FUTURE OF JPEG XT: PRIVACY AND SECURITY

by

MAITRI SHAH

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2016

ACKNOWLEDGEMENT

Firstly, I would like to express my heartfelt gratitude to my mentor Dr. K. R. Rao for the support and motivation throughout my thesis. He has always believed in my capabilities and has guided me throughout my Masters Degree program. I am extremely grateful to Dr. Thomas Richter for providing me with a strong understanding of JPEG XT and also his continuous guidance in my research. I would like to thank Dr. Jonathan Bredow and Dr. Alan Davis for serving on my committee.

I would like to extend my appreciation to Deepak Hingole, Ripudaman Arora and Shrikant Tudavekar for sharing their knowledge in C++. I want to thank Aakash Shah, Ramesh Nair and Prayag Shah for their constant support and inspiration during the Masters journey, and my friends for providing me with valuable advice and help during the course of my thesis work.

I want to thank my family for trusting and supporting me during the course of my Masters and Thesis work. Last but not the least, I want to thank the Almighty without whose blessings this journey would have never started.

April 22, 2016

List of Acronyms

AIC: Advanced Image Coding

APP: Application Marker

AR: Augmented Reality

BMP: BitMaP

CABAC: Context Adaptive Binary Arithmetic Coding

CCITT: Consultative Committee for International Telephony and Telegraphy

DCT: Discrete Cosine Transform

DPCM: Differential Pulse Code Modulation

DSLR: Digital Single Lens Reflex

DWT: Discrete Wavelet Transform

EOI: End of Image

EPFL: École Polytechnique Fédérale de Lausanne

EXIF: Exchangeable Image File Format

FDCT: Forward Discrete Cosine Transform

G3 and G4: Group 3 and Group 4 of compression fax techniques

GIF: Graphics Interchange Format

HD: High Definition

HDR: High Dynamic Range

HDRI: High Dynamic Range Imaging

HVS: Human Visual System

ICC: International Color Consortium

IDCT: Inverse Discrete Cosine Transform

IDR: Intermediate Dynamic Range

IEC: International Electrotechnical Commission

IPR: Intellectual Property Rights

ISO: International Standards Organization

ITU-T: International Telecommunication Union (Telecommunication Standardization Sector)

JBIG: Joint Bi-level Image Experts Group

JFIF: JPEG File Interchange Format

JFXX: JFIF extension APP0 marker segment

JPEG: Joint Photographic Experts Group

JPEG AR: JPEG Augmented Reality

JPEG LS: JPEG Lossless

JPEG XR: JPEG eXtended Range

JPIP: JPEG 2000 Interactive Protocol

JPSEC: JPEG 2000 Secured

JPWL: JPEG 2000 for Wireless Applications

LDR: Low Dynamic Range

LOCO: Low Complexity Lossless Compression

log: Logarithm to the base 10 ($\log_{10}$)

ln: Natural logarithm to the base e ($\log_{e}$)

MR: Modified READ

MMR: Modified Modified READ

MPEG: Moving Picture Experts Group

NLT: Non Linear Transformation

OCR: Optical Character Recognition

OpenEXR: HDRI file format

PCT: Photo Core Transform

PNG: Portable Network Graphics

POT: Photo Overlap Transform

PPM: Portable PixMap

PSNR: Peak Signal to Noise Ratio

PSP: Photo Sharing Service Providers

RES: Residual

RLC: Run Length Coding

SOI: Start of Image

SSIM: Structural Similarity Metric

SPIFF: Still Picture Interchange File Format

TIFF:	Tag Index File Format
TMO:	Tone Mapping Operator
Trafo:	Transformation
VLSI:	Very Large Scale Integration
VUB:	Vrije Universiteit Brussel
WG:	Working Group
YIQ:	Luminance, In-Phase, Quadrature
YUV:	Luminance, Blue-Luminance, Red-Luminance

Abstract

FUTURE OF JPEG XT: PRIVACY AND SECURITY

Maitri Shah, MS

The University of Texas at Arlington, 2016

Supervising Professor: K.R. Rao

JPEG is *de facto* the most important and popular image format since 1992 which provides compressed images to a great extent. JPEG XT, the latest standardization of the JPEG committee, provides some amazing features like: backwards compatibility, support for High Dynamic Range (HDR), lossless and near-lossless coding, and alpha channel coding, which were not provided by the previous JPEG standards.

With the increase in the demand for photos- capturing, sharing, editing- privacy concerns have become the primary issue. Leakage of private photos to unauthorized users is unacceptable and hence the need for protecting and developing an algorithm to maintain the privacy and security is a must.

An algorithm is developed to help embed privacy in the JPEG XT software. The algorithm is developed in Microsoft Visual Studio by using OpenCV libraries and C++. It provides a method for face detection along with blurring the region of interest. The algorithm is tested for five different test sequences while detecting a maximum of 13 faces until now. It enhances the security of images which can be extended to social media image sharing. This algorithm brings us one step closer to protecting an individual’s life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the growing digital photography market, it has become a necessity to protect the images captured and shared on the internet. Due to the advancement in social media applications such as Facebook, Whatsapp, Snapchat, there is no privacy left in the world. Such applications encourage individuals to upload or share as many pictures needed in order to receive comments or likes from their friends. Moreover, social media has a great impact on every individual despite of ruining their privacy. Every image contains some sensitive and private information and some metadata too. Individuals do not realize the outcomes of their publicly shared images unless their account is hacked. Also, nowadays the images are stored in the cloud rather than private repositories, in order to receive more storage space. For years, the common image format used worldwide has been JPEG and hence it is vital for the JPEG committee to benefit their users by providing them with additional security functionalities.

