Silesian University of Technology

FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND COMPUTER SCIENCE

PROGRAMME: CONTROL, ELECTRONIC AND INFORMATION ENGINEERING

Master Thesis

Improving the efficiency of lossless image compression using extensions of Part 2 of the JPEG 2000 standard

Author: Szymon Zosgórnik, BEng

Supervisor: Roman Starosolski, DSc PhD

Gliwice, September 2021

Załącznik Nr 2 do Zarz. Nr 97/08/09

O Ś W I A D C Z E N I E

Wyrażam zgodę/nie wyrażam* zgody na udostępnienie mojej pracy dyplomowej/rozprawy doktorskiej*

………………………………., dnia …………………………………….

…………………………………………………………………………
(podpis)

…………………………………………………………………………
(poświadczenie wiarygodności podpisu przez Dziekanat)

** właściwe podkreślić*

Abstract

Lorem ipsum. . .

Keywords: lossless image compression, image processing, JPEG 2000, discrete wavelet transform, entropy estimation, multithreading, modern C++

Contents

Chapter 1. Introduction

1.1 Preface

The usage of digital images is constantly growing across whole world. There are multiple types of applications where memory usage matters to the users. Image compression is a possible solution to this problem in some of these fields. For example it is mission critical component in medical and picture archiving and communication systems (PACSs) [18]. There are two major types of such compression. The one is lossy variant and the other one is lossless. Applying lossy methods to the image can result in the occurrence of compression artifacts. However, there are applications where such disadvantage is negligible, e.g. natural images and photographs processing in Internet day-to-day usage [4]. On the other hand lossless image compression does not produce such artefacts, sacrificing some performance and bitrates optimizations. It is employed in mentioned before medical systems. Images used for the sake of diagnostics can be taken as an example. In some countries there are regulations that forbid applying lossy compression to such images [18]. Moreover, the usage of lossless variant is more desired when there exists some uncertainty whether information contained in the image can be discarded. In these scenarios not using any variant of compression can be the only substitute of lossless one [18].

Taking into account mentioned before reasons, some compression algorithms have been introduced as ISO standards [18]. Some notable examples of such algorithms are PNG, JPEG and JPEG 2000 (often written as JP2). The latter was originally developed from 1997 to 2000 with the desire of expanding JPEG capabilities. The main feature of this standard is usage of discrete wavelet transform (DWT) instead of discrete cosine transform (DCT) which was introduced in the predecessor [5]. The other feature of JPEG 2000 is support for lossy and lossless compression. As described before, such compression is needed to be performed in mission critical systems such as medicine. Therefore, the JPEG 2000 standard is utilized in PACSs and Digital Imaging and Communications in Medicine DICOM standard [18]. This standard consist of 16 ISO parts which contain wide set of features. Some notable ones are core system coding and its extensions, motion images, testing and reference software [5].

The successor of JPEG standard improved several aspects over its predecessor. With the usage of its algorithms, e.g. DWT, it was possible to improve compression performance over JPEG. Moreover, there are other improved areas with even greater importance. The few examples of such features are scalability and editability [5]. The JPEG 2000 standard supports both very low and very high rates of the compression. It comes crucial in applications that require such flexibility. Another main advantage of this standard is the ability of effective handling large range of bit rates. It allows to reduce number of steps taken in processing certain images in comparison to JPEG. As an example, reducing the number of bits in some image below certain amount using JPEG standard compliant solution requires reducing the resolution of the input at first. Only after this procedure encoding of the image can be applied. The JPEG 2000 standard supplies adequate feature named multiresolution decomposition structure which makes such transformation transparent and one step only [5].

1.2 Objective of the project

The standard way of performing discrete wavelet transform (DWT) in the JPEG 2000 compliant with Part 1 is to decompose the image into sub-bands using a pair of low- and high-pass filters. This decomposition is applied multiple times using higher DWT orders. The standard order which is used across whole industry is five [13] [8]. The Part 2 of the standard contains several types of extensions which can be applied to modify the encoding algorithm. For instance DWT can be modified in a way that makes decomposition of the image into sub-bands of different shapes possible. Moreover, the strict selection of the pair of filters imposed by Part 1 of the standard can be broken. However, the same pair has to be used for all sub-bands of the image [13]. The other type of applicable modification is skipping some steps of discrete wavelet transform (SS-DWT). It is usually beneficial for processing non-photographic and screen content images. Another way of achieving improvement in terms of compression ratio is applying the reversible histogram packing. This type of extension significantly improves the ratio of compression when the histogram of the image is sparse. It means that unused levels appear between frequently used brightness levels. With the help of described Part 2 compliant extensions to the JPEG 2000 standard it is possible to adaptively adjust the transform for a specified image to improve the compression ratio. The result of this operation can still be correctly decoded by every decompressor which is compatible with the Part 2 of the JPEG 2000 standard.

The objective of the thesis is to develop, implement and test several forms of heuristics which can determine the optimal transform in terms of compression ratio of the given image. Transform shall be compliant with the Part 2 of JPEG 2000. The heuristics shall be rather fast and use entropy as an estimation of the JPEG 2000 encoding. Moreover, they can be greedy and use trial and error approach to some extent. The implementation of the program shall be done in modern C++ to utilize such language capabilities as cross-platform threads. The main target of the application are multi-core CPU architectures. The result of the project work is a tool that quickly determines the transform for the specified image and invokes the JPEG 2000 encoder with selected transform. However, it is acceptable to achieve small time overhead in terms of the entire compression process. The resulting image shall come with the improvement of the lossless JPEG 2000 compression ratio.

