signals in different color channels prior to this is not desirable.

In [P5] we estimated how the variations in intensity levels affect the color planes de-correlation by color difference. Based on that study, the simple method is proposed to predict when color difference will have a smaller entropy compared to the original color plane. The method relies on average ratios of signals in R, G and B color channels. The advantage of the proposed approach is that average values of R, G and B can be calculated during readout from the sensor and thus require practically no additional computations.

3.3.3 Alternatives for lossless compression

Different algorithms for lossless image compression have been compared in the application to BCFA compression in [157]. Algorithms were tested on a real raw data database with more than 200 images. The G plane was processed using the merge method and R, B coded as intra. The 12 bpp raw data were truncated to 8 bpp to make them accessible for all algorithms. The Glicbawls [125] algorithm provides the highest CR among 10 reviewed methods. It outperforms CALIC [190], JPEG-LS and JPEG200 by 5.2, 7.4 and 10.1 percent, respectively.

Instead of quincunx to rectangular transformation of the G plane, one may use lifting-scheme wavelet transform that could work directly on a quincunx grid [61]. The transform was developed for compression of remote satellite sensors. These sensor output images are sampled by two CCDs shifted by 1/2 pixel, which is motivated by modulation transfer function of the satellite.

The methods described in Sections 3.3.1 and 3.3.2 are based on separate compression of color planes. This is a simple, but sub-optimal solution. Improving this approach using transform or spatial based methods is possible.

It has been observed that application of JPEG2000 to non-separated raw data provides rather good compression results [201]. This is due to the fact that wavelet transform at the first level of decomposition decorrelates and separates color planes into four subplanes. Experimentally, it was determined that the best decorrelation is achieved by, the so-called, 5/3 Mallat wavelet packet transform. Based on this transform and low-complexity adaptive context-based Golomb-Rice coding, an efficient lossless compression technique was proposed. The above method was further improved by edge-directed prediction of residuals of 5/3 wavelet packed transform [130].

The JPEG-LS algorithm uses a simple Median Edge Detector (MED) predictor (Section 3.4.4) (Figure 3.7A). It can be shown that when MED is applied to the transformed G plane, the pixel support becomes different (Figure 3.7B-E) [P1]. This reduces coding efficiency.

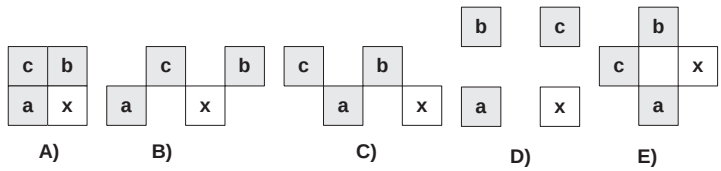

Figure 3.7: Pixel support for MED predictor with different methods: original (A), merge(B, C), separation, prediction (D), rotation (E).

A more advanced context-matching based predictor, designed specially for the quincunx grid, is presented in [24]. The R and B planes are decorrelated using adaptive color difference. The differences are calculated using directional derivatives from surrounding pixels. Another context-based method implemented on neural networks was presented in [22].

Optimal scalar and vector predictors for BCFA are studied in [8]. The derivation of an optimal scalar predictor for predefined Manhattan distance is given. For a pixel with coordinates i, j, the predictor support includes all previously encoded pixels within $i \pm a$ $j \pm b$, where $|a|+|b|$ is less than the Manhattan distance K. For optimal vector prediction, vectors are defined as non-overlapping 2x2 blocks of the BCFA image $\hat{p}_{bayer}(i,j) = [R(i,j), G^r(i,j), G^b(i,j), B(i,j)]$.

The comparison of different methods is given in Figure 3.8. Comparison was done only for images that are common in all publications. Algorithms presented in this section provide significant CR improvement for artificial CFA data. At the same time, its performance for real raw data is similar to more simple method from [P5].[1] This is mainly due to the fact that possible decorrelation gain for real data is much smaller as shown in Section 3.3.2.

[1]We have tried to contact the authors of the corresponding papers for support in evaluation on real raw data. However, no reply has been received yet. We implemented the methods in [201, 24] ourselves. Performances of both methods depend on used context modeling for the Golomb-Rice coder. There is a several percentage difference between our and the original implementations.

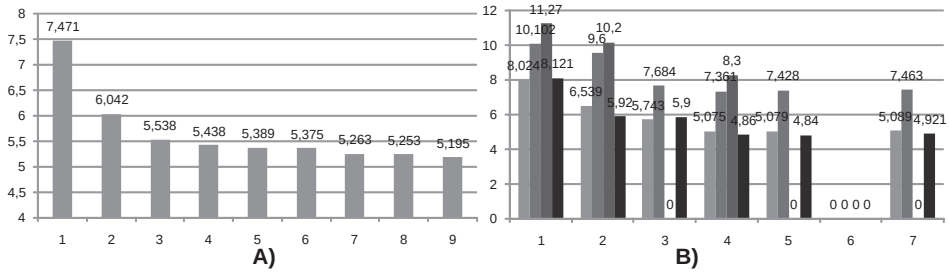

Figure 3.8: Lossless bitrates for different methods tested on artificial (A) and real raw data (B). 1 - image entropy, 2 - direct JEPG-LS, 3 - direct JPEG2000, 4 - JEPG-LS [P5], 5 - method in [201] , 6 - method in [130], 7 - method in [24], 8 - scalar method in [8], 9 - vector method in [8]; On (B): light to dark - Canon G2, Olympus C8080, Fuji 9600, Nokia N95

3.4 Near-Lossless Compression of Raw Data

3.4.1 Near lossless compression

As a compromise between lossy and lossless compression a near-lossless compression could be considered. Near-lossless compression could be a good alternative to a lossless compression for applications that require high image quality. It provides higher CR at the price of small and controllable errors. An algorithm could be considered as near-lossless if it provides very high values of objective quality measures (such as PSNR). This implies that no visual differences to the original image are encountered.

However, some applications require more strict conditions than just visual similarity. This is especially important for medical and astronomical imaging applications, where higher error values may lead to a wrong diagnosis or miss-detection. For this, "near-lossless" is defined in a sense that each reconstructed image pixel ($\tilde{I}_{i,j}$) differs from the corresponding original image pixel ($I_{i,j}$) by not more than a prespecified value δ. The purpose of the near-lossless algorithm is to obtain the highest CR while the relation

$$|I_{i,j} - \tilde{I}_{i,j}| \leq \delta \tag{3.2}$$

is guaranteed. In this section only the second definition is considered.

Near-lossless compression can be realized in the pixel or in the transform domain (Figure 3.9) [9]. In the pixel domain, the encoder deals with residual error that is obtained between the original and predicted image samples $e = I - \hat{I}$. This error is quantized by δ providing near-lossless reconstruction $\tilde{I} = \hat{I} + \hat{e}$. The quantized errors $\hat{e}$ are sent to the entropy encoder.

Near-lossless encoding in the transform domain is more complex. Quantization in the transform domain aims at minimization of energy in reconstructed errors rather than constraint reconstruction error for individual pixels. The reason is that the single transform coefficient affects a group of pixels. The filterbank-based image coding technique that guarantees at least a given percentage of the reconstructed errors below the required δ is presented in [84]. In order to guarantee (3.2) the encoder would use unnecessarily high bitrates.

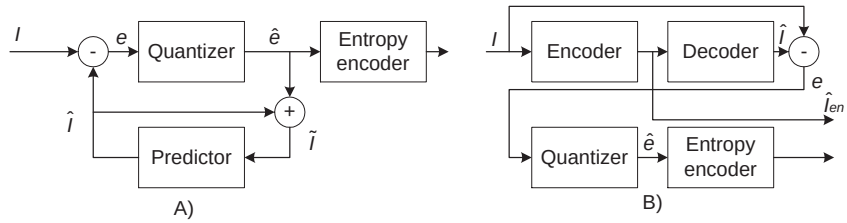

Figure 3.9: Near lossless compression in pixel (A) and transform (B) domains.

To overcome the above-mentioned problem, two-stage encoder can be used [123] (Figure 3.9B). At first, the two-stage encoder produces a lossy bitstream $\hat{I}_{en}$. Its decoded version is used as a predictor for pixel values $\hat{I}$. The prediction residuals e are quantized $\hat{e}$ and sent to the entropy encoder. Optimal bit allocation between lossy encoded image and error residuals was studied in [197]. The main benefit of the transform-based approach is the quality scalability.

3.4.2 Existing approaches

As was illustrated in Section 2.4.7, raw data are not suitable for lossy compression by methods where the allowable encoding error is uniform for all intensity levels.

Therefore, some methods [192, 193] aimed at near-lossless compression of artificial BCFA data are not suitable for real raw data compression.

There are two possible solutions:

- to non-linearly scale raw data prior to compression.
- to modify the compression method by using adaptive error quantization.

The first approach is used in the Nikon lossy NEF format for the raw data compression exploited in some DSLR cameras. The algorithm details are available from Dave Coffin's reverse-engineered, open-source RAW converter, DCRAW [27].

The 12 bpp resolution values can have $2^{12} = 4096$ values. These are reduced to 683 values by applying a quantization curve [120]. The curve resembles a gamma correction curve: linear for values up to 215, then quadratic (Figure 3.10).

These 683 values are then encoded using a variable number of bits (1 to 10) with a tree structure similar to the lossless Huffmann compression scheme. The decoding curve is embedded in the NEF file. It could be changed by a firmware upgrade without having to change NEF converters.

In this approach, practically any lossless method could be used for the compression after the quantization. The lossy and near-lossless methods are not allowed when the aim is near-lossless compression. However, the near-lossless method could potentially achieve higher CR when managing error allocation itself. Also, this method has reduced flexibility. That is, if data was corrected so that $\delta_{rec} = 3$ is guaranteed, then with subsequent near-lossless compression it is possible to achieve only δ_{rec} multiple of 3 (δ_{rec} =1, 2, 4, 5, etc. not accessible).

As an alternative, a modified compression method with adaptive error quantization is described in the following subsections. This is relatively easy to implement based on

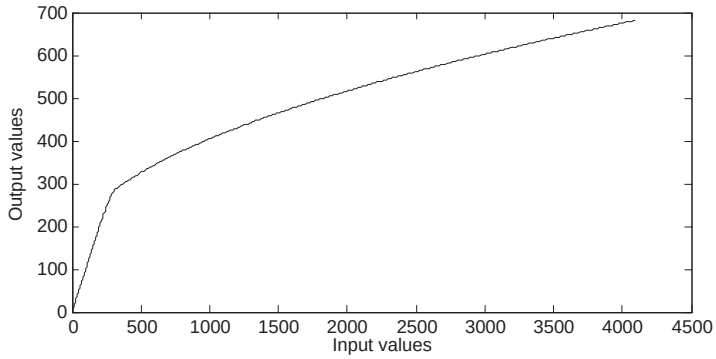

Figure 3.10: Nikon quantization curve.

a method that processes images on a pixel-by-pixel base (e.g. JPEG-LS). The problem arises for those which process whole image at once, or by big tiles (e.g. JPEG2000), or in a block-by-block (e.g. JPEG) manner.

3.4.3 Adaptive error quantization function

The crucial point for the method with adaptive error quantization is what should be the function of the allowable encoding error for intensity levels. In the considered case, GC is the process that masks the higher errors for greater intensity levels. Thus, one can use a straightforward approach to find the allowable encoding error for different intensity levels [P2].

For each luminance value x in the range between 0 and $N = 2^P - 1$ (P is bpp) we take an 'etalon' output value c and two values a and b deviating by $m \in [0, 1, 2, ...]$, all three being corrected with the reciprocal of the desired γ (F means rounding):

$$a = F\left(N\left(\frac{x-m}{N}\right)^{\frac{1}{\gamma}}\right); b = F\left(N\left(\frac{x+m}{N}\right)^{\frac{1}{\gamma}}\right); c = F\left(N\left(\frac{x}{N}\right)^{\frac{1}{\gamma}}\right) \tag{3.3}$$

If either of the values a and b deviates from the 'etalon' value by more than the *allowable reconstruction error*, the *allowable encoding error* for x is made equal to $m-1$ and stored as the allowable encoding error for the current value of x. A pseudo-C source code for the above procedure can be found in [P2].

Figure 3.11 depicts the allowable encoding errors ($F(\gamma, \delta, P)$) obtained for a allowable reconstruction error equal to one, gamma equal to 2.2 and 10 bpp precision. "Tricky" regions where the allowable encoding error is not stable are avoided by bounding the threshold function (the thick dashed line). Consequently, a zero encoding error is assumed for intensity levels below 220, an unit error is allowed between 221 and 810, while for intensities higher than 810 it is equal to 2.

For 8 bpp precision (256 intensity levels), the typical tolerable reconstruction errors are 1, 2 and 3. An error equal to 3 out of 256 levels accounts for 1.17 percent. This is around the visibility threshold of 1%. Higher precision or raw data (10 or 12 bpp) allow increasing the tolerable reconstruction error. For example, for 10 bpp precision the

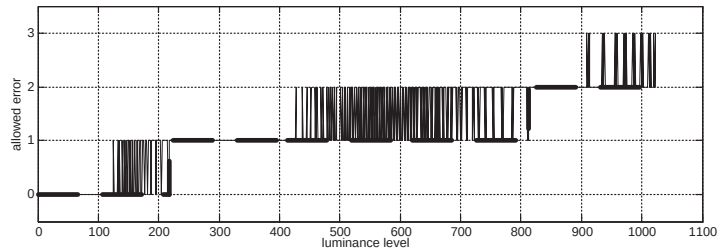

Figure 3.11: Allowable encoding error produced by proposed procedure (see text) (thin solid line); the same function restricted from bottom (thick dashed line).

encoding errors of 4, 8 and 12 correspond to 1, 2 and 3 error values for 8 bpp precision (12 is 1.17 percent of 1024). For 12 bpp precision these are 14, 32 and 48. Example curves for different combinations of precision, allowable encoding errors and gamma can be found in Figure 6 in [P2].

The image reconstruction in different systems may require different γ [146]. To allow condition (3.2) to be satisfied for several possible γs one should calculate the allowable encoding error 3.3 for the smallest γ. As can be seen from Figure 6 in [P2] this would guarantee the reconstructed error for all higher values of γ for all intensity levels except very low ones. To make this condition satisfying for all values, the combined allowable encoding error function $min(F(\gamma_{min}), F(\gamma_{max}))$ should be used. This comes at the expense of a decreased allowable encoding error and thus CR.

3.4.4 Modified JPEG-LS algorithm

In this subsection we illustrate how the allowable encoding error functions calculated earlier are used with JPEG-LS to achieve near lossless compression.

The block diagram of the modified JPEG-LS encoder [187] is presented in Figure 3.12. The encoder consists of two parts: source modeler and entropy coder. The JPEG-LS modeler is composed of a fixed and an adaptive predictor. The fixed predictor performs the primitive edge detection test, while the adaptive part is a context-dependent integer additive term. As a fixed predictor, the MED is used; a simple predictor with rudimentary edge detection capabilities.

The obtained prediction value x_{MED} is corrected by a context-dependent term calculated using previous image statistics. The distribution of prediction residuals is approximated by two-sided geometric distribution and efficiently encoded by adaptively selected Golomb-type codes.

In near-lossless mode, prediction residual e is the quantized by maximum allowed reconstruction error δ: $e_q = F(e/(\delta + 1))$ (F means rounding). Entropy coding is done similarly as in the lossless mode. In the decoder, dequantization of decoded error is performed.

To allow non-uniform reconstruction errors, the quantization value is adaptively selected depending on image pixel value. A complete description of the algorithm is presented in Section 5 [P2].

The results of near-lossless compression by the modified JPEG-LS of real CFA images from our database (Figure 3.4) are presented in Figure 3.13. All color planes were

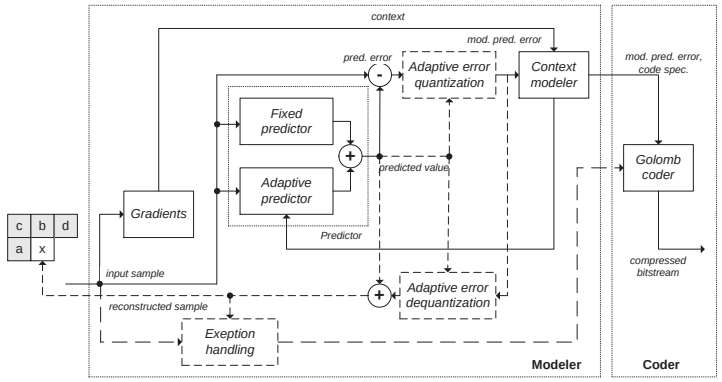

Figure 3.12: Block diagram of the modified JPEG-LS algorithm that allows near lossless compression of raw data.

compressed separately. For the G plane the separation method was used. The allowed encoding errors were selected as δ_1=4, δ_2=8, δ_3=12 for 10 bpp precision (Canon G2, Nokia N95) and δ_1=16, δ_2=32, δ_3=48 for 12 bpp precision (rest of cameras). Both these correspond to δ_1=1,δ_2=2, δ_3=3 for 8 bpp precision. The gamma used for the allowable encoding calculation 3.3 was fixed at 2.2.

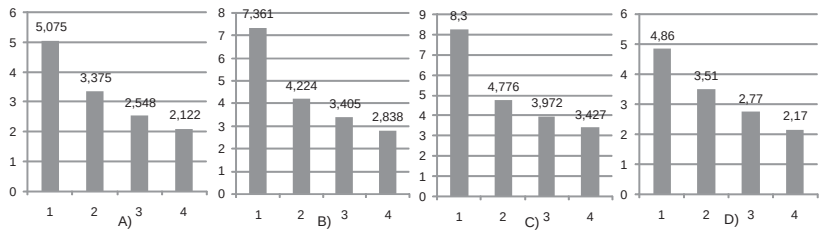

Figure 3.13: Near lossless bitrates of the modified JPEG-LS algorithm: 1 - lossless, 2 - δ_1, 3 - δ_2, 4 - δ_3. Canon G2(A), Olympus C8080 (B), Fuji FinePix (C), Nokia N95 (D).