1.2 Overview

The JPEG committee is developing a new standard called JPEG XT [1]. This new standardization effort provides support for High Dynamic Range (HDR) [2] images, lossless and near-lossless coding, and alpha channel coding. JPEG XT was brought into the market to overcome the flaw of backward compatibility which was not provided by other JPEG [3] standards such as JPEG 2000 [4] and JPEG XR [5].

The JPEG XT standard supports High Dynamic Range (HDR) by splitting the image data in a Low Dynamic Range (LDR) JPEG legacy compliant codestream and a residual stream, allowing to scale-up to an HDR decoded image, which is smartly embedded in the JPEG legacy codestream such that it does not jeopardize backward or forward compatibility. Mapping the HDR image onto an LDR image is performed by a tone-mapping algorithm, which is signaled in the codestream as look-up tables (LUT) or tone-mapping graphs to allow the decoder to perform an approximate tone-mapping operation. [6]

Images of higher bit depth (more than 8bpp) are used which can be represented in floating point as JPEG XT is compatible with 8 bit mode. The JPEG XT can reduce the file size more than 16 times to store the floating point HDR sample data than lossless OpenEXR while insuring the quality. [7]

While compressing the HDR images, the JPEG XT standard maintains the backward and forward compatibility by following a two layer coding which is described as follows:

- Base layer: A Tone Mapping Operator (TMO) generates an LDR image which is compressed in this layer.
- Enhancement layer: In this layer, coding of the residual HDR data takes place and uses maximum JPEG legacy coding tools.

Figure 1 shows the basic outline of the JPEG XT decoder [7].

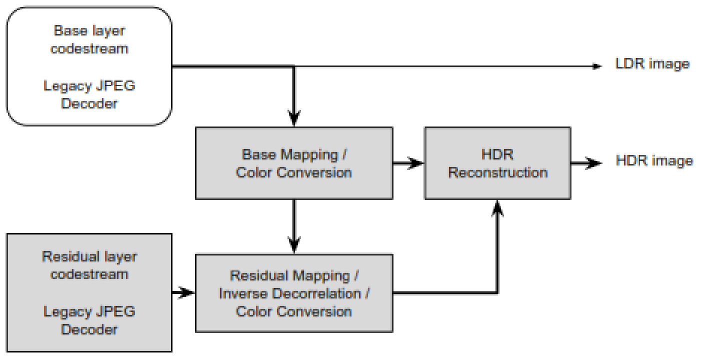

Base layer codestream
Legacy JPEG Decoder
LDR image
Base Mapping / Color Conversion
HDR Reconstruction
HDR image
Residual layer codestream
Legacy JPEG Decoder
Residual Mapping / Inverse Decorrelation / Color Conversion

Figure 1 Overview of the JPEG XT decoding process [7]

With the development of the new JPEG XT standardization, came the need for adding security and privacy to it. The JPEG XT has three potential real world extensions as of now: JPEG Privacy and Security, JPIP: Interactive Image Browsing and Delivery, and Recording Image Editing Operations. [6]

1.3 Thesis Outline

In this thesis, research is focused on one of the JPEG XT future extensions: JPEG Privacy and Security. An algorithm to encrypt and decrypt the images is developed using C++ and OpenCV libraries. This algorithm can be included as a privacy feature in the open source JPEG XT software. Several attempts were made to do so but they turned out to be unsuccessful due to some unresolved compatibility issues between the JPEG XT software and OpenCV libraries.

The thesis is explained in the form of the following chapters: The evolution and structure of JPEG is explained in Chapter 2. Chapter 3 describes JPEG XT in detail. Chapter 4 talks about the encryption and decryption. The results for different test conditions are discussed in Chapter 5. Chapter 6 explains the conclusions and future work.

Chapter 2

History of JPEG

2.1 Introduction to JPEG

The JPEG [3] standard (ISO/IEC 10918) was created in 1992 (latest version, 1994) as the result of a process that started in 1986. Though, this standard is generally considered as a single specification, in reality it is composed of separate parts as follows:

- Part 1 of JPEG (ISO/IEC 10918-1 | ITU-T Recommendation T.81) specifies the core coding technology and it incorporates many options for encoding photographic images.
- Part 2 defines the compliance testing.
- Part 3 defines a set of extensions to the coding technologies of Part 1, and via an amendment the Still Picture Interchange File Format (SPIFF) was introduced.
- Part 4 focuses on the registration of JPEG profiles, SPIFF profiles, SPIFF tags, SPIFF color spaces, SPIFF compression types, and defines the Registration Authorities.
- Part 5 specifies the JPEG File Interchange Format (JFIF).

Without any doubt, it can be stated that JPEG has been one of the most successful multimedia standards defined so far. [3]

JPEG is a dominant lossy compression algorithm for continuous-tone still images. JPEG compresses the image to occupy less memory by eliminating the data or inexactly representing the data in an image. The human visual system cannot observe colors at high frequencies (700 nm-100,000 km). JPEG removes such frequencies while compressing so that the human eyes cannot observe the change. Thus it is a good, but not a desirable, trade-off between memory and frequency. To reduce the memory drastically, the important frequencies are removed which in turn will lead to observable visual differences and degrade the image quality. JPEG compression works best on images with smooth color transitions [8].

JPEG offers distinct advantages in both picture quality and the responding file size to rival formats such as GIF (Graphics Interchange Format) and BMP (BitMaP). But sometimes JPEG suffers with the image quality as each successive image save, i.e. compression removes further information. The only solution to this problem is to use a lossless format, such as TIFF (Tagged Image File Format) for compression and then converting to the JPEG format after all the editing is completed. [9]

JPEG compression artifacts blend well into photographs with detailed non-uniform textures, allowing higher compression ratios. [9] But JPEG is not well suited for logos, line art and wide areas of flat color. [10]

JPEG is a *de-facto* standard in the digital photography industry. The pictures taken through the mobile phone cameras or high end cameras such as a DSLR, all provide compatibility with JPEG. The JPEG format has been successful in developing millions of pictures till date which are shared through the internet everyday. Social media websites such as Facebook and Flickr also allow the users to upload their pictures directly from the phone cameras too. According to Facebook, over 200 million images are uploaded online daily [10]. According to Statista over 75% of the online content is provided in the JPEG format [10].