1.3 Thesis outline

At the beginning of this paper there is introduction to the domain problem of image processing and compression. Some methods of applying this kind of compression are described in Introduction. Moreover, objective and scope of the thesis are described there.

The last chapter is Summary which wraps up all results and makes some valuable conclusions. At the end there are appendices available such as technical documentation and list of used tables, listings, etc.

Chapter 2. Problem analysis

2.1 Discrete Wavelet Transform

2.1.1 One dimensional DWT

The linear convolution (filtering) of sequences $x(n)$ and $h(n)$ is defined as in equation 2.1:

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \tag{2.1}$$

The one dimensional discrete wavelet transform can be depicted as successive applications (convolutions) of one selected pair of high and low-pass filters. The output of such application is then followed by downsampling by the factor of two. For example, it can be achieved by discarding samples with odd indices after each of filtering operation. It is better visualized in the Figure 2.1 [13]. The pair of low and high-pass filters is known as analysis filter bank in the encoding process. In the signal decoding process it is featured as a synthesis filter bank. The decoding step requires using the inverse of discrete wavelet transform.

Take into consideration a one dimensional signal $x(n) = \{55, 234, 70, 21, 88, 37\}$. It can be better understood as values of pixels in a part of the grayscale image row. It is followed with a pair of low and high-pass filters designated by $h_0(n)$ and $h_1(n)$ respectively. An example of such pair is a lowpass filter $h_0(n) = \{-1, 2, 6, 2, -1\}/8$ and a high-pass filter $h_1(n) = \{-1, 2, -1\}/2$. They are both symmetric and consist of only

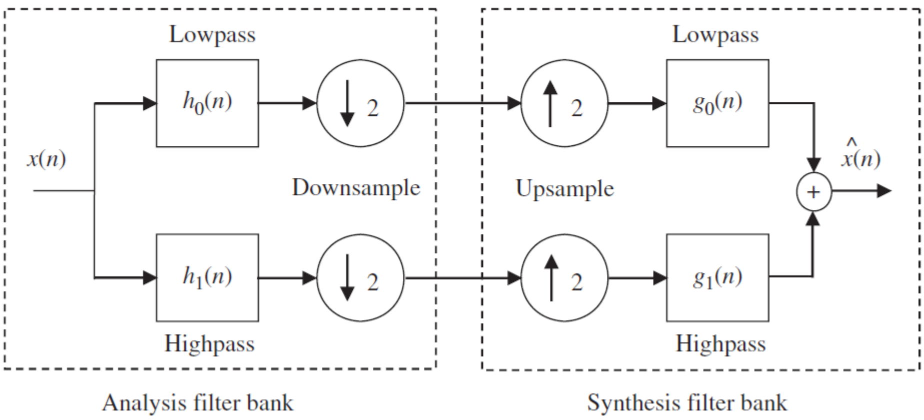

Figure 2.1: 1-D DWT, two-band wavelet analysis and synthesis filter banks [13]

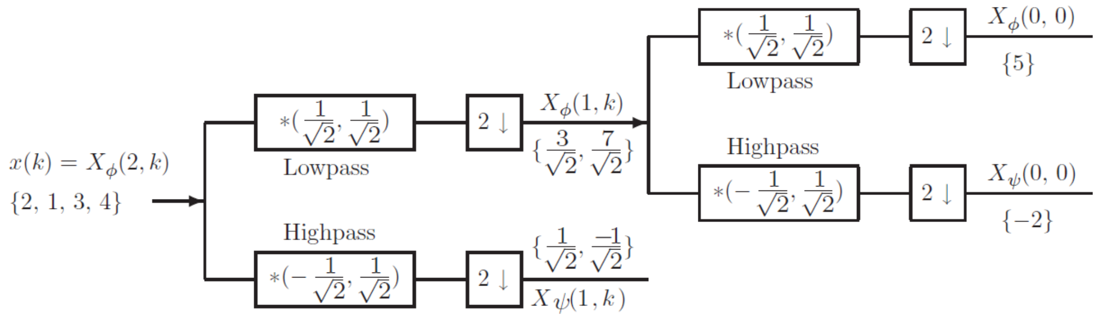

Figure 2.2: Computation of a 2-level 4-point DWT using a two-stage two-channel Haar analysis filter bank [21]

integer operations. Such pair can be presented in the notation of (5, 3) filter bank. This convention indicates that the length of lowpass filter is five and the length of high-pass filter is three. In fact the analysis filter bank presented here was firstly proposed by LeGall and Tabatabai in 1988 and is used in the JPEG 2000 standard for lossless compression of images. The filtering operation has to be defined at the signal boundaries. Therefore, the one dimensional signal is extended in both directions. The Part 1 of the JPEG 2000 standard requires symmetrical extension to be performed in such case [13]. After applying the required symmetrical padding the signal is extended to $x(n) = \{21, 70, 234, 55, 55, 234, 70, 21, 88, 37, 37, 88, 21, 70\}$. Then, the low-pass filter is applied resulting in $x'_0(n) = \{197.25, 75.5, 98.375, 67.125, 45.375\}$ and the high-pass one which results in $x'_1(n) = \{44.75, -85.75, 29, 12.75, -29.5\}$.