As can be seen, even with the smallest reconstruction error the near-lossless method provides bitrates 1.5-1.7 smaller than strictly lossless methods. With an increase of allowable reconstruction error, the compression gain becomes smaller, since this gain is mostly achieved for the pixels with high intensity levels, that occur less often in a typical non-GC image. Increasing the reconstruction error from δ_1 to δ_2 (2 times), the bpp is decreased by 21-30% only. With a further increase to δ_3 the bpp decreases even smaller (14-20%).

3.5 Lossy Compression

In this section, a simplified approach to lossy raw data compression is considered. The results are obtained for artificial raw data generated by sampling full color test images according to the BCFA pattern. The obtained data are not real CFA data since all

image formation operations (except interpolation) have already been done. However, this section is focused mainly on demosaicing and compression, where other operations are not considered here.

The original test images could be considered as real visual scenes presented to the digital camera. The synthetically generated BCFA data are considered as the data captured by a camera with a single sensor. The compression and reconstruction operations are considered. In the conventional method, the CFA data are first demosaiced and then compressed. In the alternative IPP, these operations are reversed. Here, firstly, theoretical motivations for reversing demosaicing and compression are presented. Then compression methods and experimental results are described.

3.5.1 Reversing demosaicing and compression

Originally, there were a number of empirical studies, like [97, 92], which indicated an alternative approach superior to the conventional one for low (<10) and high (>80) CR. Later, the theoretical basis supporting these results was published [103].

Let I be the original and $\bar{I}$ be the BCFA image. The compression and demosaicing operations will be $C(I) = I + \delta I$ and $D(\bar{I}) = \bar{I} + \Delta\bar{I}$, respectively. The δI and ΔI are compression and demosaicing errors. The conventional and alternative chains are then modeled as $C(D(I)) = C(I + \delta I)$ and $D(D(\bar{I})) = D(\bar{I} + \delta\bar{I})$.

The total error in the conventional chain is expressed as $C(I+\Delta I)-I = C(I+\Delta I)-C(I)+C(I)-I = C(I+\Delta I)-C(I)+\delta I$. This error is the sum of two components: compression error of the original image (δI) and the difference between the compression versions of demosaised and original image ($C(I+\Delta I)-C(I)$). The error variance will then be:

$$\Omega^2 = E((C(I+\Delta I)-C(I))^2) + E(\delta^2 I) + 2E((C(I+\Delta I)-C(I))\delta I) \tag{3.4}$$

The first term is $f(\zeta^2)$, where $\zeta^2 = E(\delta^2 I)$ is demosaicing error variance. Expanding $f(\zeta^2)$ using the Taylor series and preserving only the linear term gives $f(\zeta^2) \approx \alpha^2\zeta^2$, where α depends on compression method and bitrate. Similarly, the third term in (3.4) is approximately $\alpha E(\Delta I \delta I) = \rho_\alpha \alpha \zeta \xi_s$. Here, $\xi_s^2 = \Delta^2 I$ are compression and demosaicing error variances, and ρ is the correlation coefficient between the two errors. The simplified total error variance for the conventional chain is:

$$\Omega^2 = \alpha^2\zeta^2 + \xi_s^2 + 2\rho_\alpha\alpha\zeta\xi_s \tag{3.5}$$

Similarly, for the alternative processing chain, the error is expressed as the sum of the demosaicing error (ΔI) and the difference between demosaiced versions of the compressed and original CFA ($D(\bar{I}+\delta\bar{I}-D(\bar{I}))$). The total error variance is

$$\Psi^2 = E((D(\bar{I}+\delta\bar{I}-D(\bar{I})))^2) + E(\Delta^2 I) + 2E(D(\bar{I}+\delta\bar{I}-D(\bar{I}))\Delta^2 I) \tag{3.6}$$

Similarly to 3.4, expanding the first and third terms in the Taylor series gives $\beta^2\xi_o^2$ and $\rho_\beta\beta\zeta\xi_o$. Here, parameter β depends on the demosaicing method, ξ_o^2 is the compression error for the CFA image and ρ_β is the correlation coefficient between compression and demosaicing errors. Thus, the total error variance for the alternative processing chain is:

$$\Psi^2 = \beta\xi_o^2 + \xi_o^2 + 2\rho_\beta\beta\zeta\xi_o \tag{3.7}$$

It has been observed [103] that ρ_α and ρ_β do not exceed 0.1 in amplitude on average. The only exception is simple bilinear demosaicing, which produces high demosaicing errors and, thus, higher ρ_α and ρ_β. Therefore, the last terms in (3.5) and (3.7) could be dropped.

The conventional and alternative processing chains could be compared by their error variances:

$$\Omega^2 - \Psi^2 = (\xi_s^2 - \beta^2\xi_o^2) + (\alpha^2 - 1)\zeta^2. \tag{3.8}$$

The superiority of one processing chain over another depends on $\xi_s^2 - \beta^2\xi_o^2$ and α^2. The detailed derivation of compression errors for both chains could be found in [103]. Empirically it was shown that α^2 is reasonably close to 1 for JPEG2000 for CR up to about 30, while for JPEG it is always below 1. The parameter β^2 is close to 1 for most of the demosaicing methods, except the bilinear, where $\beta^2 < 1$. The ratio ξ_s^2/ξ_o^2 depends on relative coding gains of both chains. It should account for the higher interpixel correlation for full color images which increase the coding gain of the conventional chain. On the other hand, the full color image has three times as many pixels as CFA, which decreases the coding gain.

The important conclusions from the theoretical investigation in [103] are the following. For simple compression of CFA (as grayscale) and the same demosaicing methods (except bilinear), the alternative chain is superior to the conventional one up to CRs 5-15 when JPEG is used as the compression method. In the case of JPEG2000, the alternative chain is superior up to CRs 15-30. These are the CRs commonly used by DSCs. With more advanced compression (de-interleaving, color decorrelation, etc.) and demosaicing techniques, it is possible to increase the superiority of break point CR for the alternative processing chain.

3.5.2 Compression methods

Let us consider different techniques for lossy CFA compression according to the generalized framework (Figure 3.1). In contrast to lossless compression, a wider range of techniques could be accessed, since losslessness is not required.

Color plane decorrelation using estimated color differences was studied in [40]. The missing G pixels in R/B positions ($\hat{G}$) were estimated by a simple linear 2D low-pass filter. Color differences are formulated as: $Rd = (R - \hat{G})/2 + 128$, $Bd = (B - \hat{G})/2 + 128$. The division by 2 is used to produce the same bit depth for the differences as for the original data.

Conversion from RGB color space to YCbCr is a common approach to decorrelate color planes. A similar conversion can be carried out on the BCFA. Due to incomplete information in the BCFA the conversion is performed on a block of four pixels: two G, one B and one R (Figure 3.14A), using the following equation utilizing incompleteness of data [97]:

$$\begin{bmatrix} Y^{ul} \\ Y^{lr} \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} a_1 & 0 & a_2 & a_3 \\ 0 & a_1 & a_2 & a_3 \\ a_4 & a_4 & a_5 & a_6 \\ a_7 & a_7 & a_8 & a_5 \end{bmatrix} \begin{bmatrix} G^{ul} \\ G^{lr} \\ B \\ R \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 128 \\ 128 \end{bmatrix} \tag{3.9}$$

where $a_1 = 128.55$, $a_2 = 24.97$, $a_3 = 65.48$, $a_4 = -37.1$, $a_5 = 112$, $a_6 = -37.8$, $a_7 = -46.9$, $a_8 = -18.21$. As can be seen, a_4 and a_7 are half of the standard coefficients of RGB to YCbCr conversion and the others are the same, G^{ul}, G^{lr}, R and B are assumed to be from the range $\{0, 1\}$. This is a rather popular color conversion scheme for BCFA used in a number of publications.

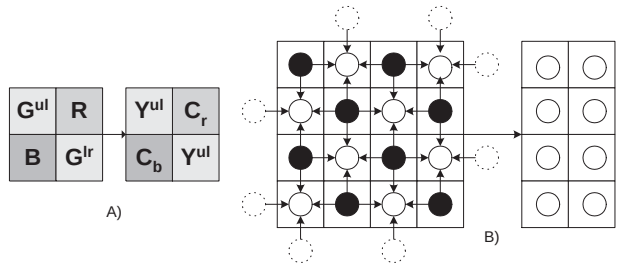

Figure 3.14: Block of 4 pixels for color space conversion (A). Optimized structure conversion (B), black - original pixels, white - filtered pixels, dashed - 'virtual' for filtering.

The quincunx G plane could be transformed into a rectangular grid by one of the methods in Section 3.3.1, or their derivatives. The merge method with low pass pre-filtering was proposed in [92]. Pre-filtering reduces the frequency content in the direction opposite to the merge direction aimed at facilitating compression.

The separation technique with pre-filtering using a non-separable diamond filter was proposed in [92]. Alternatively, separation with a reversible 2D low-pass filter was utilized in [194]. The rotation was used in methods [97, 35]. The "optimized stricture conversion" technique has been proposed in [194]. It uses pre-filtering together with placing G pixels in new positions (Figure 3.14B).

The prediction method for lossy compression has been proposed in [P3]. Our algorithm can be described in the following steps:

1. Convert a color format from RGB to YCbCr using 3.9.
2. Split luminance plane into odd and even indexed sub-planes.
3. Compress odd indexed luminance and both chrominance planes.
4. Decompress an odd indexed luminance plane.
5. Predict pixels of even indexed luminance plane from the pixels of decompressed odd indexed. This is done to ensure both decoder and encoder use the same reference (Section 3.1).
6. Compress the prediction errors.

In the decoder, all four sub-planes are decoded. The decoded odd luminance sub-plane is used to predict the even luminance sub-plane. The decoded difference is added to the predicted sub-plane in order to obtain the real one. If needed, CFA data could be converted back from YCbCr to RGB color space. The decoded CFA data are used to reconstruct the full color image. The simplified version of this algorithm can be found in [P3].

It should be noted that the reconstruction of the full color image of reduced size demosaicing is possible in the case of the prediction method for the G plane. In this case, the R, B, and odd indexed G planes could be used directly to compose the RGB image. This procedure does not require any additional calculations. The size of the reconstructed image in this case is a quarter of the image obtained during full reconstruction. Here, the the decompression of even indexed G sub-plane, prediction and, the relatively expensive, demosaicing operation are not required.

The subband coding technique for BCFA compression was used in [178]. The JPEG2000 as a compression algorithm was studied in [113]. Compression using vector quantization approaches [101, 12, 13], or spatial domain DPCM and vector quantization [21] have also been investigated. Most other algorithms are based on the JPEG compression method. The reason is that this method is the basis for lossy compression of full color images in DSC.

Application of the standard JPEG algorithm to compress separated color planes would be inefficient. The real distance between pixels in separated sub-lanes is twice that of the adjacent pixels in CFA, which causes a lower spatial correlation between the neighboring pixels and, as a result, higher frequency components. Standard JPEG quantization tables have embedded some sort of pre-filtering, which suppresses high frequencies. Due to this, the use of standard JPEG quantization tables will result in oversmoothing. This may affect compression results. In [P3], we have proposed to evaluate different de-interleaving methods with a compression technique that uses a uniform quantization of the AC DCT coefficients. We use a compression algorithm which is very similar to the "traditional" JPEG, which uses 8x8 block DCT, uniform quantization of DCT coefficients, zigzag scanning, run-length coding and an adaptive arithmetic coder at the final encoding stage.

3.5.3 Experimental results

Typical curves of PSNR comparison of the conventional and alternative image compression schemes are presented in Figure 3.15. For low CR, the alternative chain is superior to the conventional one. As CR rises, the alternative chain loses its superiority. Such behavior is well predicted by theory (Section 3.5.1). However, at very high CRs, the alternative chain could again become superior to the conventional. This is not explained by the existing theory, mostly due to simplification assumptions. The main reason, in our opinion, is that the amount of data to be compressed is three times less for the alternative chain. This is crucial for very high CRs.

In contrast to lossless compression, the appropriate method for G plane transformation significantly affects the performance of the compression algorithm. We have compared the performance of five structure separation based methods. These methods are:

1. Method 1: Only de-interleave is used, thus RGB planes are compressed. Similar

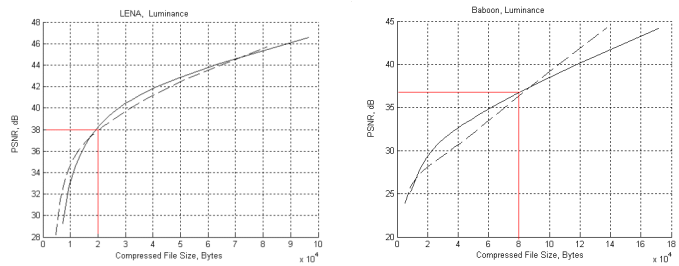

Figure 3.15: Conventional vs. alternative image compression chain. Typical PSNR curves.

to method in [113].

2. Method 2: The modified color space conversion and de-interleaving are used.

3. Method 3: The same as Method 2 except for the use of pre-filtering prior to structural transformation to limit aliasing. The method is identical to the one described in [92].

4. Method 4: The proposed prediction method.

5. Method 5: The proposed method without color space conversion. This has been motivated by the results in [194] where it has been shown that using compression in RGB, it is possible to obtain a gain in quality at low compression rates.

The results are presented in Figures 5-7 in [P3]. Our method significantly outperforms others at moderate and high completion rates. However, at low compression rates, it does not perform as well as methods that compress the RGB components directly. This can be attributed to losses incurred during forward and backward color space transformation. In contrast, a variant of the proposed algorithm without color space conversion (Method 5) is superior to other methods at high bitrates. This proves that the use of prediction for eliminating redundancy and improving coding efficiency is viable. It should be noted that some additional gains could be obtained by using more sophisticated prediction algorithms than the simple median-based prediction (Sec. II-C) [P3] used in our algorithm.

The starting PSNR value is approximately equal for all methods and is limited by the demosaicing algorithm used. After decompression and reconstruction of the Bayer CFA, practically any demosaicing method can be applied for color plane reconstruction. The combination of the proposed algorithm with the advanced demosaicing methods [65] at the decoder will result in a significant improvement in performance for small and moderate CRs (Figure 3.16). As CR increases, the performance gain decreases rapidly. At very low bitrates, a simple bilinear demosaicing approach performs better. This is mainly because sophisticated demosaicing methods are highly sensitive to loss of details. In contrast, bilinear interpolation smoothes the image and works as post-filtering.

The visual image quality is demostrated in Figures 3.17 - 3.19. At small CR (Figure 3.17), the artifacts of simple bilinear interpolation (zipper effects) are visible for both alternative and conventional chains. The image after the alternative chain looks slightly

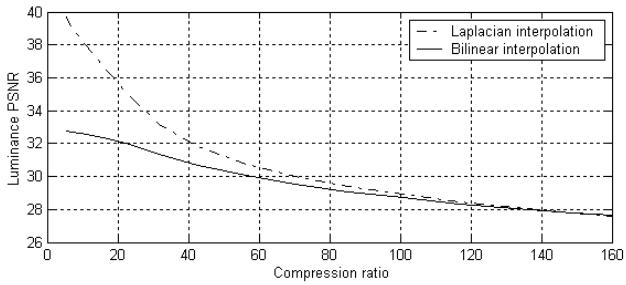

Figure 3.16: Compression algorithm performance with different demosaicing methods

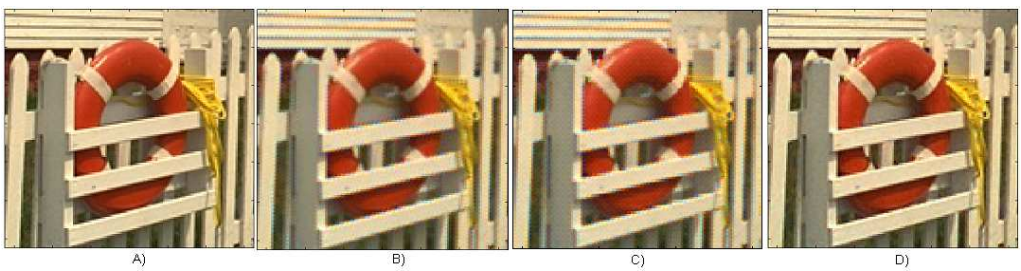

Figure 3.17: Compression results. Original scene (A), Conventional (bilinear+JPEG) CR=12, PSNR=32.2dB (B) Alternative (JPEG+bilinear) CR=12, PSNR=32,6 dB (C) Alternative (JPEG+Laplacian) CR=12, PSNR= 37,9 dB

sharper. The PSNR values are close. The use of the laplacian method for demosaicing in the alternative chain significantly improves both visual quality and PSNR values. The impact of demosaicing on subsequent compression was studied in [41]. It was shown that the more sophisticated demosaicing method produces more details in the luminance plane, and thus increases the variance of DCT coefficients, which does not facilitate compression. On the other hand, the variance of DCT coefficients in chrominance is decreased due to reduced color artifacts.