JPEG format is a source of data that can be used for the purpose of detecting forged images and some forensic analysis to criminal-cases. JPEG also stores some technical information called “Metadata”. These Metadata tags contain information about quantization of matrices, Huffman code tables of full image. JPEG tags contain important information about the photo including shooting conditions and parameters such as light levels, shutter speed information, make or model of the camera and lens, lens focal length, flash usage, color profile information and geospatial information. [10]

JPEG was famous for high degree of compression, wherein the picture quality is high with a small degree of compression, suitable for full-color realistic images with contrast transitions. JPEG faced some drawbacks such as being less suitable for working with text or monochrome graphics with clear boundaries, no support for transparency and degradation of the image quality for every next step of compression. Despite JPEG’s drawbacks, it is still widely used in the industry which is shown in the Figure 2-1 [10].

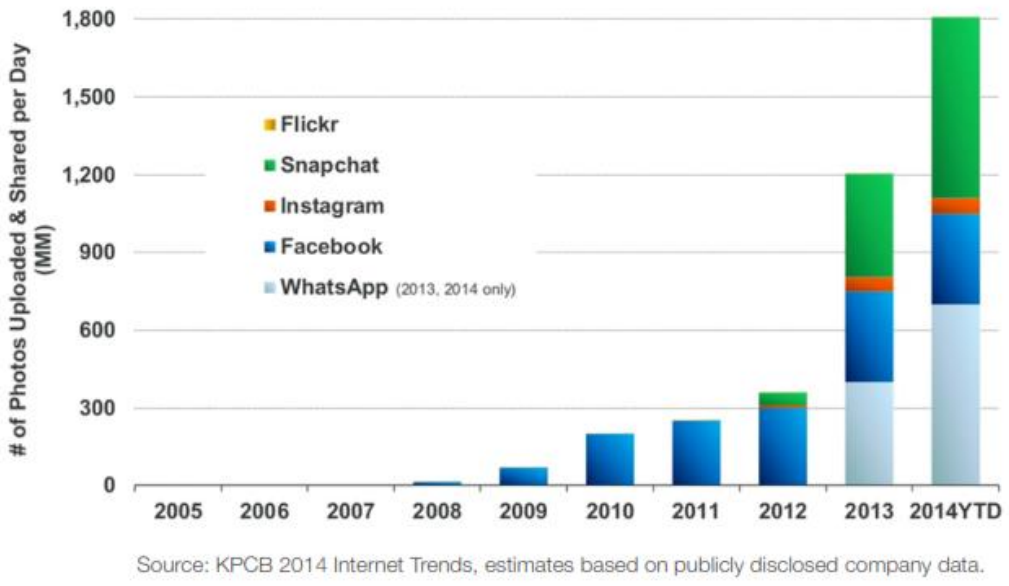


Figure 2-1 JPEG’s growing ecosystem [10]

2.2 JPEG Encoder and Decoder

Figure 2-2 outlines the JPEG encoder. Figures 2-3 and 2-4 show the key processing steps which are the heart of the DCT-based modes of operation. These figures illustrate the special case of single-component (grayscale) image compression. The essentials of DCT-based compression are the compression of a stream of 8x8 blocks of grayscale image samples. Color image compression can then be approximately regarded as compression of multiple grayscale images, which are either compressed entirely one at a time, or are compressed by alternately interleaving 8x8 sample blocks from each in turn. Each 8x8 input blocks makes its way through each processing step and yields the output in a compressed form into the data stream. [11] In Figure 2-2, Differential Pulse Code Modulation (DPCM) is done for the quantized DC coefficient whereas Run Length Coding (RLC) is done for the quantized AC coefficients.

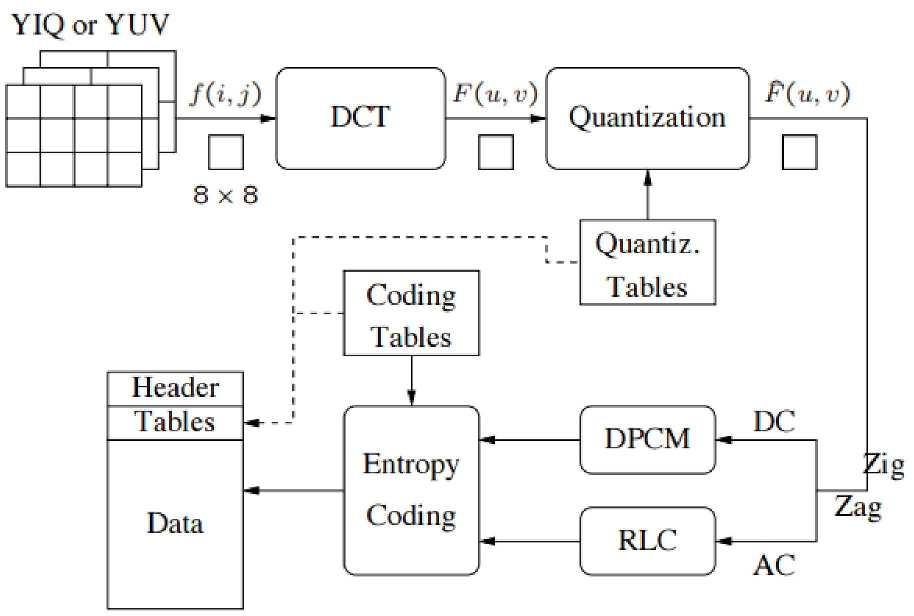


Figure 2-2 JPEG Encoder [12]