The next example shows how to compute the two levels of discrete wavelet transform. To speed up the process no padding option is chosen this time which makes it non-compliant with the JPEG 2000 standard. The filter used here is the most basic one, i.e. Haar analysis filter bank. It is the first wavelet from the Daubechies wavelet family. The calculation process is visualized in the Figure 2.2 [21].

The input is chosen as 4-point signal $X_\phi(2,k) = \{2, 1, 3, 4\}$. This notation emphasizes the fact that it is approximation of the input at scale 2. The so called scaling coefficients (or in other term approximation at scale 1) $X_\phi(1,k)$ are computed by convolving the input $x(k)$ with the low-pass Haar filter impulse response $l(k) = \{1/\sqrt{2}, 1/\sqrt{2}\}$. In the next step there is downsampling by a factor of 2 applied. The output of convolution has five values. The middle three from these fives correspond to cases where both the given input values overlap with the impulse response. As it was described earlier, the odd values are preserved in the downsampling process. In a result first and third value of these three middle ones are the approximation output $X_\phi(1,k)$. In the similar way, the detail coefficients at scale 1 $X_\psi(1,k)$ are computed. The input $x(k)$ is convolved with the high-pass filter impulse response $h(k) = \{-1/\sqrt{2}, 1/\sqrt{2}\}$. Then the downsampling by factor of 2 is

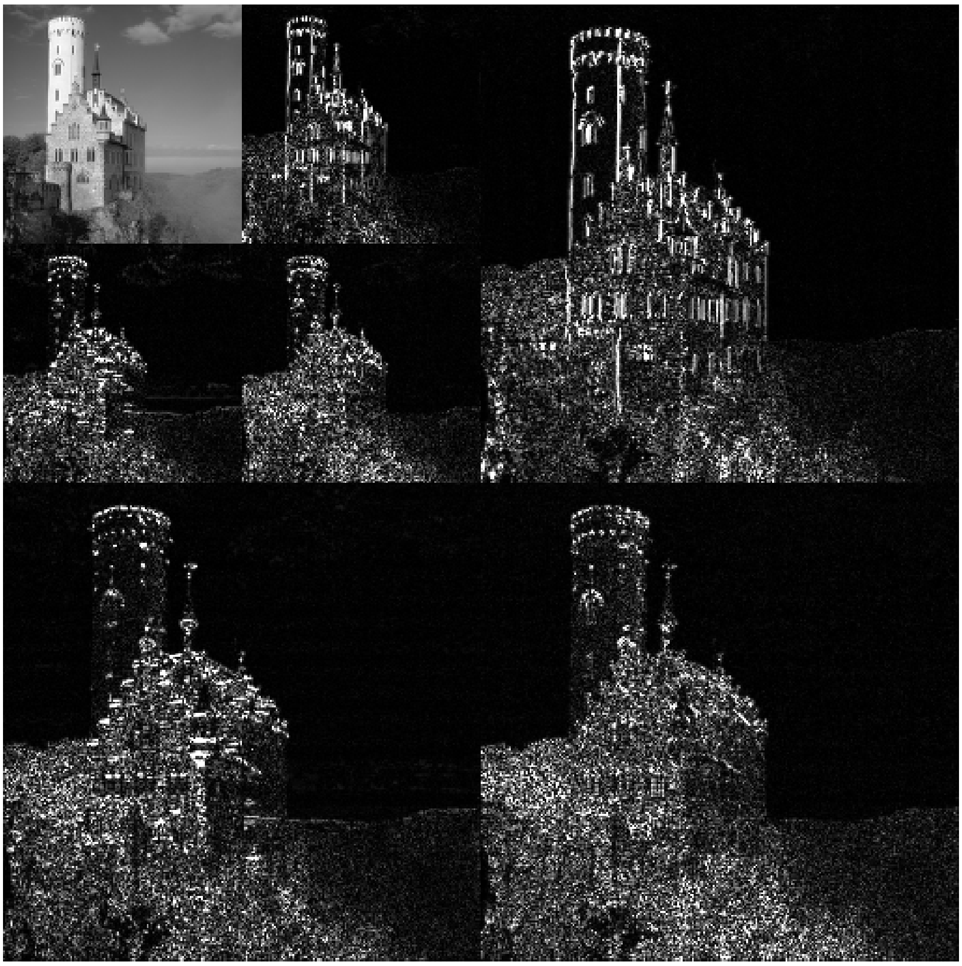

Figure 2.3: 2D DWT applied 2 times to an exemplary image [1]

performed. Note that only only approximation output $X_\phi(1,k)$ of the first stage goes to the second one. The $X_\phi(0,0)$ and $X_\psi(0,0)$ are calculated accordingly at the end of the second stage [21].

2.1.2 Two dimensional DWT

The idea of using lowpass filter is the preservation of low frequencies of a signal while trying to eliminate or at least attenuate the high frequencies. In a result the output signal is the blurred version of the original one. Therefore, the operating principle of the high-pass filter is completely opposite. As a result of applying such filter, the high frequencies of the signal are preserved and the low ones are discarded or at least diminished. The output is a signal consisting of edges, textures and other details [13].