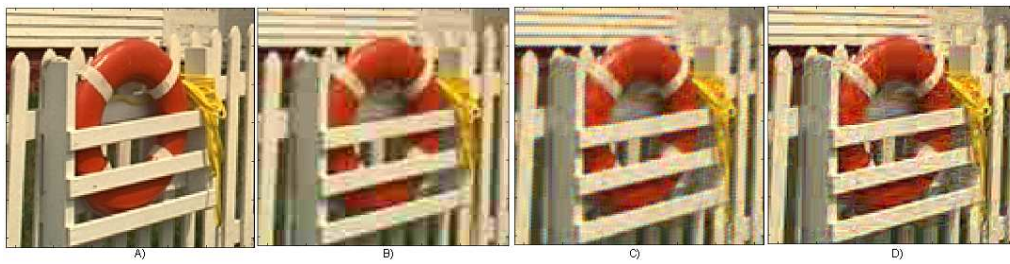

Figure 3.18: Compression results. Original scene (A), Conventional (bilinear+JPEG) CR=70, PSNR=28.9dB (B) Alternative (JPEG+bilinear) CR=70, PSNR=29,5 dB (C) Alternative (JPEG+Laplacian) CR=70, PSNR= 29,8 dB

At moderate CRs (Figure 3.18) the artifacts of bilinear demosaicing are not visible in the conventional chain. They have been suppressed during compression. However, blockiness artifacts become visible. In the alternative chain, blockiness is not observed. The laplacian demosaicing makes the image slightly sharper, with fewer artifacts.

Figure 3.19: Compression results. Original scene (A), Conventional (bilinear+JPEG) CR=125, PSNR=23dB (B) Alternative (JPEG+bilinear) CR= 125, PSNR=30,8 dB.

At very high CRs (Figure 3.19), the conventional chain suffers from strong blocking artifacts. The alternative chain provides significantly better visual results and is preferable in terms of PSNR.

Overall, we can conclude that the alternative IPP is superior for low CRs. The total improvement can be very significant (up to 7dB in Figure 3.16). This is mostly due to the usage of more sophisticated demosaicing (image restoration) algorithms. At very high CRs, the alternative chain is also superior. Here, the reduced amount of data to be compressed plays a crucial role. This make alternative processing feasible for applications targeting the highest image quality or a smaller bitrate. Moreover, by developing better

methods for de-interleaving and decorrelating the BCFA pattern, the alternative chain could be made superior for the whole range of CRs.

3.6 Lossy Compression of Noisy Raw Data

In this section, methods suitable for practical lossy compression of raw data are reviewed. Similarly to near-lossless compression (Section 3.4), there are two solutions: to non-linearly scale raw data prior the compression, or to modify the compression method by using adaptive error quantization.

The lossy raw data compression using the first approach was studied in [142, 114]. It is proposed first to quantize pixel values with non-linear LUT implementing a curve similar to that shown in Figure 3.10. Later, the quantized image is compressed by 12 bit JPEG.

Similarly as in Section 3.4, one can modify the block-based method to adaptively select quantization values for each block based on its mean value. Both approaches above rely on GC which is masking the errors. In this section, we propose the additional approach. Instead of GC, it relies on the level of the signal-dependent (photon-counting) component of the noise. The considering of only shot-noise is feasible since it is the dominating source of noise in modern cameras. First, lossy compression of noisy images is presented. This is needed to show how the quantization parameters for our method are derived. Then, the proposed method is described in detail.

3.6.1 Noisy image compression

Let us consider the sequence of operations compression/decompression and denoising as presented in Figure 3.20 [142]. Let $I^o_{i,j}$,$I^n_{i,j}$, $I^{dec}_{i,j}$ and $I^{den}_{i,j}$ be original (noisy free), noisy, decompressed and denoised images, respectively. Then $PSNRn = PSNR(I^n_{i,j}, I^{dec}_{i,j})$, $PSNRo = PSNR(I^o_{i,j}, I^{dec}_{i,j})$ and $PSNRd = PSNR(I^o_{i,j}, I^{den}_{i,j})$.

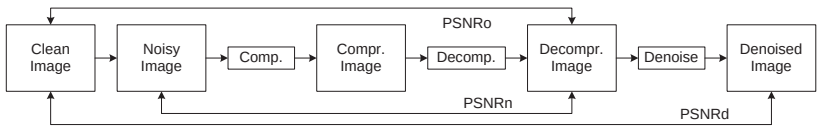

Figure 3.20: Noisy image compression

The typical dependencies of $PSNRn$, $PSNRo$ and $PSNRd$ from bitrate are presented in Figure 3.21. As expected, $PSNRn$ constantly decreases with a reduction of bpp (or, equivalently, with an increase of CR).

The dependency of $PSNRo$ on bpp has a specific behavior. For rather small CRs, the quality of the compressed image can be even better in comparison to the quality of the original noisy image. It has an obvious maximum at some bpp value. This is because under certain conditions, the main "losses" in lossy compression of noisy images relate to noise reduction. In other words, lossy compression possesses the useful ability of partial noise suppression and it can be considered as a specific pre-filter. When different compression methods are used, the positions of peaks approximately coincide, only the

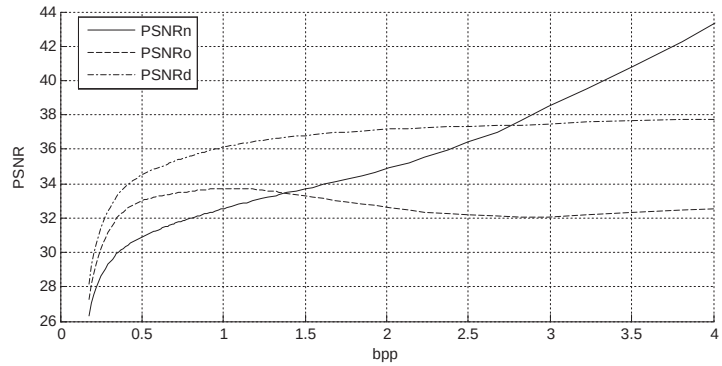

Figure 3.21: $PSNRn$, $PSNRo$ and $PSNRd$ from bitrate

absolute $PSNRo$ value differs at this point. The bpp value that corresponds to the maximum is called the optimal operation point (OOP) [114].

For compression methods based on DCT, CR and bpp depend upon image properties and the chosen quantization step QS. As shown in [114], to reach OOP it is possible to make QS bounded to the Gaussian additive noise standard deviation σ. In particular, in the case of compression without pre-processing and post-filtering, the optimal QS=4.5σ. If compression is performed after pre-filtering, then the optimal QS is of the order (1.3 .. 1.5)σ_{res}, where σ_{res} is the standard deviation of residual noise. Finally, if post-processing (post-filtering) is done after lossy compression, then the optimal QS should be about 1 .. 1.3σ.

The $PSNRd$ very slowly decreases to a bpp that approximately corresponds to a QS equal to σ. At this point, $PSNRd$ is only about 0.3dB poorer, compared to the case of lossless compression.

Thus, one can compress noisy image with QSs smaller than standard deviation σ, with a minor decrease in the effectiveness of posterior noisy reduction. On the other hand, if posterior noise suppression is not assumed, selection of QS=4.5σ guarantees the highest output $PSNRo$ value. The above is true for compression methods based on orthogonal transforms with quantization of the orthogonal transform coefficients [114].

3.6.2 Proposed method

In the case of Poisson noise, a modification of such an approach is possible [P4] by taking into account the fact that noise variance is strictly dependent upon true image values. Taking only the signal-dependent Poisson part of (2.7) one obtains: $\sigma^2(m,n) = bx(m,n)$, where b depends on the sensor model, exposure time and used analog gain.

Thus, the QS should be different for different image regions (blocks). If one divides an image into small blocks of fixed size it is reasonable to assume that in each block one is dealing with a homogeneous region. As an example, an 8x8 block size could be considered as a homogeneous region, since the resolution of modern cameras is often more than several million pixels. The variance for the image homogeneous region σ_l^2 could be considered as a constant, and approximately equal to the local mean $\bar{x}$ multiplied by the sensor and exposure dependent parameter b: $\sigma_l^2 = b\bar{x}$. It should be noted that parameter

b could be determined in advance for each sensor type and picture taken conditions [54].

Thus, the quantization step for each image block can be set as $QS_i = K\sigma_l^2$, where QS_i is the quantization step for the image block of size 8x8 pixels. K denotes the proportionality factor that bounds the quantization step with the local standard deviation. Since denoising is a practically obligatory procedure for image reconstruction, we propose to use K=1 for raw data compression, which correspond to 1σ (see Section 3.6.1).

The block diagram of the lossy coder for raw data which utilizes the above approach is presented in Figure 3.22.

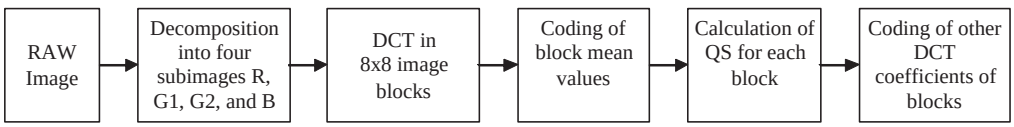

Figure 3.22: Coder for lossy compression of raw data.

The main impact of the proposed technique is that information about the quantization step in each block should be saved in the output bit-stream. For calculation of the block mean, it is enough to know the block DCT coefficient with indices (0,0). Thus, let us always quantize these coefficients with an equal and fixed $QS = 3$ (this is equivalent to QS for a block with a mean equal to 9). Using the quantized value of this coefficient, it is possible to determine the block mean value and to use it for calculation of QS_i, which applying for quantizing other DCT coefficients in a block.

At the image coding stage, the block coefficients with indices (0,0) are coded first for all blocks. Then, all other block coefficients are coded. This guarantees that to the time of other block coefficient coding the values of QS_i for their quantizing are already known. At the image decoding stage, the image block mean coefficients are decoded first for all blocks. After this, the values of QS_i used for quantizing other DCT coefficients of all blocks are calculated and these coefficients are decoded. In this manner, we avoid saving additional information about the quantization step in the compressed image bit-stream.

3.6.3 Experimental results

The experiments were carried out on artificial noisy-free 12 bit raw data. The methodology of generating such data can be found in [P4]. In experiments, each test image has been corrupted by Poisson noise with different levels.

Below, we consider two procedures of compression:

1. Lossless compression: Noisy image - Lossless compression - Denoising.

2. Lossy compression: Noisy image - The proposed compression method - Denoising.

For noise removal in both cases we applied a DCT-based spatially invariant hard thresholding filter [48] that produces good noise suppression and detail preservation for different noise types. The hard threshold has been set at equal to 2.6σ where the standard deviation is determined for each filter position.

The average results are presented in Table 3.1, where $PSNR_\Delta$ is PSNR calculated for the output images for the lossless and lossy compression schemes.

Table 3.1: Average bitrates and PSNR of Lossy and Lossless compression schemes for noisy raw data.

Input PNSR	Lossless		Lossy		$PSNR_\Delta$
PSNR, dB	bpp	PSNR, db	bpp	PSNR, dB	PSNR, dB
37.41	9.70	39.65	4.08	39.34	50.88
35.19	9.84	37.74	3.82	37.45	49.06
32.19	10.08	35.22	3.51	34.94	46.69
29.20	10.34	32.78	3.21	32.51	44.52
26.22	10.65	30.44	2.96	30.21	42.54

As can be seen, the selection of $K = 1$ for the proposed scheme has allowed us to obtain a decompressed image quality that is only 0.23-0.3 dB worse than for the lossless compression scheme. At the same time, the proposed scheme leads to 2.4 - 3.6 times greater CR. Another observation is that for lossless compression as noise level increases, the CR has become smaller, which is obvious since correlation between neighboring pixels is decreasing. For the proposed method, the amount of compression is directly related to the quality of the image. The CR is increasing as the noise level goes higher. At the same time, loss in quality still remains in the range 0.25 .. 0.3 dB, compared to the strictly lossless compression scheme. The analysis of $PSNR_\Delta$ values demonstrates that the decompressed images obtained using both schemes are practically visually undistinguishable. The CRs are also higher when compared to the near-lossless compression in Section 3.4.

The quantization value could be easily increased at the expense of an increased level of distortions. The next key point is about K = 4.5. At this point, the PSNR of the decompressed image is matching maximum. However, the fact that the efficiency of subsequent noise reduction is significantly reduced should also be taken into account. One could continue to increase the compression ratio even further, but the efficiency of any subsequent image reconstruction operations will be even further decreased. So, the goal of obtaining the maximum possible quality from the raw data at the final stage is no longer meaningful. However, in this case, the aim of obtaining the maximum possible compression can be achieved. That is, the quality of the reconstructed image could be significantly higher compared to the case when compression to the same level is applied to the reconstructed image. The evaluation of the possible gain in quality is a separate task that requires extensive study.

Part II

DCT block-based compression of images

Chapter 4

Block based DCT image compression

This chapter is dedicated to DCT-based image compression. First, image compression methods are reviewed. Different DCT-based techniques are summarized. Next, a method for compression of quantized DCT coefficients is presented. Then, applications of the proposed method to image compression, additional compression of JPEG images and three-dimensional DCT-based (3D-DCT) data compression are given.

4.1 Image compression techniques

The pixel-based representation of image data is unreasonably redundant. The aim of the compression is to remove the redundancy and allow more compact representation of data. The general lossy image compression scheme consists of [150] decorrelator, quantizer and entropy coding stages.

Depending on how the decorrelation is performed, image compression methods can be divided into two groups [18]: spatial and transform methods (Figure 1.2). In spatial-based methods, the current pixel is predicted from a context formed by neighboring and already encoded pixels. On a theoretical level, transform and predictive coding should asymptotically achieve the same coding gain. However, the predictive method can only deccorrelate data within pixel support of the context. The pixel support cannot be very big for the reason of complexity. These methods are most widely used for lossless and near-lossless compression. The spatial domain prediction can also be carried out for pieces of an image instead of a single pixel, such as in fractal coding.

In contrast, the transform-based method decorticates all the pixels to which the transform is applied. In theory, the best decorrelation can be achieved by the Karhunen-Loeve transform (KLT) [150], which is the best representation of a stochastic process as an infinite linear combination of orthogonal functions. The coefficients of the KLT are random variables and the expansion basis depends on the process. The KLT has limited applicability for the compression, because of its complexity and data dependency. Therefore, approximations of KLT by other linear transforms, like DCT [150], Fourier, wavelets, etc. are used.

In order to catch local irregularities (edges, textures, etc) transform is typically applied to blocks. Although this limits the decorrelation ability of the transform, the additional advantage is a faster processing.

Recently, image compression using DWT [121] has become popular. The DWT has good decorrelating and spatially-spectral localizing properties. Due to this, DWT is applied to the large tile of an image or even to the whole image. However, the realization of the DWT methods is more complex and resources-consuming comparing to block-based methods. Below, we focus on DCT-based compression techniques.

4.2 DCT-based compression techniques

The compression of images using DCT [7, 150] has been the field of intensive investigation during several past decades. For example, the DCT of 8x8 image blocks is the core of the JPEG [140], which has been the main image compression standard for more than 15 years.

Since, the JPEG standard was introduced, many improved versions of DCT-based coders have been proposed. These modifications have aimed at the improvement of compression efficiency, coder functionality or both.

First of all, the JPEG standard itself includes a progressive mode [140]. In this mode, the image is divided into several scans. The very first scan contains the full size image but with minimum quality and occupies very little space. Each subsequent scan gradually improves image quality as additional information is provided. This coding mode appears useful when transmission is through a slow public communication channel. However, the progressive mode itself does not improve compression efficiency, which remains the same as for the baseline JPEG.

The variant of JPEG with arithmetic coder instead of Huffman is also defined in JPEG specification. It is capable to produce 5-10 percent smaller file size. However, it is patented and not become widely used.

Besides modifications defined in the JPEG standard, a number of other options have been proposed. One way to improve the compression is to use optimized quantization tables. One of the methods is presented in [14]. The reported approach combines a statistical approach with a HVS response function to obtain a suitable quantization table for specific classes of images and specific viewing conditions. As a result, this approach provides improvement in PSNR up to 0.7 dB, together with improved visual appearance.

Additionally, it is possible to encode DCT coefficients in a more efficient way than the traditional zigzag plus run-length encoding. Vector quantization based compression of DCT coefficients was reported, e.g. in [147, 148]. The main drawback of such methods is that the optimal codebook depends on a content of the image.

One of the most efficient ways is to look at DCT coefficients as a decomposition into sub-bands. Later, coefficients are processed in a similar way as DWT coefficients. In [196], the authors apply the embedded zerotree method, which has come from a wavelet-based compression, to encode DCT coefficients. The described approach is 0.35-0.9 dB better than the JPEG with optimized quantization tables. In [196] even more efficient embedded subband codec based on zeroblock partitioning for the DCT image is presented. Nearly the same performance is provided by use of the morphological

representation of DCT coefficients with subsequent zerotree-like encoding[202]. The disadvantage of the above-described methods is that in order to be efficient they process the whole image at once. That is, first, the whole image should be transformed in the block-based manner, after which coefficients are reordered and encoded. This imposes the same memory requirements on the above methods as for DWT methods.

Such high CRs are explained by the fact that wavelet-like methods are able to efficiently catch the correlation between spectral coefficients in neighboring blocks. This correlation is relatively high due to the small block size.

However, in practice, the strongest correlation will be only between closely located blocks. Thus, if one could consider correlation only between neighboring blocks this could significantly reduce memory requirements for the price of a small lost in compression efficiency. For example, in [15] it is proposed to take into account correlation between spectral coefficients of image neighboring blocks. This allows a further decrease in the coded file size by 14 percent on the average. A more efficient method was presented in [93]. The DCT coefficient magnitudes of the current block are predicted from those of the previous four neighboring blocks, and only prediction errors are coded. The one block is chosen by the minimum mean square error technique. The indications of the chosen block and DCT signs are coded by arithmetic coding. The decompressed image quality is on average 1.5 dB better than that for JPEG under the same CR.

The quality of the decompressed image could be further improved by using deblocking [98, 118]. While these methods do not always lead to an increase in objective measures (like PSNR) the visual quality improvement can be significant.