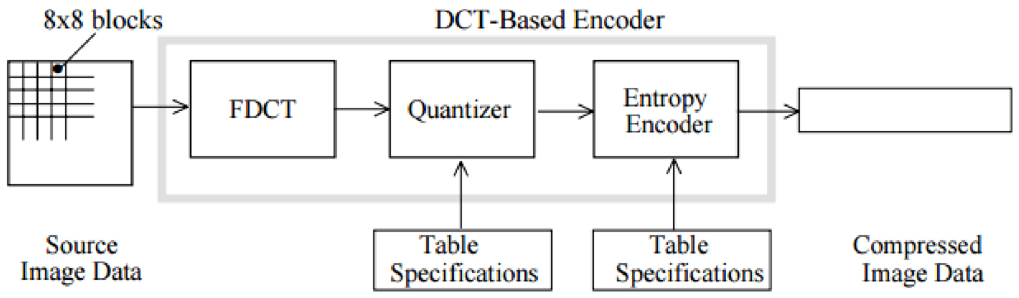


Figure 2-3 DCT-Based Encoder Processing Steps [11]

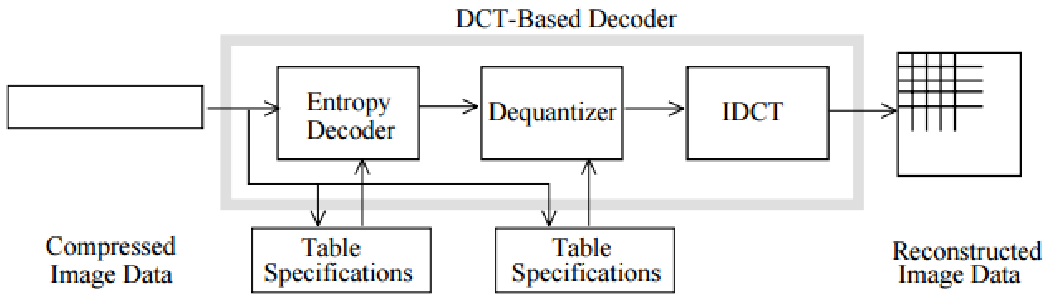


Figure 2-4 DCT-Based Decoder Processing Steps [11]

JPEG's proposed standard aims to be generic, to support a wide variety of applications for continuous-tone images. To meet the differing needs of many applications, the JPEG standard includes two basic compression methods, each with various modes of operation. A DCT-based method is specified for "lossy'' compression, and a predictive method for independent "lossless'' compression. JPEG features a simple lossy technique known as the Baseline method, a subset of the other DCT-based modes of operation. The Baseline method has been by far the most widely implemented JPEG method to date, and is sufficient in its own right for a large number of applications. This article provides an overview of the JPEG standard, and focuses in detail on the Baseline method. [11]

2.3 Advantages, Disadvantages and Applications

The emerging JPEG continuous-tone image compression standard is not a panacea that will solve the myriad issues which must be addressed before digital images will be fully integrated with all the applications that will ultimately benefit from them. For example, if two applications cannot exchange uncompressed images because they use incompatible color spaces, aspect ratios, dimensions, etc. then a common compression method will not help. However, a great many applications are “stuck” because of storage or transmission costs, because of argument over which (nonstandard) compression method to use, or because VLSI codecs are too expensive due to low volumes. For these applications, the thorough technical evaluation, testing, selection, validation, and documentation work which the JPEG committee members have performed is expected to soon yield an approved international standard that will withstand the tests of quality and time. As diverse imaging applications become increasingly implemented on open networked computing systems, the ultimate measure of the committee’s success will be when JPEG-compressed digital images come to be regarded and even taken for granted as “just another data type,” as text and graphics are today. [11]

2.4 Research Areas in JPEG

With JPEGs growing ecosystem and its popular demand, it is necessary for JPEG to overcome drawbacks. Some of the key areas for advancement are: data security, and licensing, personal privacy, file optimization, backward compatibility, data integrity, metadata, etc.

Despite the phenomenal success of the JPEG baseline system, it has several shortcomings that become increasingly apparent as the need for image compression is extended to such emerging applications as medical imaging, digital libraries, multimedia, internet and mobile. [13] Table 2-1 shows the comparison of JPEG with other standards.

Table 2-1 Comparison of JPEG with TIFF, GIF, PNG and RAW

	JPEG	TIFF	GIF	PNG	RAW
Synonym	Joint Photographic Experts Group	Tagged Image File Format	Graphic Interchange Format	Portable Network Graphics	
What does it do?	compresses to store a lot of information in a small-size file	uncompressed and thus contains a lot of detailed image data	compresses images, but the file cannot be made as small as a JPEG	created as an open format to replace GIF	The files are called raw because they haven’t been processed and therefore can’t be edited or printed yet
Compression	Lossy	No	Lossless	Lossless	No
Transparency	No	Yes	Yes	Yes	
Advantages	Smaller file size	Extremely flexible in terms of	Maintain the photo	Full range of color and	Contain a vast amount of

		color and content	original quality	better compression	uncompressed data. Converted to TIF before editing and color-correcting
Disadvantages	Bad for line drawings/ logos/ graphics	Not web-friendly	Extremely limited color range, suitable for web and photography	Never for print images and photographs Not compatible with software or applications.	
Recommended for	Photographs on web (easy to upload)	Photo software (eg. Photoshop) Page layout software (eg. Quark)	Animated graphics/icons	Static Graphics/icons Web images, text or line art	