There is presented an example of the effects of the two dimensional discrete wavelet transform on the Figure 2.3. The DWT used here is compliant with the Part 1 of the JPEG2000 standard. The number of DWT stages presented in this example

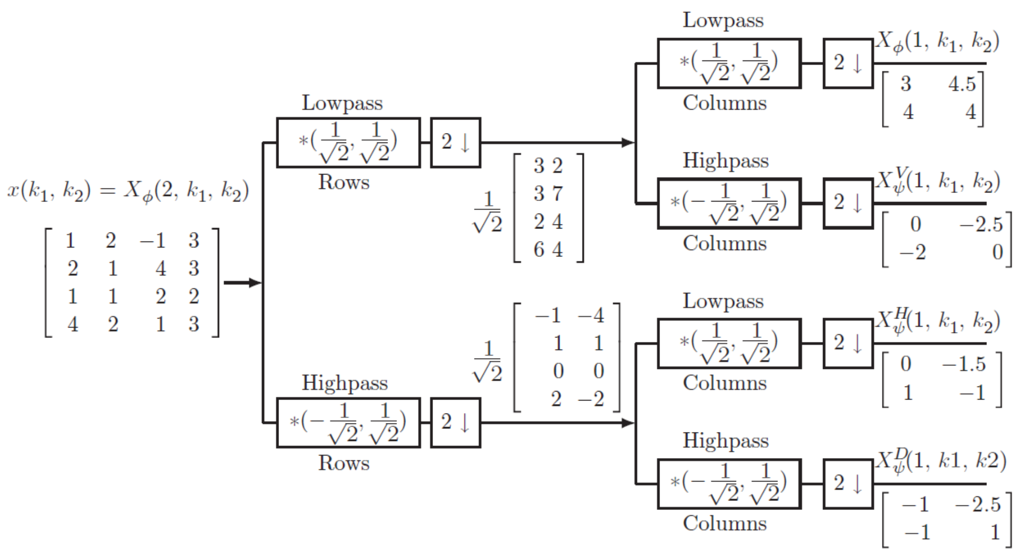

Figure 2.4: Computation of a 1-level 4 $\times$ 4 2-D Haar DWT using a two-stage filter bank [21]

is equal to two. Two dimensional discrete wavelet transform applied first time to the original image yields four same sized sub-images. The LL layer (upper left sub-image) is an approximation of the image and contains the low frequencies. This layer is once more transformed in the next stage. The LH layer (upper right sub-image) preserves high frequencies from the rows of the image. As a result vertical lines and details (brightness) can be seen in the produced sub-image. On the other hand, the HL layer (bottom left) contains high frequencies from the columns of the image. The horizontal details and lines can be noticed there. Lastly, the HH layer (bottom right) preserves the diagonal lines [1].

The process of computing a 1 level two dimensional discrete wavelet transform with usage of two-stage analysis Haar filter bank is shown in Figure 2.4. Coefficients $\mathbf{X}_\phi$ are calculated as a result of lowpass filtering and downsampling to each row of the two dimensional data. Next, similar process process, i.e. lowpass convolution and downsampling is applied to each column of resulting data. The rest of coefficients is obtained in very similar fashion to the previous ones. Coefficients $\mathbf{X}_\psi^H$ are calculated by applying high-pass filtering and downsampling to each row of the 2-D data $\mathbf{x}$ and then followed by applying sequence of low-pass filtering and downsampling to each column of the resulting data. Coefficients $\mathbf{X}_\psi^V$ are obtained by applying low-pass filtering and then downsampling to each column of the resulting data. Lastly, coefficients $\mathbf{X}_\psi^D$ are obtained by applying high-pass filtering and downsampling to each row of the 2-D data $\mathbf{x}$ followed by applying high-pass filtering and downsampling to each column of the resulting data. In the next stage of more complex dwt calculating process only the coefficients $\mathbf{X}_\phi$ are taken into consideration [21].

2.1.3 DWT features summary

- In a nutshell, the discrete wavelet transform is a set of bandpass filters. Usually it is implemented with the usage of low and high-pass filters recursively.
- The computational complexity of computing the DWT in the best case is linear, i.e. $O(N)$.
- The first approach to implement the DWT efficiently is evaluation of the required convolutions with the usage of the polyphase filter structure.
- The second approach is factorization of the polyphase matrix into a product of a set of sparse matrices.
- The two dimensional discrete wavelet transform (with separable filters) is usually computed by the row-column method. One dimensional DWT of all the columns is computed at first. Then the 1-D DWT of all the resulting rows is calculated. The order of the computation in the row-column method can be swapped. The result remains the same.
- Additional memory of approximate half the size of the given data is required in the some implementations of the DWT. However, there also exist methods of computing DWT in-place which do not require additional memory.
- Data reordering is required for an in-place computation of the DWT.
- Data expansion problem can occur due to the finite length of the data in the implementation of the asymmetric filters.
- Symmetric filters provide linear phase response and an effective solution to the border problem [21].

2.2 Part 2 of the JPEG 2000

2.2.1 Introduction

Many ideas have been emerging as the JPEG 2000 was developed. These concept were full of value-added capabilities. However, they were not that important to be gone through the time-consuming ISO standardization process. The Part 1

(ISO/IEC, 2004a) of the standard, i.e. Core coding system, was originally published in 2000. There was a need to create additional parts to include missing features. The Part 2 of the standard, published as ISO/IEC 15444-2 or ITU Recommendation T.801 (ISO/IEC, 2004b), contains multiple such extensions. There is present group of rather small additions that could not merit entire documents of their own. In the Part 1 Core of JPEG 2000 standard decoders are supposed to handle all of the code-stream functionality. The Part 2 is different from first one in this aspect. It is a collection of options that can be implemented on demand to meet very specific requirements of the given market. Moreover, sections within an extension annex can be implemented separately. For example, subsets of extended file format JPX can be used on their own. Therefore, some features of the Part 2 may be present in the wide spectrum of JPEG 2000 applications while the other ones can be less common in the decoders [13].