Since the DCT has a lack of spatial-localization properties, the decorrelation performance is significantly poorer for the blocks containing edges. This could be improved by using variable block size [189, 36, 70] and partition schemes [145, 143]. The idea of these methods is to use a larger block size for the flat areas. The edges are processed with a small block size that allows good decorrelations. The decision about whether to choose a smaller block size could be made based on the entropy of DCT coefficients. Block sizes equal to 2^N are most often used due to the simplicity of implementation and availability of fast algorithms for DCT calculation. The variable block size DCT coding is adopted as a part of the H.264 video compression standard [189].

Further development of such an idea can be carried out by utilizing object-based coding, that is, the image is divided into arbitrarily shaped objects. The coding of arbitrarily shaped objects is possible by using the shape-adaptive DCT (SA-DCT) transform proposed in [167, 164]. The drawback of such methods is the need for good image analysis (edge detection and contouring) algorithms, which are often computationally expensive. The shape of the objects also needs to be coded. Shape (object)-based coding is adopted as a part of the MPEG4 video encoding standard [20].

A lot of work has been done to speed up the DCT procedure itself. One of the most attractive proposal is the approximations of DCT with a lifting schemes [179]. These approximations require only shifts and additions. Also, it has great complexity scalability.

The methods described above are able to significantly improve compression efficiency compared to JPEG. However, it is still problematic to reach the level of JPEG2000 in terms of compression and additional functionality.

In [144], the DCT-based method called "AGU", which combines a block size of 32x32, sophisticated bit-plane coding of the DCT coefficient and postfiltering, has been devel-

oped. The key feature of this method is an efficient encoding of magnitudes of DCT coefficients using the concept of bit-planes. The magnitudes of DCT coefficients are viewed as a set of bit-planes. The bits of each bit-pale are coded using a context adaptive binary coder. The context of each bit is determined by the values of neighboring bits, as well as bits from upper bit-planes and neighboring blocks. The AGU is capable of outperforming the JPEG2000 by up to 1.9 dB in terms of PSNR [144].

4.3 Proposed method for DCT coefficients compression

In this section, we present a method for compression of DCT coefficients. The proposed algorithm employs the method in [144], but with increased functionality and speed. The original bit-plane coding in AGU has several restrictions. The first limitation is the large block size that makes DCT transformation rather slow. The second is the relatively slow bit-plane coding of quantized DCT coefficients. In the original method, the scanning of bit-planes starts from the most significant one. In upper planes, however, the number of significant bits is very small and to find the context of the bit a large number of conditions should be checked. The third drawback is the lack of additional functionality like progressive coding, spatial and complexity scalability, and ROI coding.

In order to overcome the above limitations, a number of modifications are introduced. At first, the block size is reduced to the "traditional" 8x8. Second modification includes introducing a significance map (SM), where every bit indicates whether the DCT coefficient is significant or zero in the current position. The bit-plane coding is applied only for positions indicated by SM. The third modification is a method of scanning DCT coefficients capable of providing spatial scalability and progressive coding.

The encoding procedure is as follows. A DCT is applied to nonoverlapping blocks of size 8x8 pixels. Uniform quantization of the obtained coefficients is performed. As a result, an array of integer valued DCT coefficients is obtained.

There are two main components in a block DCT spectrum: DC and AC. The position of the DC coefficient in a block is fixed and its sign is always positive. Only magnitudes of the DC coefficients are stored. In contrast, for AC, in addition to magnitudes, the signs and the positions of non-zero coefficients should be stored. This array of AC coefficients can be split into magnitudes and signs. The magnitudes of DCT coefficients are coded in the following way. First, a single bit-plane indicating non-zero DCT coefficients, that are left after the quantization location, is coded. The main purpose of including SM is to reduce computational complexity and to speed up subsequent bit-plane coding by reducing the number of coded positions. Thus, the gain in speed due to SM usage is proportional to Nnz/N_{tot} [P7], where N_{nz} and N_{tot} are number of non-zero coefficients and total number of DCT AC coefficients in the block, respectively.

The SM coding is done for DCT coefficients at positions 1-63 (Figure 4.1). The DC is assumed to be always significant.

Let $S_{l,m}(i,j)$ define the bit value of a SM where the indexes $i,j = 1..8$ define the position in the block and $l = 1..L$, $m = 1..M$ define the block of an image (L, M are number of blocks in horizontal and vertical direction). The $S_{l,m}(i,j) = 1$ denotes that the DCT coefficient in this position is non-zero, while $S_{l,m}(i,j) = 0$ means the opposite. The coding of SM is done using context probability modeling using a set of conditions.

DC	1	4	9	16	25	36	49
2	3	5	10	17	26	37	50
6	7	8	11	18	27	38	51
12	13	14	15	19	28	39	52
20	21	22	23	24	29	40	53
30	31	32	33	34	35	41	54
42	43	44	45	46	47	48	55
56	57	58	59	60	61	62	63

Figure 4.1: Scan order of the coefficients. The dashed line shows groups of coefficients that share the same sets of probability models for SM coding.

These are checked starting from the most discriminative ones to the less discriminative. The most discriminative are the values at higher bit-planes in the same position and the values of bits in neighboring positions. The set of conditions used to classify bits of SM and determine their context can be found in [P7].

The employed bit-plane coding technique differs in a number of aspects from that used by AGU. The first difference is the scan order. The used scan order of the coefficients (Figure 4.1) is different from the traditional zigzag scan used in JPEG or from the line-by-line scan used in AGU. This is done in order to provide scalability and progressive coding in the following manner. The decoding can be stopped after a DC or AC coefficient with the numbers 3, 8, 15, 24, 35 and 48 that allow progressive coding. At the same time, an image can be reconstructed at scales: 1/8, 2/8, 3/8, 4/8, 5/8, 6/8 and 7/8. In order to do this, the encoding of the magnitude of the DCT coefficient is done immediately after the coefficient is encoded as significant. Stopping of the decoding after AC with the numbers 1, 5, 11, 19, 29, 41 and 51 provides the ability to perform reconstruction with non-equal scales in horizontal and vertical directions.

The AGU takes into account only binary conditions: at least one significant bit must be present within neighboring and already coded positions. In contrast, we found that the number of non-zero bits in the neighborhood is a good discriminative factor. This fact is taken into account and used in the conditions. In AGU, only the first row of the block uses a separate set of models. This is explained by the fact that statistical characteristics for this positions considerably differ from the rest of the block. Here we have gone further and use a different set of probability models for some group of coefficients (Figure 4.1) or even for a coefficient alone. Groups differ in the sense that for them only some bits in the context are available and can be used for context determination. This allows simplifying some conditions and, thus, speeding up the coding.

The modifications described above allow more accurate distribution of bits between models and, as a result, an increase in coding efficiency. By checking conditions $C1-C11$ (Table 2 [P7]), bits of SM are assigned to the number of probability models (Table 3

[P7]).

Bits referring to each probably model are encoded with a separate instance of the dynamic binary arithmetic coder [95]. The bits of SM referring to models 29, 61, 93, 125 and 157, have no significant coefficients in the neighborhood within a distance of 2 from them. There are no significant coefficients at this position in neighboring blocks. The probability of a significant bit for these models is very low. In our method, bits referring to the aforementioned models are not coded. That is, they are assumed to be always zero. While this leads to some additional losses in image quality, the gain in compression is higher.

The magnitude of the coefficient is encoded immediately after a new significant coefficient is found during SM coding. The coding begins with a bit-plane N and ends with the bit-plane 1, where N is a highest bit level that can contain non-zero bits:

$$N = \lceil \log_2(8R_{inp}/Q) \rceil \tag{4.1}$$

where R_{inp} is the range of the input data (0...255), $8R_{inp}$ is the range of the data after 2D DCT transform with 8x8 block size [134] and Q is a quantization factor.

Let $B^k_{l,m}(i,j)$ define the bit value of a DCT coefficient at bit-level $k = 1..N$. The indexes i, j, l, m are defined as earlier for SM.

In the case of a DCT coefficient coded as significant during SM coding, the 1 is subtracted from its magnitude at the encoder before the magnitude is encoded. The 1 is added back at the decoder after decoding magnitude. This additionally increases the efficiency of bit-plane coding for small-valued coefficients, which dominate among DCT coefficients [P7].

The conditions $C_{12} - C_{19}$ used to classify bits of the bit planes are presented in Table 4 in [P7]. Using the condition $C_{12} - C_{19}$, coding of magnitudes of the AC coefficients (1-63 in Figure 4.1) is performed in the following way. First, for every bit at the current bit-level condition C_{12} is checked. Its equality to 1 implies that this coefficient was already coded earlier at higher bit-levels. As a result, the current bit could be either 0 or 1 with approximately equal probabilities. In this case the current bit is assigned to probability model 0 independently of coefficient position. In other cases, depending on the coefficient's position and conditions, $C_{15} - C_{19}$ bits are classified to a number of probability models as summarized in Table 5 in [P7].

The DC coefficients are coded using a separate set of models. Similarly to AC coefficients, the bits of the DC coefficient are scanned from the highest bit-level N to level 1. The bits are assigned to one of the 16 probability models by checking condition C_{20} for every bit.

For removal of blocking artifacts the DCT-based filter proposed in [47] is used. It operates in a sliding 8x8 window with hard thresholding of DCT coefficients:

$$I^F_{i,j} = \frac{1}{|G||U|} \sum_{k \in G} \sum_{l \in U} ID_{k,l}[i-k+1, j-l+1],$$

where $I^F_{i,j}$ is an output of the filter for pixel i, j of the image,

$$ID_{k,l} = IDCT(D^{Tr}_{k,l}),$$

$$D^{Tr}_{k,l} = \begin{cases} D_{k,l}[i,j], & |D_{k,l}[i,j]| > Tr \\ 0, & |D_{k,l}[i,j]| \leq Tr \end{cases},$$

$D_{k,l}$ are the DCT coefficients of 8x8 image block with the left upper corner coordinates k, l, IDCT is the abbreviation for inverse DCT. The only filter input parameter is a threshold Tr used in zeroing of DCT spectral coefficients of processing image block. A "quasi-optimal" value of Tr is 2.6σ , where σ is the estimated standard deviation of the noise [49]. For quantization noise that is present in our case, $2.6\sigma \approx QS/2$. Note that the filter exploits the same transform as the compression, and with the same block size. For full post-filtering (PF) we propose to set $G = \{i-7, i-6, \ldots, i\}$ and $U = \{j-7, j-6, \ldots, j\}$. For fast post-filtering (FPF) we propose to set $G = \{i-5, i-3, i\}$ and $U = \{j-5, j-3, j\}$. $|G|$ and $|U|$ denote the number of elements in these sets. The FPF version performs approximately 7 times faster than PF.

ROI coding, in our method, could be easily realized due to the following factors. The first is the relatively small block size which allows quite accurate ROI determination. The second, is the postfilter which efficiently smoothes the sharp transition from non-ROI to ROI regions in the decoding stage.

A version of this method which also employs compression of signs of DCT coefficients is presented in Chapter 5.

4.3.1 Application to image compression

The coding efficiency and decoding image quality of the proposed method were analyzed on 512x512 grayscale images for different CRs. The results were compared to JPEG2000 [175] and AGU [144]. The ones we analyzed were selected for the following reasons. First, our coder could be viewed as an extension of the AGU coder. Second, it provides a lot of additional functionality, such as spatial and quality scalability, ROI, and progressive coding. The latest image compression standard that provides such functionality is the JPEG2000. As a quality criterion the PSNR was used.

The results, summarized in Table 4.1, show that the coding efficiency is close to that of JPEG2000, even with a fast version of post-filtering. The AGU coder performs the best for images with a large number of textural elements, such as "Barbara". In contrast, JPEG2000 is the best for images with large smooth areas, such as "Peppers".

Table 4.1: Compression results (PRNR vs. CR). Proposed FPF method with fast post-filtering, Proposed PF method with full postfiltering.

Method	CR = 8	CR=16	CR=32	CR=64
JPEG2000	36.44	32.95	30.05	27.57
AGU	36.96	33.49	30.67	28.18
Proposed FPF	36.40	32.80	29.80	27.15
Proposed PF	36.62	33.03	30.03	27.34

The proposed coder lies somewhere in between. It outperforms JPEG2000 for images with a large number of textural elements, but not as much as AGU. It performs as well as JPEG2000 for complex images such as "Baboon" and "Goldhill" and slightly worse for images with a large number of smooth regions, like "Lena" and "Peppers".

For smooth regions only a small number of coefficients are present in the block after the quantization. A bit-plane coding technique performs less effectively in this case. The

proposed coder performs slightly worse for large smooth areas due to the small block size. This is the price to pay for the speeding up of the coder.

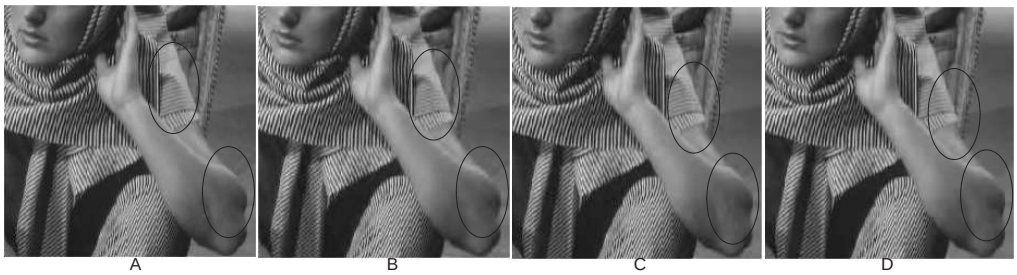

Figure 4.2: Part of the "Barbara" test image: original (A), compressed to 0.5 bpp by: JPEG2000, PSNR=32.87dB (B), AGU PSNR=34.65dB (C), proposed method with sign coding and PF PSNR=33.39dB (D). The places with the largest visual difference are indicated by ovals.

The visual quality of the compressed images is demonstrated in Figure 4.2. As can be seen, the proposed method, as well as AGU, is able to preserve texture details on the shoulder, while JPEG2000 destroys these details. Additionally, the edge between arm and background seems to be over-smoothed when compressed by JPEG2000.

A comparison with the version of the coder that utilizes coding of signs can be found in Section 5.3.

4.3.2 Additional lossless compression of JPEG images

The JPEG has been a commonly used method of compression for photographic images for more than 15 years. During the time the JPEG was used, an enormous number of images were shot. The archiving of these images requires a huge amount of memory. After JPEG was adopted as the image compression standard, more efficient lossy compression schemes were developed. Some of them were reviewed in Section 4.2.

One way to reduce the space occupied by existing JPEG images is to fully decompress them and re-compress by a new method. However, in this case, distortions introduced by JPEG are added to by distortions of the new compression method. Therefore, the quality of the transcoded image can be significantly poorer [141].

A more sophisticated way is to further lossessly compress existing JPEG images. That is, the JPEG image partially decompressed (only lossless entropy decoding is done) and quantized DCT coefficients are restored. Next, the DCT coefficients are compressed in a more efficient way than done by the original 2D zigzag and run-length coding.

There are a number of publications on this topic in the scientific literature. The authors propose to improve either the Huffman coder [168] or the zigzag scanning method [63]. More sophisticated methods were proposed in [141, 15, 170]. The method that can losslessly compress the JPEG file without partial decoding was developed in [154].

Additionally, there are a number of commercial and freeware programs that allow archiving of JPEG images images together with some lossless compression.

In this section, we evaluate the applicability of our coding technique for additional

lossless compression of JPEG images. This is feasible, since it is a DCT-based method with the same block size as the JPEG.

The five test images (512x512, grayscale "Lena", "Barbara", "Baboon", "Peppers", "GoldHill") were compressed by the standard JPEG algorithm with different quality parameters (Q=10..100). Next, images were partially decompressed, and quantized DCT coefficients were extracted. The proposed bit-plane coding method was applied to quantized coefficients. The encoding method was the same as described in section 4.3, except that the bits of SM referring to models 29, 61, 93, 125 and 157 were kept to allow fully lossless compression. Also, post-filtering was not used in this case. The achieved bit-rate savings are reported in Table 4.2. As can be seen, the proposed method improves the CR of JPEG by 12.7 - 29%.

Table 4.2: Additional lossless compression of JPEG images by proposed method, %.

Image/Q	10	20	30	50	75	90	100
Lena	28.1	19.9	17.0	14.4	13.6	15.1	13.1
Barbara	29.0	21.4	19.0	16.5	14.8	15.2	15.2
Baboon	22.3	16.4	14.3	13.1	12.7	14.3	19.9
Peppers	26.3	18.1	14.8	13.0	13.1	17.0	13.0
Goldhill	24.6	16.9	14.4	12.7	11.9	13.3	13.9
Average	26.06	18.54	15.90	13.94	13.22	14.98	15.02

There are a number of benchmarks for additional lossless compression of JPEG images [2, 1]. We tested our method for the benchmark images and the results were the following. For the "A10.jpg" image [2] additional compression was 13.5% and for the "DSCN3974.jpg" image [1] 16.6%. These are the fourth and fifth results, respectively, among the benchmarked methods.

It should be noted that the proposed method was not designed for additional compression of JPEG images. It was optimized for compression of a uniformly quantized DCT coefficient. The JPEG has a HVS-optimized quantization table and, thus, the statistics differ from that assumed by our method. The advantage of the proposed method is that, in addition to further compression, it could offer progressive scan, scalability, ROI coding, etc.

4.4 Application for 3D-DCT compression

In this section, the extension of the proposed method for compression of 3D-DCT coefficients is presented.

4.4.1 3D data compression

The collection and use of 3D data have increased in recent years due to the development of new measuring instruments and increases in the storage capacity and computational power. 3D representation of the data are required for many types of applications. The most widespread is a video, where the two first dimensions represent individual frames

and the third is the time. Other applications are multi-view images where the third dimension corresponds to the individual view. Computed tomography, magnetic resonance and microscopy, used for medical imaging and other diagnostic purposes, generate multiple slices of a single examination. Each slice represents a different cross-section of an examined subject.