Note: Adapted from http://www.ivanexpert.com/blog/2010/05/the-5-types-of-digital-image-files-tiff-jpeg-gif-png-and-raw-image-files-and-when-to-use-each-one/ & http://www.smartimage.com/whats-the-difference-between-gif-png-jpeg-and-tiff/

2.5 Evolution of JPEG

Some examples of standards created by ISO/IEC JTC1 SC29 Working Group 1 (WG 1) include the Joint Photographic Experts Group and Joint Bi-level Image experts Group. They are shown in the Figure 2-5 and are listed as follows: [3]

- JPEG: ISO/IEC 10918-1 | ITU-T Recommendation T.81
- JPEG 2000: ISO/IEC 15444-1 | ITU-T Rec. T.800 (2000)
- JPEG 2000 Extensions: ISO/IEC 15444-2 | ITU-T Rec. T.801 (2001)
- JPEG LS: ISO/IEC 14495-1:1999 | ITU-T Rec. T.87 (1998)
- JPEG LS Extensions: ISO/IEC 14495-2:2003 | ITU-T Rec. T.870 (03/2002)
- JPEG XR: ISO/IEC 10918-1 | ITU-T Rec. T.81 (2008)
- JPEG XT: ISO/IEC 10918-1 | ITU-T Rec. T.81
- JBIG: ISO/IEC 11544 | ITU-T Rec. T.82
- JBIG2: ISO/IEC 14492 | ITU-T Rec. T.88
- AIC: ISO/IEC 29170-1:2 | ITU-R BT.500-11
- JPSearch
- JPEG AR

Figure 2-5 JPEG Family of Standards [10]

2.5.1 JPEG 2000

JPEG 2000 (JP2) [4] is an image coding system which was developed in 2000 by the Joint Photographic Experts Group committee. JP2 uses state-of-the-art compression techniques which are based on wavelet technology. Due to its architecture, it has numerous applications from portable digital cameras to advanced pre-press, medical imaging and many other key sectors. The main advantage offered by JPEG 2000 is the significant flexibility of the codestream which provides scalability after compression. The aim of JPEG 2000 is not only improving compression performance over JPEG but also adding/improving features such as scalability and editability. [14]

JPEG 2000 refers to all parts of the standard. Below is the list of current parts that make up the complete JPEG 2000 suite of standards [4]:

- Part 1, Core coding system
- Part 2, Extensions
- Part 3, Motion JPEG 2000
- Part 4, Conformance

- Part 5, Reference software
- Part 6, Compound image file format
- Part 7, has been abandoned
- Part 8, JPSEC
- Part 9, JPIP
- Part 10, JP3D
- Part 11, JPWL
- Part 12, ISO
- Part 13,
- Part 14

The features offered by JPEG 2000 are all developed from a single algorithm, thus proving that JPEG 2000 is not just a compression algorithm but is much more than that.

The JPEG 2000 standard was developed to meet the demands for efficient, flexible, and interactive image representations. In particular, an important goal of JPEG 2000 manuscript is that all implementations, from the simplest to the most sophisticated, should be able to effectively interact with the same, efficiently compressed image, regardless of the resolution, bit depth, bit rate, or number of components in that image. With few exceptions, this goal has been achieved. This is a direct consequence of JPEG 2000's emphasis on scalable compressed representations, as described below. [15]

JPEG 2000 was known for some good features: a single approach to lossy and lossless compression, progressive display with scalable resolution and quality, compressing image to specified size or quality, support for domain-specific metadata in file format, low loss across multiple decompress-compress cycles and lastly error resilience. [16] Region of interest (ROI) coding is an important feature provided by JPEG 2000. It is important in applications where certain parts of an image are of a higher importance than the rest of the image. In such cases, the ROI is decoded with higher quality and/or spatial resolution than the background. There are three mechanisms available in JPEG2000 to encoded and decoded images with varying spatial detail: tiling, code-block selection and coefficient scaling. [68]

JPEG 2000 provided reduced costs for storage and maintenance, enhanced handling of large images and enabled new opportunities. But it did have some major drawbacks such as requiring encoders/decoders which are complex and computationally demanding. In comparison with JPEG, JPEG 2000 only produces ringing artifacts, manifested as blur and rings near edges in the image, while JPEG produces both ringing artifacts and 'blocking' artifacts, due to its 8×8 blocks. [14]

Figure 2-6 shows the fundamental building blocks of JPEG 2000. It consists of a pre-processing block followed by Discrete Wavelet Transform (DWT) block, Uniform Quantization block, Arithmetic Coding (tier-1 coding) block and lastly a Bit-stream Organization block (tier-2 coding). The input image to JPEG 2000 may contain one or more components. Although a typical color image would have three components (e.g., RGB or YC_bC_r), up to 16,384 (2^{14}) components can be specified for an input image to accommodate multi-spectral or other types of imagery. The sample values for each component can be either signed or unsigned integers with a bit-depth in the range of 1–38 bits. Given a sample with a bit-depth of B bits, the unsigned representation would correspond to the range (0, 2^{B-1}), while the signed representation would correspond to the range (-2^{B-1}, $2^{B-1}-1$). The bit-depth, resolution, and signed versus unsigned specification can vary for each component. If the components have different bit-depths, the most significant bits of the components should be aligned to facilitate distortion estimation at the encoder. [13]