As it was shown in the previous paragraph, the extensions present in the Part 2 consist of very different set of topics that can modify or add some features to the Part 1 JPEG 2000 compliant processing chain. Some tools can result in the compression efficiency improvement. Others can ameliorate the visual appearance of compressed images. Another group of extensions can modify or extend some functionalities in the other ways. The list of the major topics is presented below [13].

Compression efficiency:

- Variable DC offset (VDCO) - Annex B
- Variable scalar quantization (VSQ) - Annex C
- Trellis coded quantization (TCQ) - Annex D
- Extended visual masking - Annex E
- Arbitrary wavelet decomposition - Annex F
- Arbitrary wavelet transform kernel - Annexes G and H
- Multiple component transform - Annex J
- Nonlinear point transform - Annex K [13]

Functionalities:

- Geometric manipulation - Annex I

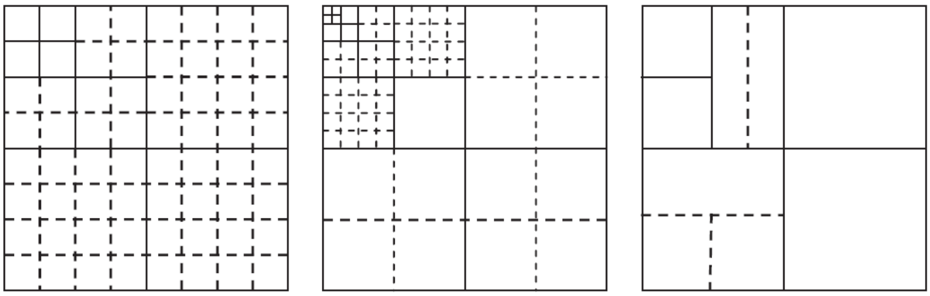

Figure 2.5: Some examples of decomposition compliant with the Part 2 [13]

- Single-sample overlap (SSO/TSSO) - Annex I
- Precinct-dependent quantization - Amendment 1
- Extended region of interest - Annex L
- Extended file format/metadata (JPX) - Annexes M and N
- Extended capabilities signaling - Amendment 2 [13]

2.2.2 Arbitrary Decomposition

In the Part 1 of the JPEG 2000 standard there is only one wavelet decomposition structure allowed. This wavelet is called Mallat dyadic decomposition. Such decomposition is a good first choice to be applied across a wide spectrum of images. However, other ones can improve the quality of the image over specialized classes of the applications. The other effect of applying such decompositions are unequal reductions in the horizontal and vertical dimensional of reduced resolution extracts [13].

Other decomposition styles can be found in the wavelet literature. They include the full packet tree processing and some of its derivatives. The applied packet decomposition derivatives can outperform the solution from Part 1 of the JPEG 2000 standard in some applications. For instance, they come crucial at maintaining regular fine-grain texture. Moreover, the applications that require processing synthetic aperture radar images can benefit from using this extension. The US Federal Bureau of Investigation actively uses a 500 ppi fingerprint compression standard, i.e. WSQ (CJIS, 1997). The decomposition is specialized for the characteristics of fingerprint imagery at 500 dpi [13].

Some of these decomposition can be seen of the Figure 2.5. Resolution decomposition is depicted as solid lines. Dashed lines represent extra sublevel decomposition. On the first example, i.e. image (a), there is available full packet

Table 2.1: Analysis and synthesis filter taps for the floating-point Daubechies (9, 7) filter bank

n	Low-pass, $h_0(n)$	Low-pass, $g_0(n)$
0	+0.602949018236360	+1.115087052457000
± 1	+0.266864118442875	+0.591271763114250
± 2	-0.078223266528990	-0.057543526228500
± 3	-0.016864118442875	-0.091271763114250
± 4	+0.026748757410810	

n	High-pass, $h_1(n)$
-1	+1.115087052457000
$-2, 0$	-0.591271763114250
$-3, 1$	-0.057543526228500
$-4, 2$	+0.091271763114250

n	High-pass, $g_1(n)$
1	+0.602949018236360
$0, 2$	-0.266864118442875
$-1, 3$	-0.078223266528990
$-2, 4$	+0.016864118442875
$-3, 5$	+0.026748757410810

decomposition with such parameters: NL = 3: Ddfs = 111, Doads = 321, Dsads = all 1s. The next picture illustrates FBI decomposition wit specified parameters: NL = 5: Ddfs = 11111, Doads = 2321, Dsads = 1110111111111111. The last image is juts an arbitrary example [13].

The prespecified decomposition structures are not the only feature of this extension. Wavelet packet analysis can be also used to design custom decompositions for specific images or some types of images. It was implemented in these papers [14], [11] and [9]. Such applications often start with a large decomposition tree. Then, they tend to locate a good decomposition based upon specified optimization metric [13].