Remote sensing using the imaging spectrometers, RADAR, SONAR and LIDAR also produce 3D volumetric data [122]. Each slice corresponds to an image on a different spectral/radio band, or at a different distance.

3D volumetric data can be stored as a set of 2D slices or as a mesh and a distance map. The latter method is not suitable for representing, for example, video. Therefore, here we consider only representation of 3D data using a set of slices or frames.

Due to its huge amount, 3D data require large storage space and time to transmit. This requires an advanced compression technique to keep the required storage space and transmission bandwidth reasonable. Each individual slice or frame can be processed as 2D images. The main challenge is the decorrelation of 3D data along the third dimension.

For the minor dissimilarity between individual frames, a simple difference technique could produce adequate results. When there is significant disparity, some techniques to align neighboring frames could be used. The most widely used is a block-matching technique (also called motion estimation in the case of video compression). The correct order of slices (e.g. in hyperspatial imaginary) could be important.

Block-matching is a computationally expensive procedure. Moreover, most algorithms deal only with horizontal/vertical transition of a block, and not with holding non-linear transformations e.g. zooming, rotation etc.

As an alternative to block-matching, the decorrelation properties of transforms could be used. This is done by applying transform along the third dimension: mainly DCT and wavelet transforms are in use [87, 158]. Also prediction techniques in the spatial domain have been used for a lossless compression [37].

The 3D-DCT is found to be a useful tool for many applications that require compression of multidimensional data, such as images and video compression. For video compression, 3D-DCT can be used as a low-complexity alternative to the traditional hybrid 2D-DCT and motion-estimation/compensation approach which is utilized in current video coding standards [159, 56, 19]. In addition to video compression, 3D-DCT was also successfully used for compression of still images [171, 102], hyper-spectral images [122, 3], multi-view images and videos [161, 161], etc.

4.4.2 3D-DCT coder

The block diagram of the 3D-DCT coder for multidimensional data is shown in Figure 4.3. First, 3D data are divided into groups of sequential frames (commonly, 2^N frames per group, where N is an integer). Then, each group of frames is partitioned into 3D cubes. Later on, 3D-DCT is applied to every cube. Due to the decorrelation property of the DCT, most of the energy is concentrated in a small number of coefficients. The quantization is used to remove some of the less relevant information for observer, and to facilitate further compression. Finally, the entropy encoding is performed. For the reconstruction of the data, the steps describedabove inverted and executed in the reverse order.

In order to be efficient, the entropy encoding is done separately for the different

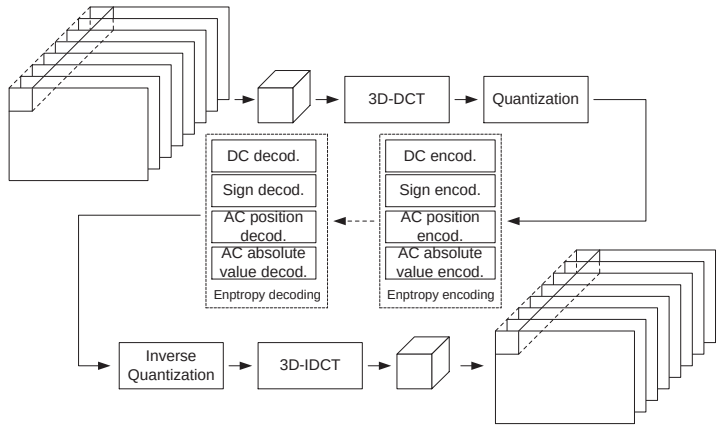

Figure 4.3: Block diagram of 3D-DCT coder for multidimensional data.

components of the DCT spectrum. The DC coefficient can be efficiently predicted from the neighboring blocks. Thus, only the difference between the predicted and the real value is encoded. The signs of DCT coefficients are assumed to be practically random and each sign is coded using one bit. The most significant part of the bitstream is utilized for the AC coefficient's positions and magnitude information. The AC coefficients are encoded via scanning and subsequent zero-run-length coding using variable-length codes. The purpose of scanning is to order quantized coefficients into a vector suitable for entropy coding.

A good scanning method should be simple and preferably predefined, from one side. On the other hand, it should efficiently group significant coefficients at the beginning of the sequence allowing zero-run-length coding to be applied in the most efficient way.

Traditionally, the 3D-DCT-based compression method uses a simple plane-by-plane 2D-zigzag scan [159], which come from the JPEG standard. An improved 3D version of the zigzag scan [198] allows, in some applications, slightly better compression compared to the 2D-zigzag. A more complex parabolic scan [171] and hyperboloid scan [122] have been reported. They are based on the assumption that significant coefficients tend to get grouped along the x, y and z axes of the cube. This assumption, however, holds only in the case of small dissimilarities between frames. An amplitude-based scan was proposed in [56].

All the methods mentioned above are non-adaptive to the coefficient's position. However, in practice, localization of significant coefficients within the cube can be very complex even in the case of a small dissimilarity between frames, as illustrated in Figure 2 in [P8]. Thus, often, a scanning method could not efficiently group significant coefficients. This leads to non-optimal coefficient ordering and shorter zero-runs, which significantly reduces the efficiency of the compression.

The authors in [19] studied the properties of the 3D DCT spectrum in the case of video sequences with motion. It was shown that, in this case, significant coefficients tend to be grouped along a plane whose orientation is defined by a dominant motion in the video sequence. Based on this study, two adaptive scanning methods were proposed. Despite the fact that adaptive methods provide better performance than non-adaptive

ones, they require the analyzing of some features of the input data (such as motion) or analysis of the distribution of coefficients within the cube. All these preliminary calculations require additional computations.

We can summarize that the 3D-DCT coefficients tend to concentrate in some areas of the cube. Depending on the nature of the input data these areas could be different and, additionally, varying from one cube to another. Thus, it is a challenging task to find a universal coding method that would be able to efficiently compress coefficients with different localizations within a cube.

4.4.3 Proposed compression method and experimental results

Here, the compression method for the coefficients of 3D-DCT in 8x8x8 blocks is presented. The proposed method is based on bit-plane coding of 3D-DCT coefficients and is an extension of the bit-plane coding for 2D-DCT introduced in Section 4.3.

Calculation of 3D-DCT in 8x8x8 blocks and quantization of obtained coefficients, results in cubes containing integer-valued DCT coefficients. Let us divide the cube of magnitudes of DCT coefficients into N bit-planes, where N is the number of the highest bit-plane that contains non-zero values [P8].

For the same reason as in Section 4.3, we introduce SM coding. That is, at the beginning, a single bit-plane is coded, where each bit indicates whether the DCT coefficient is significant at this position or not. Later, bit-plane coding is applied only at positions indicated by a SM. Similarly to the 2D case, the achieved speed up factor is proportional to N_{tot}/N_{nz}, where N_{nz} and N_{tot} are the number of non-zero coefficients and the total number of DCT coefficients, respectively.

First, the SM coding is performed. Let $S_{l,m}(x, y, z)$ define the bit value of a SM with the index $x, y, z = 0..7$ within the cube in a set of 8 frames with the index $l = 0..L-1$, $m = 0..M-1$ (L,M are number of cubes in horizontal and vertical direction).

The bits at positions with indexes $x = y = z = 0$ correspond to the DC coefficient of the cube and are assumed to be always significant and, thus, they are not coded. The bits at the rest of the positions are classified into a number of probability models using the flowchart presented in Figure 3 and the conditions $C1$-$C16$ in [P8].

The conditions $C1$-$C9$ separate bits that belong to the plane or the volume of the cube defined by the corresponding equation. The use of combinations of conditions allows us to identify a particular region inside the cube more precisely (Figure 4.4), for example, the $C1$ separate bits belonging to the front plane of the cube. A combination of $C1$ and $C2$, or $C1$ and $C3$ separate bits belonging to the first row and the first column of the front plane and so on. This is needed because for the bits located near the borders of the cube a smaller number of neighboring bits is available. As a result, the distribution of bits between models differs significantly.

The conditions $C10$-$C16$ form the context of the bit. The context depends on how many non-zero bits then are among neighboring and already coded bits within current and neighboring blocks.

When the DCT coefficient is significant, the 1 is subtracted from its magnitude at the encoder. This 1 is added back at the decoder after the decoding of magnitude. The bits belonging to each probability model are coded using the dynamic version of the binary arithmetic encoder [95].

After compression of SM, the coding of magnitude is started. The coding is done only

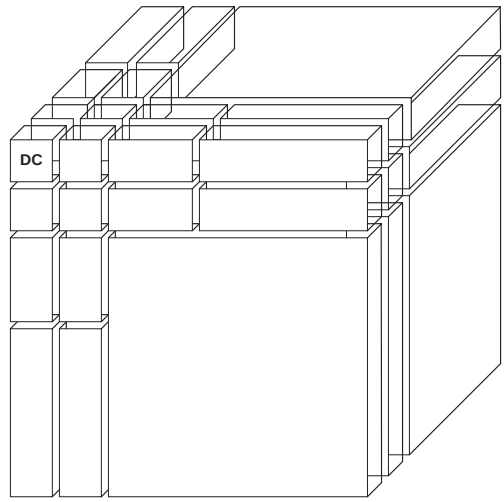

Figure 4.4: Division of the 3D DCT cube.

for positions indicated by the SM. The coding starts from the bit-plane N and ends with the bit-plane 1. Let $B^k_{l,m}(x, y, z)$ define the bit value of a DCT coefficient at bit-level $k = 1..N$, where N is the highest bit level. The indices x, y, z, l, m are defined as earlier for the SM.

The bits of every bit-plane are classified into a number of probability models using the flowchart presented in Figure 4 and the conditions $C17$-$C23$ in [P8]. The same as for conditions $C10$-$C19$, these form the context of the bit. Additionally, they utilize a number of non-zero bits in higher bit-planes. Similarly to the process of coding of SM, the bits belonging to every probability model are coded using the dynamic binary arithmetic encoder [95].

The coding efficiency of the proposed method was analyzed for compression of test video sequences in QCIF format. The four sequences were the Glasgow, Akiyo, Foreman and Carphone. The proposed method was compared to other coding methods: plane-by-plane 2D-zigzag scan [159], 3D-zigzag scan [198], hyperboloid-based scan [122] and magnitude-based scan [56]. The compression was done only for the luminance components. The sequences were processed by 8x8x8 blocks. All coefficients were uniformly quantized by the quantization factor (Q). The quantized DC coefficients and signs of AC coefficients were transferred to an output bit-stream without any compression (since this work is not focusing on encoding DC and signs). The quantized AC coefficients were compressed by each of the five methods. The average CRs obtained for all methods and quantization factors are presented in Table 4.3.

As can be seen, the proposed method increases CR 1.28-1.36 times. The amount of the increase mainly depends on the type of video sequence rather than on the CR. For the sequences with a fast stochastic motion, the DCT coefficients are widely distributed within the cube. The proposed method localizes such areas better, and, as a result, codes them more efficiently. For low-motion videos, most 3D-DCT coefficients are localized in a few front planes and could be quite well captured by a simple 2D-zigzag scan. For this reason, the reduction in bitrate is smaller, but still significant. The improvement

in frame-by-frame PSNR varies by 0.72-2.05 dB (1.56 dB on average) (see Figure 5 in [P8]).

Table 4.3: Compression results.

Method/CR	2D-Zigzag	3D-Zigzag	Hyper.scan	Abs. scan	Proposed
Q=8	10.30	10.35	9.917	9.370	13.67
Q=16	18.04	18.11	17.68	17.02	24.30
Q=32	34.15	34.70	34.12	33.35	47.15

To conclude, the proposed method can be used for compression of other than video types of multidimensional data. Also, it could be integrated with any existing 3D DCT-based codec. Large block sizes (16x16x16, 16x16x8, 32x32x32, 32x32x16, etc.) could be utilized for image data that do not require online processing. This will provide higher CRs for the price of additional computational resources. The method could provide complexity scalability in such a way that some conditions are excluded or added "on a fly". This will either improve or degrade CRs. The conditions $C10$, $C19$ and $C22$ take into account the correlation between neighboring cubes. If such a dependency is not desirable for some reason, the checking of these conditions could be skipped. This leads to an increase in compressed file size of about 3-5%.

Chapter 5

Sign coding for block based DCT compression

In this chapter, a rather new area of compressing signs of transform coefficients is presented. First, the problem to be considered and existing solutions for compressing signs of wavelet coefficients are described. Then, the proposed method for coding signs of DCT coefficients in block-based image compression is presented. Finally, the use of signs coding for the image coder from Section 4.3.1 is illustrated. This chapter is mainly based on publications [P6] and [P7].

5.1 Problem formulation

In transform-based compression, image energy is concentrated in a small number of coefficients, and compression is achieved by coding these coefficients. The energy of transform coefficients is restricted to having non-negative values. However, the coefficients themselves are defined by magnitude and sign. The energy compaction capability deals with magnitudes of coefficients, while the nature of the signs is undetermined.

The practical impossibility of the prediction of signs of transform coefficients is generally accepted. Shapiro [162], Said and Pearlman [155] argue that a sign of transform coefficient is a random variable which is equiprobable to be positive or negative. In most modern image and video compression standards and methods (e.g. JPEG [140], MPEG, H.264, SPIHT [155], JPEG2000 [175]), the encoded sign occupies 1 bit of memory in compressed data.

At the same time, the data of all coded signs of transform coefficients occupy a significant part of a total compressed image. We have calculated the percentage of compressed files used for signs in JPEG and AGU [144] coders. The results are presented in Table 1 in [P7]. As could be seen, for a relatively old and simple JPEG the percentage lies in the range from 10 to 18. It depends more on compression ratio than on image type. For more sophisticated methods, like AGU, this percentage can reach more than 24, and it is less dependent on the compression ratio. This is because highly adaptive methods more efficiently compress other information about the DCT spectrum such as coefficient locations and magnitudes.

Thus, signs can occupy a rather significant part of the bitstream, and the task of reducing the space occupied by signs is challenging. Moreover, while the performance of compression methods gets better, improving sign coding becomes more important.

Some attempts to predict signs of wavelet coefficients for wavelet-based image compression have been made recently [174]. An in-depth analysis of signs of wavelet transform coefficients was carried out in [38].

In wavelet transform, the following properties can be used to predict signs of coefficients: correlation along edges and correlation across edges.

Wavelet transforms are spatially-spectral with good localization properties. Thus, strong horizontal and vertical edges are well localized in the transform domain. There is a strong correlation of wavelet coefficients in the direction along the edge. Thus, neighboring coefficients along the edge provide valuable context information and are expected to have the same sign. This property is independent of the type of wavelet transform used.

There is also correlation across edges, which is directly affected by the structure of the high-pass wavelet filter. Some wavelets, like the Daubechies 9/7, are shaped similarly to the second order derivative of the Gaussian-like function. The latter is often used for edge detection. Thus, similarly to edge detection, it could be expected that the sign of the wavelet coefficient will change when crossing the edge. Therefore, there is strong negative correlation across edges for similarly-shaped wavelets. For wavelets with different structures, another sort of correlation exists.

By combining both along and across edge contexts via projection onto neighboring basis vectors, it is possible to correctly predict from 56-84% of signs of wavelet coefficients [38].

The authors of [38] go even further. They use context information to restore the value and sign of coefficients that was quantized to zero. The decoder also leaves these coefficients as zero. This is the best estimate in the least-square sense under the assumption that the sign is unknown, and can equally be either negative or positive. However, in some contexts, the magnitude and sign of a coefficient could be predicted with high accuracy. By using prediction of coefficients from a selected number of "good" contexts, it is possible to improve the PSNR value.

By utilizing sign coding, up to a 0.4 dB increment in PSNR can be achieved (0.15 on average). When coefficient extrapolation is also included, the gain can be up to 0.7 dB (0.31 on average).

Below we concentrate on coding signs of DCT coefficients, which has not yet been studied.

5.2 Proposed method

5.2.1 General idea

Let us consider block-based DCT image compression. The method for prediction of signs of DCT coefficients in the considered case was reported in [P6].

The primary assumption for the method is that values of pixels located at the borders of the neighboring blocks should be highly correlated. That is, at least pixels of the first row and the first column of a current block could be accurately predicted using values of pixels of already coded/decoded neighbor blocks of the image.

The pixel values in the block are a function of the corresponding DCT coefficients:

$$X_j = IDCT(\Phi(C_i), \Psi(C_i), \Omega(C_i)),$$

where $IDCT$ denotes inverse DCT transform, $\Phi(C_i)$, $\Psi(C_i)$ and $\Omega(C_i)$ are the functions which describe position, magnitude and signs of DCT coefficient C_i. Here, j is the pixel index inside a block and i is the pixel index of a non-zero DCT coefficient. When the magnitude and positions of DCT coefficients in a block are known, the only factor that affects pixel values is the sign assigned to each coefficient:

$$X_j = IDCT(\Omega(C_i)).$$

Thus, one can select such a combination of signs that minimizes the dissimilarity between pixels at the borders of the blocks (Figure 5.1). For the right combination of signs for the example in Figure 5.1a-d, the MSE was 84. For examples shown in Figure 5.1 e-h, g-h, i-j the MSE was 347, 4843, 787, respectively. Thus, even for the variant which differs from the original by a sign of two small-valued coefficients, the MSE is increased over factor four.