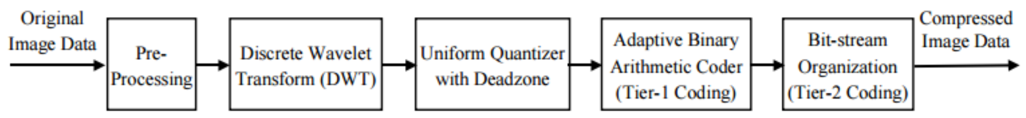

Original Image Data
Pre-Processing
Discrete Wavelet Transform (DWT)
Uniform Quantizer with Deadzone
Adaptive Binary Arithmetic Coder (Tier-1 Coding)
Bit-stream Organization (Tier-2 Coding)
Compressed Image Data

Figure 2-6 JPEG 2000 fundamental building blocks [13]

Part 1 of the JPEG 2000 standard defines a file format referred to as JP2. Although this file format is an optional part of the standard, it is expected to be used by many applications. It provides a flexible, but restricted, set of data structures to describe the coded image data. Part 2 of the standard defines extensions to the JP2 file format, encapsulated in an extended file format called JPX. These extensions increase the colorspace flexibility by providing more enumerated color spaces (and also allows vendors to register additional values for colorspaces) as well as providing support for all ICC profiles. [13]

In addition to highly scalable compressed data streams, JPEG 2000 offers numerous advantages over its predecessor, JPEG. Among these are [15]

- Improved compression efficiency;
- Progressive lossy to lossless performance within a single data stream;
- The ability to resequence compressed data to suit a wide range of different applications;
- The ability to arbitrarily crop images in the compressed domain without compression noise buildup;

- The ability to enhance the quality associated with selected spatial regions in selected "quality layers";
- The ability to work with truly enormous images without breaking them into independently compressed "tiles."

2.5.2 JPEG- LS

JPEG-LS [17], also known as lossless JPEG, was defined to address the need for effective lossless and near-lossless compression of continuous-tone still images. This standard can be broken into two parts: ISO/IEC 14495-1:1999 | ITU-T Rec. T.87 (1998), defining the core technology and ISO/IEC 14495-2:2003 | ITU-T Rec. T.870 (03/2002), containing the extensions. JPEG-LS is especially suited for low-complexity hardware implementations of very moderate complexity, while at the same time providing state-of-the-art lossless compression performance. [17]

JPEG-LS was developed with the aim of providing a low-complexity lossless and near-lossless image compression standard that can offer better compression efficiency than lossless JPEG. [18]

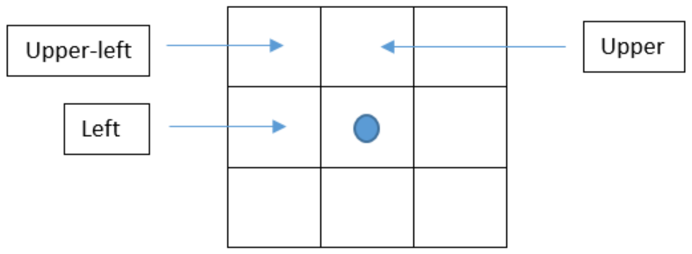


Figure 2-7 Nearest three neighbors

Lossless JPEG (a 1993 addition to the JPEG standard) makes use of a predictive scheme which includes the nearest three neighbors: upper, left and upper-left, as shown in the Figure 2-7. The Lossless data compression is divided into two distinct and independent components: modeling and coding. In the lossy mode of JPEG-LS, termed “near-lossless,” the reconstructed image sample values differ by a small preset amount from the corresponding values in the original image. For multicomponent (color) images, the JPEG-LS syntax supports both interleaved and non-interleaved (i.e., component by component) modes. For some color spaces (e.g., an RGB representation), good decorrelation can be obtained through simple lossless color transforms as a pre-processing step to JPEG-LS. [19]

The overall simplicity of LOCO-I/JPEG-LS is attributed to its success in matching the complexity of the modeling and coding units, combining simplicity with the compression potential of context models, thus “enjoying the best of both worlds.” The main blocks of the algorithm are depicted in Figure 2-8, including the causal template actually employed. The shaded area in the image icon at the left of the figure represents the scanned past data xt, on which prediction and context modeling are based, while the black dot represents the sample xt+1 currently encoded. The switches labeled mode select the operation in “regular” or “run” mode. [19]The JPEG-LS decoder performs inverse operations of the encoder in the reverse order.

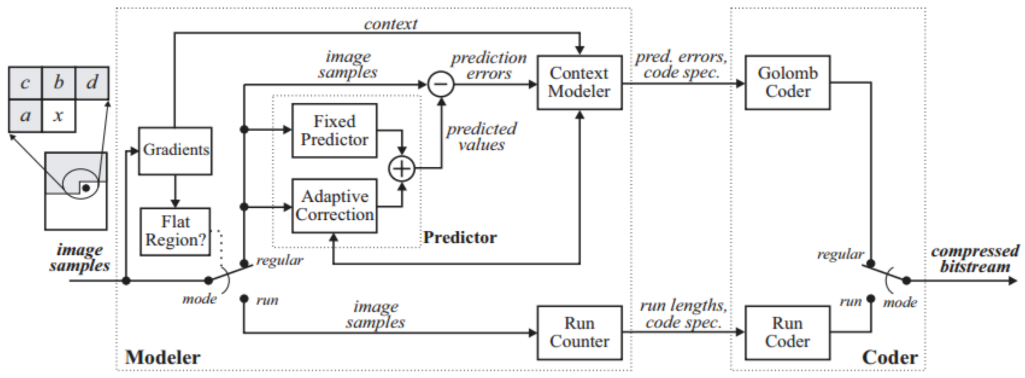


Figure 2-8 JPEG-LS encoder block diagram [19]

Lossless JPEG was never widely adopted despite its popularity in medical imaging and digital cameras (to compress raw images). Lossless JPEG is a mode of operation of JPEG which exists because the discrete cosine transform (DCT) based form cannot guarantee that encoder input would exactly match decoder output.