2.2.3 Arbitrary Wavelet Transforms

The Part 1 of the JPEG 2000 standard specifies only two possible wavelet transforms. The reversible one (5-3R, Table 2.2) and the irreversible one (9-7I, Table 2.1). As it was stated before, both are required to perform periodic symmetric signal extension at the boundaries. It is similar case to the Mallat dyadic decomposition in terms of generic implementation. These filters can compress quite well a wide set of image types. However, certain image classes can be compressed more efficiently with other types of wavelets. Such a flexibility is allowed in the Part 2 compliant applications. The range of wavelet transforms is broadened to include not only the wider range of whole-sample symmetric ones but also half-sample and generic non-symmetric ones. Such ability to handle generic filters makes JPEG 2000 standard

Table 2.2: Analysis and synthesis filter taps for the integer (5, 3) filter bank

n	Low-pass, $h_0(n)$	Low-pass, $g_0(n)$
0	+0.75	+1
±1	+0.25	+0.5
±2	-0.125	

n	High-pass, $h_1(n)$
-1	+1
$-2, 0$	-0.5

n	High-pass, $g_1(n)$
1	+0.75
$0, 2$	-0.25
$-1, 3$	-0.125

a powerful research tool, together with supporting more than niche compression applications [13].

2.3 Computer architecture

2.4 Known solutions

2.4.1 Part 1 compliant applications

There are multiple solutions that implement features from the Part 1 of the JPEG 2000 standard. The OpenJPEG library is an example of such application. It is open-source solution developed to promote wider usage of this standard. The main part of this project is the codec compliant with the Part 1 of the JPEG 2000 standard. Moreover, the OpenJPEG library integrates other parts of the standard [13]. Few of them can be seen on the list below:

- from the Part 2: handling the JP2 boxes and extended multiple component transforms for multi and hyperspectral imagery.
- from the Part 3: MJ2 (Motion JPEG 2000).
- from the Part 11: JPWL (JPEG 2000 Wireless).
- OPJViewer, a GUI based tool for visualization of the J2K, JP2, JPWL and MJ2 files.
- OPJIndexer, a code-stream indexer to view information about the headers and packets localization. This tool is also capable of reading the rate-distortion contribution of each packet to the image [13].

The OpenJPEG library is written in the C programming language and released under the BSD license. The main targets of this software are desktop platforms, i.e. Win32, Unix and Mac OS platforms. The Communications and Remote Sensing Lab (TELE) of the Catholic University of Louvain (UCL) is the main developer group of this library. CS company and CNES is the supporting side. Some of the modules are maintained by the Digital Signal Processing Lab (DSPLab) of the University of Perugia, mainly the JPWL and OPJViewer. [13].

Another example of Part 1 compliant application is the JasPer. This computer software project is aimed to create reference implementation of the standard codec. The project was started by the Image Power Inc. and the University of British Columbia back in the 1997. As OpenJPEG JasPer is written in the C programming language. There are some sample applications available in the codebase. They can come in handy while testing the codec. JasPer is currently released publicly under the MIT license. The library is a component of several notable software projects. It includes but is not limited to netpbm, ImageMagick and KDE. In series of the JPEG 2000 compression tests conducted in 2004 JasPer turned out to be the top performing solution, closely followed by IrfanView and Kakadu. The disadvantage of this implementation is its time performance. In the same tests JasPer was the slowest software. However, it is a feature as this codec was designed to be used as reference in non performance-critical systems.

There is one more notable implementation of the Part 1, i.e. Grok. It is open-sourced software released under the GNU Affero General Public License (AGPL) version 3 license. Its design aims to provide stable, high performant and low memory using solution. The function responsible for decoding process (grk_decompress) is currently over 0.5 the speed of the Kakadu software. Moreover, Grok supports fast sub-tile decoding to standard output for png, jpeg, bmp, pnm and raw formats. This library supports both TLM and PLT code stream markers for fast single-tile and sub-tile decoding of large tiled images. The support of meta-data formats such as XML, IPTC, XMP and ICC profiles is also included. There is also initial version of Part 15 implementation (High Throughput JPEG 2000) available. The final version of this solution should be ten times faster over the original Part 1 of the JPEG 2000 standard.

2.4.2 Kakadu

Kakadu is not only a complete implementation of the JPEG 2000 standard Part 1 but also an application that supports Part and Part 3 in a significant amount of features. The software was originally developed by David Taubman of the University of New South Wales (UNSW) in Australia. The author is also noticeably known for being the designer of EBCOT, i.e. the core coding component of JPEG 2000. The name of this library comes from “Kakadu National Park” which is located in the Northern Territory of Australia [13]. Licensing is more advanced in comparison to solutions mentioned in Part 1 compliant applications. There are separate licensing schemes for research, commercial and demonstration-only applications.

The Kakadu software framework is widely adopted in the substantial range of JPEG 2000 products. The few examples are Apple’s Quicktime v6 for MAC, Yahoo’s Messenger, Google Earth and Internet Archive [13]. Features that can be realized thanks to Kakadu’s implementation by the named beforehand solutions include live video and products for geospatial imagery such as MicroImages TNT. Moreover, this framework is used in medical imaging applications, interactive image rendering applications, remote browsing of large collection and images, digital cinema applications and the other fields that require compression and decompression of JPEG 2000 images and videos [13].

Kakadu is considered as a comprehensive, heavily optimized and fully compliant software toolkit for JPEG 2000 developers. There are multiple features available that make runtime execution so flawless. Multithreaded processing is supported in such way that makes the most of parallel processing resources, i.e. multiple CPUS, multicore CPUs and hyperthreading [13]. Compiler intrinsic functions are utilized to manually vectorize processing of data in processor’s pipeline. Moreover, Kakadu comes with built-in thread scheduler. Such implementation enables possibility of utilizing all computational resources close to 100%. In a 2007 the JasPer library was outperformed by Kakadu in terms of speed.