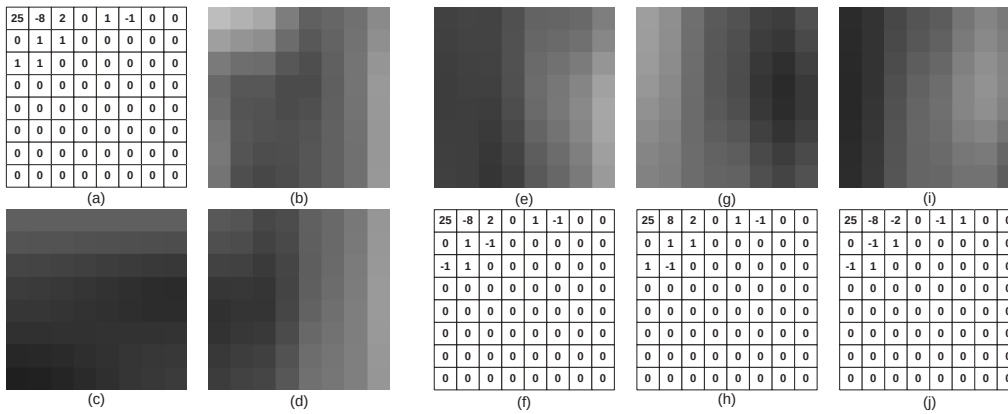

Figure 5.1: Image blocks: current block - (d); DCT coefficients of the current block after transform and uniform quantization (a), block located to north (b), block located to west (c), from current block. Blocks are generated using the same combinations of DCT coefficients (e-j). The magnitude of DCT coefficients and their position were kept the same. The signs were randomly selected for each coefficient. Pairs of blocks and their DCT coefficients (e-f), (g-h), (i-j).

Let us call the first row and the first column of the block in which we aim to find signs as "test rows". Additionally, let us call prediction of test rows from the neighboring blocks as "estimate of test rows".

The algorithm for encoding the signs can be summarized as follows:

---

1. Find the estimate of test rows

2. Find the combination of signs which minimizes MSE between test rows and test rows estimate

3. Encode a "guess or not" value for each sign

The last step is needed since the smallest value of MSE does not always correspond to the right combination of signs. Therefore, we cannot fully exclude information about signs from the coded bitstream. The "guess or not" value could be indicated with one bit for each sign. The number of correctly predicted sign is greater than the number of noncorrectly predicted. Thus, the entropy of such sequence is smaller than 1 and it can be efficiently compressed by binary arithmetic coding [95]. Let us consider each algorithm step in detail.

5.2.2 The search method for sign coding

The sign of each DCT coefficient can be either positive or negative. The total number of all possible combinations of signs is $2^{N_{nz}}$, where N_{nz} denotes the number of non-zero DCT coefficients in the block. For complex blocks and small quantization values, N_{nz} can reach several tens. An exhaustive search over all possible combinations is not applicable in real time. However, it provides an upper bound limit for the search method.

In more a practical approach, signs of each coefficient could be checked individually. Let us consider the objective function as MSE between "estimate of test row" and "test rows" obtained for a particular combination of signs from another. This objective function was found to have a large number of local minima. Therefore, the order in which signs of DCT coefficients inside a single block is predicted is important for good performance of the prediction.

The simplest way is to check signs of the coefficients in row-wise or column-wise order. Another possibility is to check DCT coefficients based on their magnitude. That is, coefficients with higher magnitudes are checked first. The reason for this is that the probability of false sign prediction is lower for the coefficients of higher magnitude and, thus, there is a smaller chance to fall into the local minima. The third possibility is to check DCT coefficients based on their distance from the beginning of the block (to DC coefficient). The reason for this is that the probability of prediction error is usually lower for low frequency coefficients.

These three search methods have been compared in [P6]. The magnitude-based search was slightly better compared to the others. Here we compare the magnitude-based search with the full search. To keep calculation time reasonable, we perform comparison only for blocks with less than 15 significant coefficients (32768 combinations). The average results for five test images (Lena, Barbara, Baboon, Pepper,Goldhill) are presented in Table 5.1.

It can be seen that the magnitude-based search was superior to the full search method. The possible explanation is that the selected criteria (similarity between "test rows" and "test rows estimate"), is more optimal for large value coefficients. The signs of lower value coefficients that minimize MSE do not always correspond to the right combination of signs.

Table 5.1: The percentage of non-correctly predicted signs

Method	QS = 10	QS = 20	QS = 40
Magnitude search	34.50	29.08	25.01
Full search	35.06	30.21	25.82

5.2.3 Test row estimate

The quality of prediction of signs by the proposed method depends on the accuracy of the test rows estimates. In [P6] it was shown that, if one knows the test rows exactly, the percentage of falsely predicted signs decreases by 5-7 times. Therefore, it is important to improve the test rows estimates in order to improve prediction of the signs.

The simplest way is to use, as test rows estimate, the closest row and column of neighboring blocks located to the north and to the west from a currently processed block.

A more sophisticated method was proposed in [P6]. Each pixel of the test rows is predicted using a context-adaptive predictor similar to that described in [60].

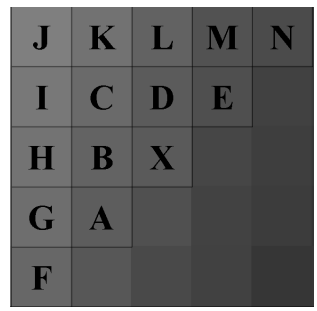

Figure 5.2: Prediction context.

The context of a pixel X is presented in Figure 5.2. According to the place of the pixel in the block (in the predicted row or predicted column) all the values of pixels A-N may be known or only a part of them. The value of the pixel X is predicted by this known part of values of pixels A-N.

The prediction relies on the hypothesis that the value of X is to the same degree similar to the pixel A as the content (neighborhood) of the pixel X (the pixels A-D) is similar to the content of the pixel A (pixels F, G, H, B).

The difference between contexts of the pixels A-E and the context of the pixel to be predicted X is calculated as:

$$L_A = ((F-A)^2 + (G-B)^2 + (H-C)^2 + (B-D)^2)/4$$

$$L_B = ((G-A)^2 + (H-B)^2 + (I-C)^2 + (C-D)^2 + (D-E)^2)/5$$

$$L_C = ((H-A)^2 + (I-B)^2 + (J-C)^2 + (K-D)^2 + (L-E)^2)/5$$

$$L_D = ((B-A)^2 + (C-B)^2 + (K-C)^2 + (L-D)^2 + (M-E)^2)/5$$

$$L_E = ((D-B)^2 + (L-C)^2 + (M-D)^2 + (N-E)^2)/4$$

If the values of L_A, L_B, L_C, L_D or L_E are equal to zero, we assign a unity value to it. The prediction P_X is a weighted sum of the nearest pixels A-E:

$$P_X = \frac{(A/L_A + B/L_B + C/L_C + D/L_{D+}E/L_E)}{(1/L_A + 1/L_B + 1/L_C + 1/L_D + 1/L_E)}$$

Table 5.2: MSE between true and estimated test rows

Method	QS = 10	QS = 20	QS = 40
Neighbor	219.3	239.2	289.1
Prediction	167.2	185.7	236.1

The results of comparison of the proposed predication technique with simple the nearest-neighbor technique for five test images (Lena, Barbara, Baboon, Pepper, Goldhill) are given in Table 5.2. As can be seen, the proposed method always provides a significantly smaller estimation error for test rows.

5.2.4 Coding the signs

The "guess or not value" is indicated using a single bit. Assuming that the number of correctly predicted signs is significantly greater, the entropy of such a sequence is smaller than 1. Thus, it can be efficiently compressed by binary arithmetic coding [95].

The coding performance could be even further improved by taking into account the contexts in which the current sign occurs. There are a number of discriminative factors observed in practice that could determine the context. The probability of a false prediction is smaller for low-frequency DCT coefficients ($C1$). The probability of a prediction error is smaller for DCT coefficients with larger magnitudes ($C2$). If any signs have been falsely predicted in a current block, a probability of prediction error for the rest of the signs is larger (C3).

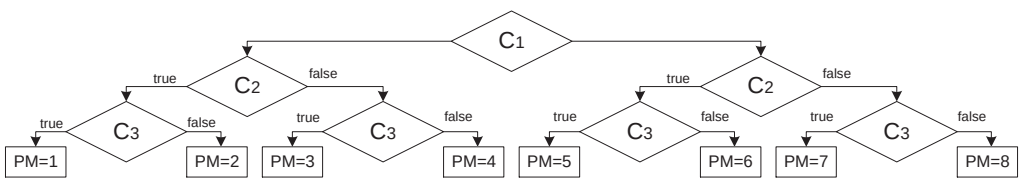

Figure 5.3: Classification of guess values using conditions C1-C3 (See text). PM-probability model number.

Simple context modeling for sign coding is presented in Figure 5.3. A more sophisticated method that uses some additional discriminating factors is presented in [P7].

5.3 Application for image compression

The proposed sign coding method was tested with the image coder presented in Section 4.3.1. The block diagrams of the encoder and decoder which support sign coding are presented in Figure 5.4 [P7].

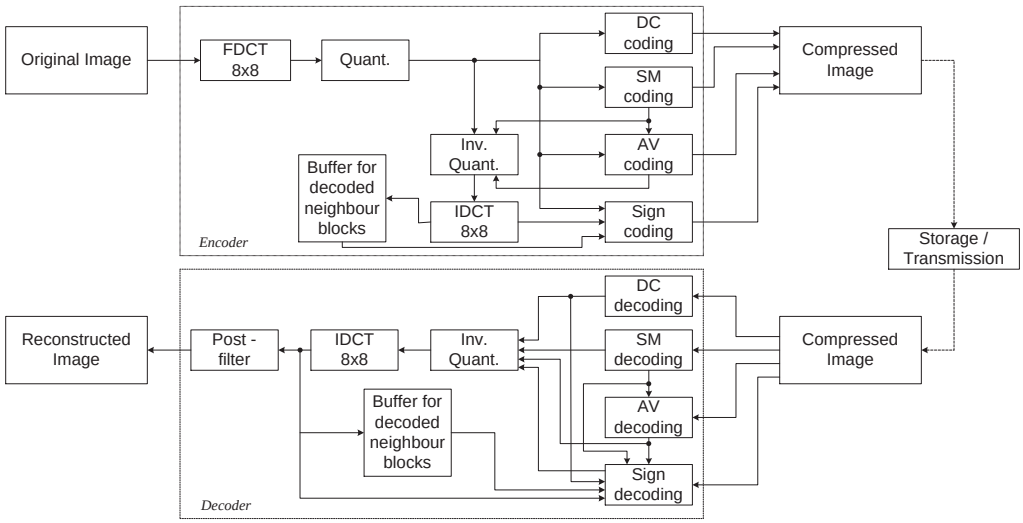

Figure 5.4: Block diagrams of the encoder and decoder.

A DCT is applied to non-overlapping blocks of size 8x8 pixels. Uniform quantization of the obtained coefficients is performed. All components of the DCT spectrum are encoded in the same manner as in Section 4.3.1. Additional blocks for sign coding are: dequantization and DCT transform and memory buffer for neighboring blocks.

The used algorithm for encoding the signs can be summarized as:

---

1. Find estimate of test rows (Section 5.2.3)
2. For each non-zero coefficient found after coding its magnitude:
   (a) Calculate test rows that correspond to positive and negative values of the sign
   (b) Find a variant with the smallest error and use it as a prediction of the sign
   (c) If prediction is correct, assign guess value to 0, otherwise to 1
   (d) Encode guess values using binary arithmetic coding (see [P6])

In [P6], a fast method for "test rows" calculation of a particular combinations of signs is proposed. The test rows could be found by calculating inverse DCT of a whole block. However, this is impractical and can be avoided. The number of pixels in test rows is far smaller than the total number of pixels in the block. Thus, it is reasonable to store values at the positions of test rows which correspond to the one unit DCT coefficient in every position within the block. The values of test rows that correspond to a particular

combination of DCT coefficients are found using linearity properties of DCT transform, such as multiplicativity and additivity. For an 8x8 block, the number of pixels in test rows is 15, while the total number of pixels in the block is 64. The memory requirement then is 15x63 = 945 numbers (the DC DCT coefficient is not counted).

The experimental results, for coder with and without sign coding, for the five test images (Lena, Barbara, Baboon, Pepper,Goldhill) are presented in Table 5.3. As can be seen, including sign compression improves PSNR values by about 0.18 dB in average for all CRs. The improvement is similar to that obtained for sign coding of wavelet coefficients [38]. Although the total improvement is not very significant, it could be increased by an improved test rows estimate.

Table 5.3: Compression results.

Method	CR = 8	CR = 16	CR = 32	CR = 64
No sign coding	36.61	33.03	30.02	27.33
With sign coding	36.77	33.21	30.22	27.52

Chapter 6

Concluding remarks

In this thesis the problem of image compression for digital cameras was addressed. Image acquisition and reconstruction with digital cameras was also studied. The trade-offs for conventional and alternative image formation chains were analyzed. Due to the limited computational and power recourses of a camera it is challenging to apply optimal reconstruction algorithms in the convectional chain. Thus, the quality of the images reconstructed by the camera is always compromised due to hardware constraints. The main advantage that the alternative image formation chain can provide is superior image quality by transferring of raw data to an external powerful PC. In our opinion, the key challenges for alternative image formation are efficient raw data compression methods. We argue that the storing of raw data can be beneficiary not only for applications requiring high quality imaging but also for everyday camera usage. Therefore, all types of compression methods (lossy, lossless and near-lossless) for raw data are required. All these types have been studied in this thesis.

We started by formulating raw data peculiarities. The influences of raw data peculiarities on different compression types were studied. Next, a generalized framework for raw data compression was proposed. In this framework, we aimed to utilize the already existing methods for image compression. Quality evaluation approaches for different types of compression have been described. We argued that for lossless and near-lossless compression, evaluation should be done with real data. This is supported by the completed experimental investigations.

Lossless, near-lossless and lossy compression of raw data was investigated. We reviewed existing approaches, as well as our own present methods. We demonstrated that for lossless compression different methods for transformation of the G plane from the quincunx to the rectangular grid produce similar results when applied to real raw data. We illustrate, both theoretically and experimentally, that advanced techniques for color decorrelation developed for artificial raw data often fail when applied to real raw data. For near-lossless compression we developed a method based on JPEG-LS. It allows a non-uniform encoding error, while guaranteeing a uniform error after image reconstruction.

In contrast, for lossy compression, the method for transformation of the G plane from the quincunx to the rectangular grid significantly affects the compression results. We proposed a method that predicts pixels of odd-indexed G pixels from even-indexed ones.

The aim of the method is to decorrelate G sub-planes and facilitate compression. The experiments confirm the high efficiency of the proposed method for raw data compression.

Real raw data are always contaminated by noise. Therefore, we considered noisy image data compression. The significant impact of the noisiness of the data is that for a rather wide range of CRs, the compression errors can be masked by noise. We proposed a lossy compression method for raw data that automatically selects quantization parameters based on the noisiness of the data. Because the signal-dependent part is often the main component of the noise, our algorithm adaptively selects quantization parameters for different areas of the raw data. The experiments confirm the high compression efficiency of the proposed method compared to lossless and near-lossless methods. An important property of our technique is that CR becomes higher (in contrast to lossless and near-lossless), as the noise level increases. At the same time, the loss in quality still remains small and constant compared to the strictly lossless compression scheme.

In addition to raw data compression, this thesis addresses the problem of image compression. According to our understanding, the lossless or near-lossless compression for images reconstructed by the camera IPP is rarely required. This is because for a camera reconstructed image, the quality is already compromised due to hardware limitations. Therefore, we concentrated on lossy image compression. We consider the block-based DCT compression as less demanding on memory resources.

We proposed an efficient method for encoding quantized DCT coefficients. It is based on bit-plane-based scanning of absolute values of DCT coefficients. The SM is encoded in the beginning to indicate the positions of non-zero DCT coefficients. The used scan order allows progressive coding and resolution scalability. The encoding technique itself can provide complexity scalability. The ROI coding could be realized due to the relatively small block size. The proposed technique can be used also for encoding with variable block size or SA-DCT.

The applications of the proposed technique for image compression, additional lossless compression of JPEG images and for 3D-DCT based compression were demonstrated. In image compression, the proposed technique combined with an 8x8 block-size DCT and post-filtering provides compression efficiency comparable to that of JPEG2000. For additional lossless compression of JPEG images, our technique allows 13-26% bitrate reduction. For 3D-DCT-based compression our method improved CR 1.28-1.36 times.

The signs of DCT coefficients are normally considered as a random variable and thus not coded. In modern methods, however, signs can occupy a significant part of the bitstream. In this thesis we propose a technique for predicating signs of DCT coefficients for block-based compression. The proposed method utilizes the fact that when the magnitudes and positions of all DCT coefficients are known, the signs are the only variables that influence the pixel values. We proposed an efficient way to predict signs by predicting pixel values on block borders from neighboring and previously encoded blocks. Integration of the proposed technique into a real image coder was demonstrated.

Bibliography

[1] Data Compression Programs. Website. `http://www.cs.fit.edu/~mmahoney/compression/`.

[2] Jpeg lossless image compression test. Website. `http://www.maximumcompression.com/data/jpg.php`.

[3] G. Abousleman, M. Marcellin, and B. Hunt. Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCM / DCT. *IEEE Trans. on Geoscience and Remote Sensing*, 33:26–34, Jan. 1995.

[4] J. Adams, K. Parulski, and K. Spaulding. Color Processing in Digital Cameras. *IEEE Micro*, 18:20–30, Nov./Dec. 1998.

[5] Adobe. Tiff: Specification for revision 6.0., 1992. `http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf`.

[6] Adobe. The AdobeRGB(1998) Specification, 2005. `http://www.adobe.com/digitalimag/pdfs/AdobeRGB1998.pdf`.

[7] N. Ahmed, K. R. Rao, and T. Natarajan. Discrete Cosine Transform. *IEEE Trans. on Computers*, 23:90–93, Jan. 1974.

[8] S. Andriani, G Calvagno, and D Menon. Lossless compression of Bayer mask images using an optimal vector prediction technique. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, Sep.