2.5.3 JPEG XR

JPEG XR [5], i.e. JPEG extended range, is a still-image compression standard and targets the continuous tone images such as photographic images, based on technology originally developed and patented by Microsoft under the name HD Photo (formerly Windows Media Photo).] [20] JPEG XR was designed to provide a special support to the needs of the emerging high dynamic range (HDR) imagery applications.

JPEG XR was developed to overcome the difficulties faced by the photographers by switching to raw image encoding over JPEG baseline encoding. By then the 20-year old JPEG technology had reached its limits for providing support to the newer inventions whereas the raw encoding had issues such as very high storage capacity requirements (camera specific) and lacked interoperability. Thus, the “extended range” provided support for new applications which the original JPEG standard could not.

The JPEG XR standard provides a practical coding technology for a broad range of applications with excellent compression capability and important additional functionalities. [5]

The JPEG XR specification enables greater effective use of compressed imagery with this broadened diversity of application requirements. JPEG XR supports a wide range of color formats including monochrome, RGB, CMYK and n-component encodings using a variety of unsigned integer, fixed point, and floating point decoded numerical representations with a variety of bit depths. The primary goal is to provide a compressed format specification appropriate for a wide range of applications while keeping the implementation requirements for encoders and decoders simple. JPEG XR combines the benefits of optimized image quality and compression efficiency together with low-complexity encoding and decoding implementation requirements. [21]

The coding flow of JPEG XR is shown in Figure 2-9. First of all, color space transformation is applied to an input image to transform RGB image to YUV image. Then a transform called photo core transform (PCT) is applied to decompose an image into frequency components. PCT is a lossless nonlinear 4x4 transform operation inspired by a 4x4 DCT. The PCT is defined on integers. The entire 4x4 PCT transform process consists of cascade of 2x2 Hadamard 2D transforms combined with 1D and 2D rotate operations. [69] The PCT is applied to a rectangular area called a macroblock. To reduce block noise, this occurs around macroblock boundaries, a transform called photo overlap transform (POT) is used with the PCT. These transforms are based on lapped biorthogonal transforms. Next, the transformed coefficients are quantized. In an inter-block coefficient prediction process, the quantized coefficients of a block are replaced by prediction errors to enhance compression rate. After the inter-block coefficient prediction, an adaptive scanning is processed and coefficients are rearranged from two-dimension form to one dimensional form. Finally, the scanned coefficients are entropy coded using adaptive Huffman tables. [22]

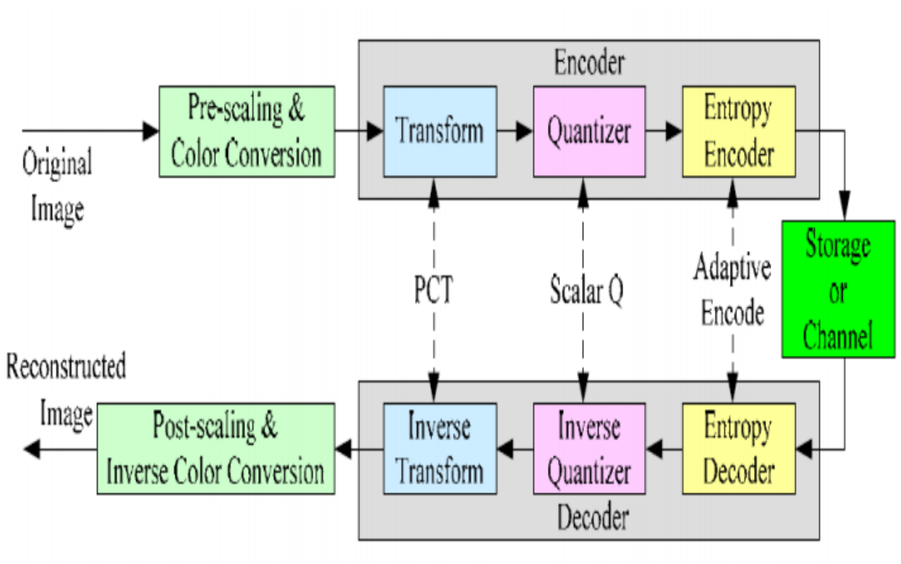


Figure 2-9 The data flow of JPEG XR encoding [22]

It also provides an extensive set of additional functionalities, including: [21]

- high compression capability,
- low computational and memory resource requirements,
- lossless and lossy compression,
- image tile segmentation for random access and large image formats,
- support for low-complexity compressed-domain image manipulations,
- support for embedded thumbnail images and progressive resolution refinement – embedded codestream scalability for both image resolution and fidelity,
- alpha plane support,
- bit-exact decoder results for fixed and floating point image formats.

The algorithm provides native support for both RGB and CMYK color types by converting these color formats to an internal luma-dominant format through the use of a reversible color transform. In addition, YUV, monochrome and arbitrary n-channel color formats are supported. [21] JPEG use a linear transformation from RGB to YCbCr specified as follows: [22]

$$Y = 0.299R + 0.587G + 0.114B \tag{1}$$

$$C_b = -0.1687R - 0.3313G + 0.5B + 2^{Ps-1} \tag{2}$$

$$C_r = 0.5R - 0.4187G - 0.0813B + 2^{Ps-1} \tag{3}$$

In Equations "1", "2", "3" Ps denotes the precision of sample like Ps = 8 means that each color component has 8 bits. The transform is a little bit lossy due to round-off error. JPEG XR specifies a lossless color space conversion, given by: [22]

$$Y = \frac{G + [R - G + \left[\frac{B-R}{2}\right]]}{2} \tag{4}$$

$$C_b = -[R - G + \left[\frac{B-R}{2}\right]] \tag{5}$$

$$C_r = B - R \tag{6}$$

JPEG XR is an integer-based standard. Equations "4", "5", "6" do not contain the rounding operation; therefore the inverse operation can be performed in decoder without loss. [22]