Supported features of Part 2 include arbitrary wavelet transform kernels and general multicomponent transforms. The Kakadu library offers extensive support for interactive mode of client–server applications. It is done thanks to the implementation of the most notable features from the JPIP (JPEG 2000 Internet Protocols) standard [13].

2.4.3 Reversible denoising and lifting based color component transformation

Methods such as reversible denoising and lifting based color component transformation aim to improve overall quality of image compression. Color space transformation in reversible manner is required by the lossless image compression framework used in the JPEG 2000 standard. The undesirable side effect of such transformation is contamination of transformed components with noise from other ones. Therefore, the compression ratio of given image is substantially diminished [17].

The work in this specific paper is aimed to remove correlation without increasing the noise. Therefore, a reversible denoising and lifting step (RDLS) was proposed which integrates usage of denoising filters into lifting step. New image component transformation is a result of applying RDLS to color space transformation. The main feature of this operation is being reversible despite involving the inherently irreversible denoising [17].

The main targets using application of RDLS to the RDgDb color space transformation with simple denoising filters are the JPEG-LS, JPEG 2000 and JPEG XR lossless modes of the standard algorithms. The images which have native optical resolution of acquisition devices benefit the most from this application. Improvement in terms of compression ratios is the most visible in subset of images which unmodified color space transformation either improved or worsened ratios in comparison to the untransformed image. On the average improvement is between 5% and 6% for such specific images. However, things differ for images from standard test-sets resulting in the improvement only up to 2.2% [17].

2.4.4 Skipping Selected Steps of DWT Computation

The other method of improving image compression, i.e. bitrates of lossless JPEG 2000 is skipping selected steps of discrete wavelet computation (SS-DWT). As a precondition fixed SS-DWT variants were employed. Then, heuristic was employed to select from the mentioned before variants best one for certain image [15].

The experiments on diverse set of images resulted in the improvement of bitrates vastly of non-photographic images. The entropy estimation of encoding effects was used to select the most feasible variant of applied fixed SS-DWT. It is especially important from the practical standpoint as time execution of such application is faster. Such compression scheme is compliant with the Part 2 of the JPEG 2000 standard as opposed to the general SS-DWT case [15].

The average improvement in terms of bitrate was around 5% for the entire test-set. Moreover, the time needed to perform compression of the image was only 3% great than the unmodified JPEG 2000 variant. However, the results are quite different across the set of photographic and non-photographic images. The first ones are improved on average by 0.5% while the latter by 14%. The heuristic can be further exploited to perform more modifications and result with improvement of bitrate up to 17.5%. These extra modifications include skipping the steps based on the actual bitrate rather than the estimated one and applying reversible denoising and lifting steps to SS-DWT. However it is achieved with great time penalty [15]. Furthermore, it has been evaluated that applying the fixed skipped steps discrete wavelet transform (fixed SS-DWT) variants with the lossless compression compliant with the Part 2 of JPEG 2000 standard can provide another improvements [16].

Chapter 3. Subject of the thesis

3.1 Solution to the problem

3.1.1 Proposed algorithm

3.1.2 Different variants of DWT decomposition

3.1.3 Selected filters

3.1.4 Entropy as JPEG 2000 coder estimator

3.2 Implementation details

3.2.1 Chosen programming language

3.2.2 Build environment

3.2.3 DWT interface

3.2.4 Testing

3.2.5 Parallel for

3.2.6 OpenCV

```cpp
#ifndef JPEG2000_PARALLEL_FOR_HPP
#define JPEG2000_PARALLEL_FOR_HPP

#include "config.hpp"

#include <concepts>
#include <functional>
#include <thread>
#include <vector>

namespace mgr {
namespace detail {

// clang-format off
template<typename Func, typename... Args>
// std::invocable<Func, Args...> should really be used in conjunction
// enable it with the release of libc++13 (support of <concepts> header)
concept no_returnable = std::same_as<std::invoke_result_t<Func, Args...>, void>;

template<typename Func, typename... Args>
concept returnable = !no_returnable<Func, Args...>;
// clang-format on

template<typename Func>
void parallel_for(std::size_t n_threads, const Func& func) {
    std::vector<std::thread> threads;
    threads.reserve(n_threads);
    for (std::size_t thread_idx{}; thread_idx < n_threads; thread_idx++) {
        threads.emplace_back(func, thread_idx);
    }
    for (auto& thread : threads) {
        thread.join();
    }
}
} // namespace detail
```

Listing 1: parallel_for.hpp: Base function

```cpp
template<typename Func>
requires detail::no_returnable<Func, std::size_t>
void parallel_for(std::size_t n_threads, std::size_t n_elements, Func&& func) {
    detail::parallel_for(
        n_threads,
        [n_threads, n_elements, func = std::forward<Func>(func)](
            std::size_t thread_idx) mutable {
            for (std::size_t i{thread_idx}; i < n_elements; i += n_threads) {
                func(i);
            }
        });
}

template<typename Func>
requires detail::returnable<Func, std::size_t>
auto parallel_for(std::size_t n_threads, std::size_t n_elements, Func&& func) {
    std::vector<std::invoke_result_t<Func, std::size_t>> result(n_elements);
    detail::parallel_for(
        n_threads,
        [n_threads, n_elements, func = std::forward<Func>(func), &result](
            std::size_t thread_idx) mutable {
            for (std::size_t i{thread_idx}; i < n_elements; i += n_threads) {
                result[i] = func(i);
            }
        });
    return result;
}
} // namespace mgr

#endif // JPEG2000_PARALLEL_FOR_HPP
```

Listing 2: parallel_for.hpp: User interface

Chapter 4. Experiments

This chapter presents the experiments. It is a crucial part of the thesis and has to dominate in the thesis. The experiments and their analysis should be done in the way commonly accepted in the scientific community (eg. benchmark datasets, cross validation of elaborated results, reproducibility and replicability of tests etc).