[9] R. Ansari, N. Memon, and E. Ceran. Near-lossless image compression techniques. *Journal of Electronic Imaging*, 7:486–494, July 1998.

[10] ANSI/I3A. IT10.7466, PhotographyElectronic Still Picture ImagingReference Input Medium Metric RGB Color Encoding (RIMM-RGB). Standard, 2002.

[11] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms-part I: methodology and experiments with synthesized data. *IEEE Trans. Im. Proc.*, 11:972–983, Sep. 2002.

[12] S. Battiato, A. Bruna, A. Buemi, and F. Naccari. Coding techniques for CFA data images. In *Proc. of 12th Int. Conf. on Image Analysis and Processing*, pages 418–423, Sep. 2003.

[13] S. Battiato, A. Buemi, L. Torre, and A. Vitali. A Fast Vector Quantization Engine for CFA Data Compression. In *Proc. of IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, pages 11–20, 2003.

[14] S. Battiato, M. Mancuso, A. Bosco, and M. Guarnera. Psychovisual and statistical optimization of quantization tables for DCT compression engines. In *Proc. of 11th Int. Conf. on Image Analysis and Processing*, pages 602–606, Sep 2001.

[15] I. Bauermann and E. Steinbach. Further lossless compression of JPEG Images. In *Proc.of Picture Coding Symposium*, 2004.

[16] B.E Bayer. Color imaging array. U.S. Patent 3 971 065, 1976.

[17] A. Bosco, M. Mancuso, S. Battiato, and G. Spampinato. Adaptive temporal filtering for CFA video sequences. In *Proc. of ACIVS*, pages 1–6, Singapore, 2002.

[18] A. Bovik. *Handbook on Image and Video Processing.* Academic Press, New York, 2003.

[19] N. Bozinovic and J. Konrad. Scan order and quantization for 3D-DCT coding. In *Proc of SPIE Visual Communications and Image Processing*, volume 5150, pages 1204–1215. SPIE, 2003.

[20] N. Brady. MPEG-4 standardized methods for the compression of arbitrarily shaped video objects. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(8):1170–1189, Dec. 1999.

[21] A. Bruna, F. Vella, A. Buemi, and S. Curti. Predictive differential modulation for CFA compression. In *Proc. of 6th NORSIG*, pages 101–104, June 2004.

[22] Y. Cheng, K. Xie, and G. Zhang. Context-Based Lossless Compression of Mosaic Image with Bayer Pattern. In *Proc. of CISP*, volume 1, pages 481–485, May 2008.

[23] K.S. Choi, E.Y Lam, , and K. K. Wong. Automatic source camera identification using the intrinsic lens radial distortion. *Optics Express*, 14(24):11551–11565, Nov. 2006.

[24] K. Chung and Y Chan. A Lossless Compression Scheme for Bayer Color Filter Array Images. *IEEE Tran. on Im. Proc.*, 17(2):134–144, Feb. 2008.

[25] CIE. Commission internationale de l'Eclairage proceedings. Cambridge University Press, 1931.

[26] R. N. Clark. Digital Camera Sensor Performance Summary. Website. `http://www.clarkvision.com/imagedetail/digital.sensor.performance.summary/`.

[27] D. Coffin. Decoding raw digital photos in Linux (DCRAW). Website. `http://www.cybercom.net/~dcoffin/dcraw/`.

[28] D. R. Cok. Single-chip electronic color camera with color-dependent birefringent optical spatial frequency filter and red and blue signal interpolating circuit. U.S. Patent 4 605 956, 1994.

[29] Dalsa Company. CCD vs. CMOS. Website, 2005. `http://www.dalsa.com/markets/ccd_vs_cmos.asp`.

[30] Foveon Company. Website. `www.foveon.com`.

[31] Kodak Company. Charge-Coupled Device (CCD) Image Senrors. Technical report, Eastman Kodak Company - Image Sensor Solutions, Rochester, New York, 2001.

[32] Kodak Company. Conversion of Light (Photons) to Electronic Charge. Technical report, Eastman Kodak Company - Image Sensor Solutions, Rochester, New York, 2001.

[33] Kodak Company. Kodak Image Sensors - ISO Measurment. Technical report, Eastman Kodak Company - Image Sensor Solutions, Rochester, New York, 2001.

[34] J. Compton and J. Hamilton. Color Filter Array 2.0. Website. `http://johncompton.pluggedin.kodak.com/default.asp?item=624876`.

[35] H.I. Cuce, A.E. Cetin, and M.K. Davey. Compression of Images in CFA Format. In *Proc of ICIP*, pages 1141–1144, Oct. 2006.

[36] D. Dai, L. Liu, and T. Tran. Adaptive Block-Based Image Coding with Pre-/Post-Filtering. In *Proc. of the Data Compression Conference*, pages 73–82, 2005.

[37] W. Dajun and T. Chong. Lossless medical image compression algorithm exploring three dimensional space. In *Proc of 5th ICIP*, volume 2, pages 1062–1064, 2000.

[38] A. Deever and S. Hemami. What's Your Sign?: Efficient Sign Coding for Embedded Wavelet Image Coding. In *Proc. of Data Compression Conference*, pages 273–283, 2000.

[39] C. Doutre and P. Nasiopoulos. An Efficient Compression Scheme for Colour Filter Array Video Sequences. In *Proc. IEEE 8th Workshop on Multimedia Signal Processing*, pages 166–169, Oct. 2006.

[40] C. Doutre and P. Nasiopoulos. An Efficient Compression Scheme for Colour Filter Array Images Using Estimated Colour Differences. In *Proc. Canadian Conf. on Electrical and Comp. Engin.*, pages 24–27, Apr. 2007.

[41] C. Doutre and P. Nasiopoulos. Analysis of the Impact of Demosaicking on JPEG Image Compression. In *Proc. of Int. Workshop on Image Analysis for Multimedia Interactive Services*, pages 71–71, June 2007.

[42] C. Doutre, P. Nasiopoulos, and K. N. Plataniotis. H.264-Based Compression of Bayer Pattern Video Sequences. *IEEE Tran. on Circuits and Systems for Video Technology*, 18(6):725–734, June 2008.

[43] Dpreview. Worldwide digital camera sales. Website, 2004. `http://www.dpreview.com/news/0401/04012601pmaresearch2003sales.asp`.

[44] J. Driesen and P. Scheunders. Wavelet-based color filter array demosaicking. In *Proc. of ICIP*, pages V: 3311–3314, 2004.

[45] E. Dubois. Frequency-Domain Methods for Demosaicking of Bayer-Sampled Color Images. 12(12):847–850, Dec. 2005.

[46] E-Gear. How to buy your next digital camera. Website, 2005. `http://www.e-gear.com/story/story.bsp?sid=29608&var=story`.

[47] Helsingius M. Kuosmanen P. Egiazarian, K. and J. Astola. Removal of blocking and ringing artefacts using transform domain denoising. In *Proc. of ISCAS*, volume 4, pages 139–142, 1999.

[48] K. O. Egiazarian, J. T. Astola, M. Helsingius, and P. Kuosmanen. Adaptive denoising and lossy compression of images in transform domain. *Journal of Electronic Imaging*, 8:233–245, July 1999.

[49] K. O. Egiazarian, V. P. Melnik, V. V. Lukin, and J. T. Astola. Local transform-based denoising for radar image processing. In *Proc. of SPIE Nonlinear Image Processing and Pattern Analysis XII*, volume 4304, Jan 2001.

[50] FillFactory. Technology image sensor: the color filter array. Website. `http://www.fillfactory.com/htm/technology/htm/rgbfaq.htm`.

[51] G. D. Finlayson, S. D. Hordley, and Hubel P. M. Color by correlation: a simple, unifying framework for color constancy. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23:1209–1221, Nov. 2001.

[52] A. Foi. Practical denoising of clipped or overexposed noisy images. In *Proc. 16th European Signal Process. Conf., EUSIPCO*, Lausanne, Switzerland, Aug. 2008.

[53] A. Foi, V. Katkovnik, and K. Egiazarian. Signal-dependent noise removal in Pointwise Shape-Adaptive DCT domain with locally adaptive variance. In *Proc. of 15th EUSIPCO*, Poznan, Poland, Sep. 2007.

[54] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single image raw-data. *to appear in IEEE Trans. on Im. Proc.*

[55] E.R. Fossum. Digital Camera System on a Chip. *IEEE Micro*, 18:8–15, May/June 1998.

[56] D. Furman and M. Porat. Video Compression in the Cosine Time-Spatial Domain using Human Vision Characteristics. In *Proc of IEEE IAPR Conf. on Image and Vision Comp*, pages 333–337, 2001.

[57] F. Gasparini and R. Schettini. Color balancing of digital photos using simple image statistics. *Pattern Recognition*, 37:1201–1217, 2004.

[58] F. Gastaldi, C. C. Koh, M. Carli, A. Neri, and S. K. Mitra. Compression of videos captured via Bayer patterned Color Filter Array. In *Proc. 13th EUSIPCO*, Antalya, Turkey, 2005.

[59] G. Gilder. *The Silicon Eye: How a Silicon Valley Company Aims to Make All Current Computers, Cameras, and Cell Phones Obsolete*. W. W. Norton & Company, 2005.

[60] F. Golchin and K.K. Paliwal. A context-based adaptive predictor for use in lossless image coding. In *Proc. of IEEE Conf. in Speech and Image Technologies for Computing and Telecommunications*, volume 2, pages 711–714, Dec. 1997.

[61] A. Gouze, M. Antonini, M. Barlaud, and B. Macq. Optimized lifting scheme for two-dimensional quincunx sampling images. In *Proc. of ICIP*, volume 2, pages 253–256, Oct 2001.

[62] J. E. Greivenkamp. Color dependent optical prefilter for the suppression of aliasing artifacts. *Applied Optics*, 29:676–684, Feb. 1990.

[63] H.-J. Grosse, M.R. Varley, T.J. Terrell, and Y.K. Chan. Sub-block classification using a neural network for adaptive zigzag reordering in JPEG-like image compression scheme. In *Proc. IEE Colloquium on Neural and Fuzzy Systems*, volume 9, pages 1–4, May 1997.

[64] B.K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Color plane interpolation using alternating projections. *IEEE Trans. Image Process.*, 11:9:9971013, Sep. 2002.

[65] J. Hamilton and J.E. Adams. Adaptive color plane interpolation in single sensor color electronic camera. U.S. Patent 5 629 734, 1997.

[66] C.H. Han, S.G Kwon, S.J. Lee, E.S Kim, and K. Sohng. Color correction method for CMOS camera phone image. In *IEEE Int. Symp. on Consumer Electronics*, pages 138–141, Sep. 2004.

[67] K. Hanma, M. Masuda, H. Nabeyama, and Y. Saito. Novel technologies for automatic focusing and white balancing of solid state color video camera. *IEEE Trans. Consumer Electron.*, 29:376–382, Aug. 1983.

[68] Hasselblad. Hasselblad H3DII-50 datasheet. Website, 2008. `http://www.hasselbladusa.com/products/h-system/h3dii-50.aspx`.

[69] K. Hirakawa and T. Parks. Joint Demosaicing and Denoising. *IEEE Trans. on Im. Proc.*, 15:2146–2157, 2006.

[70] Y. Huh, K. Panusopone, and K.R. Rao. Variable block size coding of images with hybrid quantization. *IEEE Trans. on Circuits and Systems for Video Technology*, 6(6):679–685, Dec. 1996.

[71] IEC. 61966-2-1:1999-10, Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB. Standard, 1999.

[72] K. Iizuka, M. Miyamoto, H. Matsui, and K. Hashiguchi. A 0.2 b/pixel 16 mW real-time analog image encoder in 0.8 m CMOS. In *Proc. of ISSCC*, pages 190–191, Feb. 1997.

[73] Texas Instruments. Digital still camera. Website. `http://focus.ti.com/docs/solution/folders/print/80.html`.

[74] K. Ishikawa. Color reproduction of a single-chip color camera with a frame transfer CCD. *IEEE Journ. Solid-State Circuits*, SC-16:101–103, Apr. 1981.

[75] ISO. 11664: Colorimetry – CIE 1976 L*a*b* Colour space. Standard, 1976.

[76] ISO. 12232: Photography - Electronic still picture cameras - Determination of ISO speed. Standard, 1998.

[77] ISO. 12234-2: Electronic still-picture imaging Removable memory Part 2: TIFF/EP image data format. Standard, 2001.

[78] J. Janesick. Dueling Detectors: CCD or CMOS? *SPIE **Oe**magazine*, pages 30–34, Feb. 2002.

[79] R Jehyuk and J. Youngjoong. A new wide dynamic range fixed point ADC for FPAs. In *Proc. of MWSCAS*, volume 2, pages 243–245, 2002.

[80] O. Kalevo and H. Rantanen. Noise Reduction Techniques for Bayer-Matrix Images. In *Proc. of SPIE Sensors and Camera systems for scientific, industrial, and digital photography applications III*, volume 4669, 2002.

[81] O. Kalevo and H. Rantanen. Sharpening Methods for Images Captured through Bayer Matrix. In *Proc. of SPIE Sensors, Cameras, and Applications for Digital Photography V*, 2003.

[82] W.C. Kao, S.H. Wang, L.Y. Chen, and S.Y. Lin. Design considerations of color image processing pipeline for digital cameras. *IEEE Trans. Consumer Electron.*, 52:1144–1152, Nov. 2006.

[83] W.C. Kao, S.H Wang, C.C Kao, C.V. Huang, and S.Y. Lin. Color reproduction for digital imaging systems. In *Proc. of ISCAS*, pages 4599–4603, May 2006.

[84] L. Karray, P. Duhamel, and O. Rioul. Image coding with an L-infinite norm and confidence interval criteria. *IEEE Trans. on Im. Proc.*, 7(5):621–631, May 1998.

[85] S. Kawahito, Y. Tadokoro, and A. Matsuzawa. CMOS image sensors with video compression. In *Proc. Asia and South Pacific Design Automation Conference, ASP-DAC 1998.*, pages 595–600, Feb 1998.

[86] S. Kawahito, M. Yoshida, M. Sasaki, K. Umehara, D. Miyazaki, Y. Tadokoro, K. Murata, S. Doushou, and A. Matsuzawa. A CMOS image sensor with analog two-dimensional DCT-based compression circuits for one-chip cameras. *IEEE Journal of Solid-State Circuits*, 32(12):2030–2041, Dec. 1997.

[87] Y. Ke, J. Sun, J. Zhang, and J. Li. 3D Volume Data Compression Based on Adaptive Wavelet. In *Proc. of 6th Congress on Intelligent Control and Automation*, volume 2, pages 10440–10444, June 2006.

[88] S.E. Kemeny, H.H. Torbey, H.E. Meadows, R.A. Bredthauer, M.A. La Shell, and E.R. Fossum. CCD focal-plane image reorganization processors for lossless image compression. *IEEE Journal of Solid-State Circuits*, 27(3):398–405, March 1992.

[89] A.I. Khuri. *Advanced Calculus with Applications in Statistics*. John Wiley & Sons; Ed.2, 2003.

[90] R. Kimmel. Demosaicing: image reconstruction from color CCD samples. *IEEE Trans. on Im. Proc.*, 7:1221–1228, 1999.

[91] Kodak. Color Corrections for Image Sensors. Application Note, Revision 2.0, Oct. 2003.

[92] C.C. Koh, J. Mukherjee, and S.K. Mitra. New efficient methods of image compression in digital cameras with color filter array. *IEEE Trans. on Cons. Electr.*, 49(4):1448–1456, Nov. 2003.

[93] H. Kondo and Y. Oishi. Digital image compression using directional sub-block DCT. *Proc. of Int. Conf. on Communication Technology*, 1:985–992, 2000.

[94] E. Y. Lam. Combining gray world and retinex theory for automatic white balance in digital photography. In *Proc. IEEE Int. Symp. Cons. Elect.*, page 134139, June 2005.

[95] G. Langdon and J. Rissanen. A simple general binary source code. *IEEE Trans. on Information Theory*, 28(5):800–803, Sep. 1982.

[96] C. A. Laroche and M.A. Presscott. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. U.S. Patent 5 373 322, 1994.

[97] S.Y Lee and A. Ortega. A novel approach of image compression in digital cameras with a Bayer color filter array. In *Proc. of ICIP*, volume 3, pages 482–485, 2001.

[98] Y. L. Lee, H. C. Kim, and H. W. Park. *IEEE Trans. on Im. Proc.*, 7(2):229–234, 1998.

[99] W.D Leon, S. Balkir, K. Sayood, and M. W. Hoffman. Charge-based prediction circuits for focal plane image compression. In *Proc. of ISCAS*, pages 936–939, 2004.

[100] X. Li. Demosaicing by successive approximation. *IEEE Tran. on Im. Proc.*, 14(3):370–379, March 2005.

[101] X. Li, X. Chen, X Xie, and etc. Pre-Processing and Vector Quantization Based approach for CFA Data Compression in Wileless Endoscopy Capsule. In *Proc. of 4th IEEE Int. Symp. on Biomedical Imaging*, pages 1172–1175, Apr. 2007.

[102] X. Li and B. Furht. An Approach to Image Compression Using Three-Dimensional DCT. In *Proc. of 6th Int. Conf. on Visual Information System*, Sep. 2003.

[103] N.-X. Lian, L. Chang, V. Zagorodnov, and Y.-P. Tan. Reversing Demosaicking and Compression in Color Filter Array Image Processing: Performance Analysis and Modeling. *IEEE Trans. on Im. Proc.*, 15(11):3261–3278, Nov. 2006.

[104] Z. Lin, M. W. Hoffman, W. D. Leon, N. Schemm, and S. Balkir. A CMOS image sensor with focal plane SPIHT image compression. In *Proc. of ISCAS*, pages 2134–2137, May 2008.

[105] L. Lindsay MacDonald and M. R. Luo, editors. *Colour Imaging: Vision and Technology*. Wiley, 1999.