4.1 Methodology

- description of methodology of experiments

- description of experimental framework (description of user interface of research applications – move to an appendix)

4.1.1 Time execution

4.1.2 Image compression

4.2 Data sets

Brief description and visualization.

- description of data sets

4.3 Results

- presentation of results, analysis and wide discussion of elaborated results, conclusions

4.3.1 Reference results

4.3.2 Estimated best configuration

4.3.3 Comparison with reference

4.3.4 Time comparison

4.3.5 Visualized sample results

Chapter 5. Summary

5.1 Realized tasks

- Initial research in fields of image processing and compression, analysis of JPEG 2000 algorithm.
- More advanced research of algorithms such as DWT, SS-DWT, HP and JPEG 2000 implementation - Kakadu.
- Development of basic DWT implementation.
- Development of advanced 2D DWT implementation with possibility of skipping transformation of columns or rows.
- Setup of Continuous Integration system and implementation of DWT testing component
- Development of initial heuristics allowing to study the effects of DWT modifications compliant with the JPEG 2000 standard such as decompositions into sub-bands and usage of different filters
- Support of loading and storing both grayscale and color images.
- Initial implementation of multi-threaded heuristics.
- Conducting preliminary tests and selecting modifications or their variants to be included in the final heuristics.
- Development of multi-threaded optimized implementation of final heuristics.
- Research on final heuristics - comparison in terms of obtained compression ratio and time with: unmodified JPEG 2000, SS-DWT transformation and the transformation determined by an exhaustive search.

5.2 Results

- synthetic description of performed work

5.3 Conclusions

- conclusions
- Has the objective been reached?

5.4 Future development

- Future development, potential future research

Bibliography

[1] 2d dwt applied 2 times to an exemplary image. `https://en.wikipedia.org/wiki/File:Jpeg2000_2-level_wavelet_transform-lichtenstein.png`. [Latest available: 13.07.2021].

[2] C++ reference. `https://en.cppreference.com/`. [Latest available: 12.07.2021].

[3] Cmake. `https://cmake.org/`. [Latest available: 12.07.2021].

[4] Image compression. `https://en.wikipedia.org/wiki/Image_compression`. [Latest available: 12.07.2021].

[5] Jpeg 2000. `https://en.wikipedia.org/wiki/JPEG_2000/`. [Latest available: 12.07.2021].

[6] Kakadu. `https://kakadusoftware.com/`. [Latest available: 12.07.2021].

[7] Python. `https://www.python.org/`. [Latest available: 12.07.2021].

[8] Touradj Ebrahimi Athanassios Skodras, Charilaos Christopoulos. The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, 2001.

[9] J. O. Strömberg F. G. Meyer, A. Z. Averbuch. Fast adaptive wavelet packet image compression. *IEEE Transactions on Image Processing*, 2000.

[10] David A. Patterson John L. Hennessy. *Computer Architecture: A Quantitative Approach*. Elsevier, Incs, 2012.

[11] Martin Vetterli Kannan Ramchandran. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Processing*, 1993.

[12] Scott Meyers. *Effective Modern C++*. O'Reilly Media, 2014.

[13] Touradj Ebrahimi Peter Schelkens, Athanassios Skodras. *THE JPEG 2000 SUITE*. John Wiley & Sons, Singapore Pte. Ltd., 2009.

[14] M.V. Wickerhauser Ronald Coifman. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 1992.

[15] Roman Starosolski. Skipping selected steps of dwt computation in lossless jpeg 2000 for improved bitrates. *Plos one*, 2016.

[16] Roman Starosolski. A practical application of skipped steps dwt in jpeg 2000 part 2-compliant compressor. *Springer Link*, 2018.

[17] Roman Starosolski. Reversible denoising and lifting based color component transformation for lossless image compression. *Springer Link*, 2019.

[18] Roman Starosolski. Hybrid adaptive lossless image compression based on discrete wavelet transform. *Entropy*, 2020.

[19] Bjarne Stroustrup. *The C++ Programming Language*. Pearson Education, 2013.

[20] Bjarne Stroustrup. *A Tour of C++*. Pearson Education, 2013.

[21] D. Sundararajan. *Discretewavelet Transform: A Signal Processing Approach*. John Wiley & Sons, Singapore Pte. Ltd., 2015.

[22] Anthony Williams. *C++ Concurrency in Action*. Manning Publications, 2019.

Appendices

Technical documentation

List of abbreviations and symbols

JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
PACSs	Picture Archiving and Communication Systems
DICOM	Digital Imaging and Communications in Medicine
ISO	International Organization for Standardization
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
SS-DWT	Skipped Steps Discrete Wavelet Transform
HP	Histogram Packing
LL	Low and then low-pass filtered image
LH	Low and then high-pass filtered image
HL	High and then low-pass filtered image
HH	High and then high-pass filtered image
ppi	pixels per inch
MJ2	Motion JPEG 2000
JPWL	JPEG 2000 Wireless
DSP	Digital Signal Processing

Contents of attached CD

The thesis is accompanied by a CD containing:

- thesis (`pdf` file),
- source code of applications,
- data sets used in experiments.

List of Figures

List of Tables

List of listings