[106] D. Litwiller. CMOS vs. CCD: Maturing Technologies, Maturing Markets. *Laurin publishing: Photonics Spectra*, Aug. 2005.

[107] R. Lukac. New framework for automatic white balancing of digital camera images. *Signal Processing*, 88:1144–1152, March 2008.

[108] R. Lukac, K. Martin, and K.N. Plataniotis. Demosaicked Image Postprocessing Using Local Color Ratios. *IEEE Ttrans on Circ. and Sys. for Video Tech*, 14(6):914–920, June 2004.

[109] R. Lukac and K.N. Martin, K.and Plataniotis. Digital camera zooming based on unified CFA image processing steps. *IEEE Trans. Cons. Electron.*, 50:15–24, 2004.

[110] R. Lukac and K.N. Plataniotis. Color filter arrays: design and performance analysis. *IEEE Trans. Consum. Electron.*, 51:1260–1267, Nov. 2005.

[111] R. Lukac and K.N. Plataniotis. Universal demosaicking for imaging pipelines with an RGB color filter array. *Pattern Recognition*, 38:2208–2212, Nov. 2005.

[112] R. Lukac and K.N. Plataniotis. Color Filter Arrays for Single-Sensor Imaging. In *Proc. 23rd Biennial Symp. on Comm.*, pages 352– 355, June 2006.

[113] R. Lukac and K.N. Plataniotis. Single-sensor camera image compression. *IEEE Tran. on Consumer Electronics*, 52(2):299–307, May 2006.

[114] V.V. Lukin, N.N. Ponomarenko, M. S. Zriakhov, A. A. Zelensky, K. O. Egiazarian, and J. T. Astola. Quasi-optimal compression of noisy optical and radar images. In *Proc. of the SPIE Image and Signal Processing for Remote Sensing XII*, volume 6365, Oct. 2006.

[115] G. Luo. Color filter array with sparse color sampling crosses for mobile phone image sensors. In *Proc. IEEE Int. Image Sensor Workshop*, 2007.

[116] G. Luo. A novel color filter array with 75 percent of transparent elements. In *Proc. of SPIE, V6502, Digital Photography III*, 2007.

[117] Q. Luo and J.G. Harris. A novel integration of on-sensor wavelet compression for a CMOS imager. In *Proc. of ISCAS*, volume 3, pages 325–328, 2002.

[118] Y. Luo and R.K. Ward. Removing the blocking artifacts of block-based DCT compressed images. *IEEE Trans. on Im. Proc.*, 12(7):838–842, July 2003.

[119] P. Magnan. Detection of visible photons in CCD and CMOS: A comparative view. *ElSevier Nuclear Instruments and Methods in Physics Research, section A*, 504:199–212, 2003.

[120] F. Majid. Is the Nikon D70 NEF (RAW) format truly lossless? Website. `http://www.majid.info/mylos/weblog/2004/05/02-1.html`.

[121] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[122] D. Markman and D. Malah. Hyperspectral image coding using 3D transforms. In *Proc of ICIP*, volume 1, pages 114–117, 2001.

[123] D. Marpe, G. Blattermann, J. Ricke, and Maass. A two-layered wavelet-based algorithm for efficient lossless and lossy image compression. *IEEE Tran. on Circuits and Systems for Video Technology*, 10(7):1094–1102, Oct. 2000.

[124] K.L. Mehdi, H.T. Sencar, and N. Memon. Blind source camera identification. In *Proc. of ICIP*, volume 1, pages 709–712, Singapore, 2004.

[125] B. Meyer and P. Tischer. Glicbawls - Grey Level Image Compression by Adaptive Weighted Least Squares. In *Proc. of the Data Compression Conf.*, page 503, 2001.

[126] A. Morimura, K. Uomori, Y. Kitamura, A. Fujioka, J. Harada, S. Iwamura, and M. Hirota. A digital video camera system. *IEEE Trans. Consumer Electron.*, 36:3866–3876, Nov. 1990.

[127] D.D. Muresan and T.W. Parks. Demosaicing using optimal recovery. *IEEE Tran. on Im. Proc.*, 14(2):267–278, Feb. 2005.

[128] H. Nabeyama. All-solid-state color camera with single-chip MOS imager. *IEEE Trans. Consumer Electron.*, CE-27:40–45, Feb. 1981.

[129] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, 2005.

[130] K. Nallaperumal, S. Christopher, S.S. Vinsley, and R.K. Selvakumar. New Efficient Image Compression Method for Single Sensor Digital Camera Images. In *Proc. of Int. Conf. on Computational Intelligence and Multimedia App.*, volume 3, pages 113–117, Dec. 2007.

[131] J.I. Nutt, A.C.G. ; Hirsh. Optical phase noise filter using randomly placed spots. U.S. Patent 6 031 666, 2000.

[132] Y. Okano. Optical Phase-Noise Filter for Color Portrait Photography. In *Proc. of SPIE Int. Commission for Optics 13th Conf. Digest*, volume 29, pages 104–105, 1984.

[133] Technical Standardization Committee on AV & IT Storage. JEITA CP-3451: Exchangeable image file format for digital still cameras: Exif Version 2.2. Standard. `http://www.exif.org/Exif2-2.PDF`.

[134] W. Ouyang, C. Xiao, W. Ju, and W. Song. The dynamic range acquisition of DCT and IDCT algorithms. In *Proc. of 48th Midwest Symposium on Circuits and Systems*, volume 1, pages 429–431, Aug. 2005.

[135] D. Paliy. *Local approximations in demosaicing and debluring if digital sensor data*. PhD thesis, Tampere University of Technology, 2007.

[136] D. Paliy, R. Bilcu, V. Katkovnik, and M. Vehvilinen. Color filter array interpolation based on spatial adaptivity. In *Proc. of SPIE IS&T Electronic Imaging, Image Processing: Algorithms and Systems V.*, pages 12–20, San Jose, California, Jan. 2007.

[137] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian. Spatially adaptive color filter array interpolation for noiseless and noisy data. *Int. Journal of Imaging Systems and Technology*, 17:3:105–122, 2007.

[138] M. Parmar and S.J. Reeves. A perceptually based design methodology for color filter arrays. In *Proc. of ICASSP*, volume 3, pages 473–47, May 2004.

[139] K.A. Parulski. Color filters and processing alternatives for one-chip cameras. *IEEE Trans. Electron Devices*, 32:1381–1389, Aug. 1985.

[140] W.B. Pennebaker and J.L. Mitchel. *JPEG Still Image Data Compression Standard*. Van Nostrand Rainhold, New York, 1993.

[141] N. Ponomarenko, K. Egiazarian, V. Lukin, and J. Astola. Additional lossless compression of JPEG images. In *Proc. of the 4th Int. Symp. on Image and Signal Processing and Analysis*, pages 117–120, Sep. 2005.

[142] N. Ponomarenko, V. Lukin, M. Zriakhov, K. Egiazarian, and J. Astola. Lossy Compression of Images with Additive Noise. In *Proc. of ACIVS*, pages 381–386, 2005.

[143] N. N. Ponomarenko, K. Egiazarian, V. V. Lukin, and J. T. Astola. Compression of image block means for non-equal size partition schemes using delaunay triangulation and prediction. In *Proc. of the Data Compression Conference*, pages 239–243, 2002.

[144] N. N. Ponomarenko, K. Egiazarian, V. V. Lukin, and J. T. Astola. DCT Based High Quality Image Compression. In *Proc. of Scandinavian Conference on Image Analysis*, volume 3540, pages 1177–1185, Geneva, Switzerland, 2005.

[145] N. N. Ponomarenko, K. O. Egiazarian, V. V. Lukin, and J. T. Astola. High-Quality DCT-Based Image Compression Using Partition Schemes. *IEEE Signal Processing Letters*, 14(2):105–108, Feb. 2007.

[146] C. Poynton. *A Technical Introduction to Digital Video*. Wiley, New York, 1996.

[147] R.L. Queiroz and P. Fleckenstein. Very fast jpeg compression using hierarchical vector quantization. *IEEE Signal Processing Letters*, 7(5):97–99, May 2000.

[148] R.J. Rak. A system for transform vector coding of images. In *Proc. of 3rd Int. Conf. on Signal Processing*, volume 2, pages 994–997, Oct. 1996.

[149] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew. Color Image Processing Pipeline in Digital Color Cameras. *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, 22:34–43, Jan. 2005.

[150] K. R. Rao and P. Yip. *The Transform and Data Compression Handbook*. CRC Press, 2006.

[151] S.M. Ross. *Introduction to Probability Models*. Academic Press; Ed.8, 2002.

[152] C. Rushforth. *Image Recovery: Theory and Application Chap. Signal restoration, functional analysis, and Fredholm integral equations of the first kind.* Academic Press, 1987.

[153] S. Saha. Image compression - from DCT to wavelets: A review. *ACM Crossroads Magazine*, 6:3:12–21, Spring 2000.

[154] S.K. Saha, M.K. Baowaly, M.R. Islam, and M.M. Rahaman. Lossless compression of JPEG and GIF files through lexical permutation sorting with Greedy Sequential Grammar Transform based compression. In *Proc. of 10th Int. Conf. Computer and information technology*, pages 1–5, Dec. 2007.

[155] S. Said and W. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Technology*, 6:243–250, 1996.

[156] J. Savard. Website. `http://www.quadibloc.com/other/cfaint.htm`.

[157] G Schaefer and J. Obstoj. *Advances in Visual Computing, Chapter:Lossless Compression of CCD Sensor Data.* Springer House, 2005.

[158] P. Schelkens, A. Munteanu, A. Tzannes, and C. Brislawn. JPEG2000. Part 10. Volumetric data encoding. In *Proc of IEEE Int. Symp. on Circuits and Systems ISCAS*, pages 3874–3877, May 2006.

[159] Marc Servais and Gerhard De Jager. Video compression using the three dimensional discrete cosine transform (3D-DCT). In *Proc. of COMSIG 97, South African*, pages 27–32, 1997.

[160] R. A. Serway. *Physics for scientists and engineers.* Saunders College Publishing, 1990.

[161] N.P. Sgouros, S.S. Athineos, P.E. Mardaki, A.P. Sarantidou, M.S. Sangriotis, P.G. Papageorgas, and N.G. Theofanous. Use of an adaptive 3D-DCT scheme for coding multiview stereo images. In *Proc. of IEEE Int. Symp. on Signal Processing and Information Technology*, pages 180–185, Dec. 2005.

[162] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Tran, on Signal Processing*, 41(12):3445–3462, Dec. 1993.

[163] G Sharma, editor. *Digital Color Imaging Handbook.* CRC Press, 2003.

[164] G. Shen, B. Zeng, and Ming Lei Liou. Arbitrarily shaped transform coding based on a new padding technique. *IEEE Trans. on Circuits and Systems for Video Technology*, 11(1):67–79, Jan. 2001.

[165] C. Shoushu, A. Bermak, W. Yan, and D Martinez. Adaptive-Quantization Digital Image Sensor for Low-Power Image Compression. *IEEE Tran. on Circuits and Systems*, 54:1:13–25, Jan. 2007.

[166] H. Siddiqui and C.A. Bouman. Training-based color correction for camera phone images. In *Proc. of ICASSP*, volume 1, pages 733–736, Apr. 2007.

[167] T. Sikora. Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments. *Signal Processing: Image communication*, 7(4-6):381–395, Nov. 1995.

[168] Hakon J. Skretting, K. and S.O. Aase. Improved Huffman coding using recursive splitting. In *In Proc. of NORSIG*, Sep. 1999.

[169] T. Smith and J. Guild. The C.I.E. colorimetric standards and their use. *Trans. of the Optical Society*, 33:73134, 1932.

[170] M. Stirner and G. Seelmann. Improved redundancy reduction for jpeg files. In *Proc. of PCS*, 2007.

[171] T. Tai, Y. Wu, and C. Lin. An adaptive 3-D discrete cosine transform coder for medical image compression. *IEEE Trans. on Information Technology in Biomedicine*, 4(3):259–263, Sep. 2000.

[172] E.J. Tan, Z. Ignjatovic, and M.F. Bocko. A CMOS Image Sensor with Focal Plane Discrete Cosine Transform Computation. In *Proc. of ISCAS*, pages 2395–2398, May 2007.

[173] T. Tang and K.F. Lee. An efficient color image acquisition system for wireless handheld devices. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP 2004)*, volume 3, pages 105–108, May 2004.

[174] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Tran. on Im. Proc.*, 9(7):1158–1170, July 2000.

[175] D. Taubman and M. Marcellin. *JPEG2000: Image compression Fundamentals, Standards and Practice*. Kluwer, Boston, 2002.

[176] A.J Theuwissen. Image processing chain in digital still cameras. In *Proc. of Symp. on VLSI Circuits*, pages 2–5, June 2004.

[177] A. Thuewissen. Image Sensor Architectures for Digital Cinematography. Technical report, DALSA Digital Cinema, 2005.

[178] T. Toi and M. Ohita. A subband coding technique for image compression in single CCD cameras with Bayer color filter arrays. *IEEE Trans. on Consumer Electronics*, 45(1):176–180, Feb. 1999.

[179] T. Tran. The BinDCT: Fast multiplierless approximation of the DCT. *IEEE Signal Processing Letters*, 7:141–144, 2000.

[180] D. Paliy M. Vehvilainen M. Trimeche, M. and V. Katkovnik. Multi-Channel Image Deblurring of Raw Color Componentsr. In *Proc. of SPIE Computational Imaging III*, volume 5674, pages 169–178, 2005.

[181] Y.T. Tsai. Color image compression for single-chip cameras. *IEEE Tran. on Electron Devices,*, 38(5):1226–1232, May 1991.

[182] P. Vora and C. Herley. Trade-offs between color saturation and noise sensitivity in image sensors. In *Proc. of ICIP*, volume 1, pages 196–200, Oct. 1998.

[183] B. A. Wandell. *Foundations of Vision*. Sunderland Mass: Sinauer Press, 1995.

[184] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Im. Proc.*, 13(4):600–612, Apr. 2004.

[185] T. Watanabe. A CCD Color Separation IC for Single - Chip Imagers. In *Proc. IEEE Custom Integrated Circuits*, pages 446–445, 1983.

[186] C. Weerasinghe, W. Li, I. Kharitonenko, M. Nilsson, and S. Twelves. Novel color processing architecture for digital cameras with CMOS image sensors. *IEEE Trans. Consumer Electron.*, 51:1092– 1098, Nov. 2005.

[187] M.J. Weinberger, G. Seroussi, and G. Sapiro. From LOGO-I to the JPEG-LS standard. In *Proc. of ICIP*, volume 4, pages 68–72, 1999.

[188] C. C. Weng, H. Chen, and C. F. Fuh. C novel automatic white balance method for digital still cameras. In *Proc. IEEE Int. Symp. Circuits and Systems*, pages 3801–3804, May 2005.

[189] M. Wien. Variable block-size transforms for H.264/AVC. *IEEE Trans. on Circuits and Systems for Video Technology,*, 13(7):604–613, July 2003.

[190] X. Wu and N. Memon. CALIC - A context based adaptive lossless image codec. In *Proc. of ICASSP*, pages 1890–1893, 1996.

[191] X. Xiang, G. Li, and Z Wang. Low-complexity and high-efficiency image compression algorithm for wireless endoscopy system. *Journal of Electronic Imaging*, 15(2):22–29, 2006.

[192] X. Xie, G. Li, X. Li, and etc. A new approach for near-lossless and lossless image compression with bayer color filter arrays. In *Proc. 3rd Int. Conf. Image and Graphics*, pages 357–360, Dec. 2004.

[193] X. Xie, G. Li, and Z. Wang. A near-lossless image compression algorithm suitable for hardware design in wireless endoscopy system. *EURASIP J. Appl. Signal Process.*, 2007(1):48–48, 2007.

[194] X. Xie, G. Li, Z. Wang, and etc. A novel method of lossy image compression for digital image sensors with Bayer color filter arrays. In *Proc. of ISCAS*, volume 5, May 2005.

[195] S. Xiong, R. Ramchandran, M.T. Orchard, and Y.Q. Zhang. A Comparative Study of DCT- and Wavelet-Based Image Coding. *IEEE Tran. on Circuits and Systems for Video Technology*, 9:5:692–695, Aug. 1999.

[196] X. Xiong, O. Guleryuz, M. Orchard, and Z. Xiong. A DCT-based embedded image coder. *IEEE Signal Processing Letters*, 3:289–290, 1996.

[197] S. Yea and W.A. Pearlman. A Wavelet-Based Two-Stage Near-Lossless Coder. *IEEE Tran. on Im. Proc.*, 15(11):3488–3500, Nov. 2006.

[198] B. Yeo and E. Liu. Volume Rendering of DCT-Based Compressed 3D Scalar Data. *IEEE Trans. on Visualization and Computer Graphics*, 1(1):29–43, 1995.

[199] L. Zhang and X.L. Wu. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Tran. on Im. Proc.*, 14(12):2167–2178, Dec. 2005.

[200] M. Zhang and A. Bermak. Architecture of a Low Storage Digital Pixel Sensor Array with an On-Line Block-Based Compression. In *Proc. Proc. 4th IEEE Int. Symp. on Electronic Design, Test and Applications, DELTA 2008.*, pages 167–170, Jan. 2008.

[201] N. Zhang and X. Wu. Lossless compression of color mosaic images. *IEEE Trans. on Im. Proc.*, 15(6):1379–1388, June 2006.

[202] D. Zhao, W. Gao, and Y.K. Chan. Morphological representation of DCT coefficients for image compression. *IEEE Tran. on Circuits and Systems for Video Technology,*, 12(9):819–823, Sep. 2002.

[203] J. Zhou. Getting the most out of your image-processing pipeline. White Paper, Texas Instruments, Oct. 2007.

Publications