The transforms employed are reversible; both lossless and lossy operations are supported using the same algorithm. Using the same algorithm for both types of operation simplifies implementation, which is especially important for embedded applications. A wide range of numerical encodings at multiple bit depths are supported: 8-bit and 16-bit formats, as well as additional specialized packed bit formats, are supported for both lossy and lossless compression. (32-bit formats are supported using lossy compression.) Up to 24 bits are retained through the various transforms. While only integer arithmetic is used for internal processing, lossless and lossy coding are supported for floating point and fixed point image data – as well as for integer image formats. [21]

JPEG XR offers a few advantages over JPEG: it offers greater dynamic range- the span between the brightest bright and darkest darks in a photo, uses a more efficient compression algorithm that provides either twice the image quality as JPEG of the same file size, or half the file size for the same quality, according to Microsoft. Unlike JPEG, setting JPEG XR to record at its highest quality level loses no information to compression artifacts. [23] JPEG XR also provides support for more color accuracy, transparency map support, compressed domain image modification and metadata support. On comparison to JPEG 2000, it was observed that JPEG XR required low computational resources and storage capacity. Figure 2-10 shows how JPEG XR bridges the gap between JPEG and JPEG 2000 for performance and complexity.

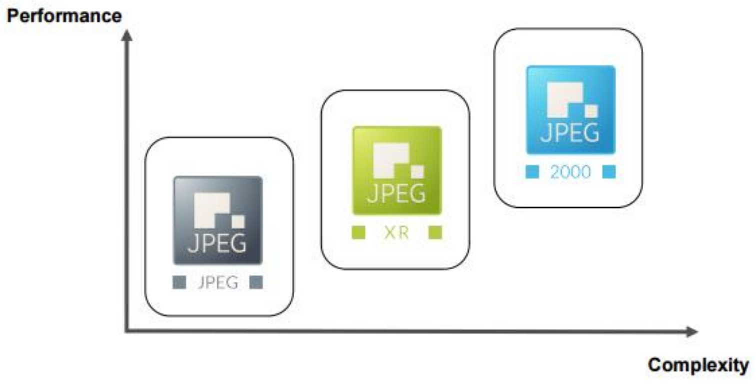


Figure 2-10 JPEG XR bridges the gap[10]

2.5.4 JBIG

JBIG [24] is a lossless image compression standard from the Joint Bi-level Image Experts Group. JBIG has developed an ISO/IEC standard 11544 and as ITU-T recommendation T.82 for lossless compression of a bi-level image. It is used for coding greyscale and color images with limited numbers of bits per pixel. It can be regarded as a form of facsimile encoding, similar to Group 3 or Group 4 fax, offering between 20 and 80% improvement in compression over these methods (about 20 to one over the original uncompressed digital bit map) [24].

Optimized for compression of black and white images, JBIG can also be applied to gray images of about six bits per pixel, or sixty-four gray levels by encoding each bit plane separately, while maintaining compression efficiency. JBIG uses a ten-pixel context to estimate the probability of the next pixel being white or black. It then encodes the next pixel with an arithmetic coder based on that probability estimate. The more heavily biased toward one color, the more the rate can be reduced below one bit per pixel, and the greater the compression ratio. [25]

JBIG is intended to completely replace the less efficient MR (Modified READ) and MMR (Modified Modified READ) compression algorithms used by the CCITT Group 3 (G3) and Group 4 (G4) data transmission protocols, respectively. In 1995, the International Telecommunication Union (ITU) proposed an extension of the G3 and G4 standards to allow the use of JBIG-compressed image data in conjunction with these protocols. [26]

JBIG achieves these impressive compression ratios by adapting to the information content of the image data being encoded. An adaptive arithmetic coder is used to predict and code future data symbols based on the characteristics of the data currently being encoded. G3 and G4 however, are non-adaptive and use the same fixed patterns and algorithms to encode all image data regardless of the content. [26]

JBIG also supports both sequential and progressive encoding methods. Sequential encoding reads data from the top to bottom and from left to right of an image and encodes it as a single image. Progressive encoding allows a series of multiple-resolution versions of the same image data to be stored within a single JBIG data stream. In contrast, G3 and G4 only support sequential coding at a fixed resolution. [26]

JBIG is platform-independent and implements easily over a wide variety of distributed environments. It achieves excellent compression ratios on bi-level images, and it is capable of efficiently encoding some types of color and gray-scale images as well. JBIG's progressive encoding capabilities appear to make it the obvious choice for transmitting and storing bi-level (two levels only) information on networked environments, such as the World Wide Web. [26]

To achieve the highest compression ratios possible, LEADTOOLS takes full advantage of all the functional blocks of the JBIG compression standard: [27]

- The Adaptive Arithmetic Encoder, which represents the heart of JBIG, predicts and encodes future data symbols based on the data that is currently being encoded.
- Adaptive Template Block provides improved compression efficiency (up to 80%) for images rendered with halftoning.
- Typical Prediction Block skips the encoding of regions of solid pixels to improve compression.
- Resolution Reduction Block generates a lower resolution version of the image data.

- Deterministic Prediction Block skips the encoding of data that can be determined by the neighboring pixels in a lower resolution version of the data.

LEADTOOLS supports sequential and random-access JBIG2. LEADTOOLS supports the various JBIG2 flavors of text and generic region encoding and decoding procedures based on sequential coding of the image pixels using arithmetic coding. Saving images in JBIG2 results in smaller image file sizes when compared to other industry standard compressed formats such as JBIG, CCITT G3 and G4. JBIG2 can compress bitonal images 2 - 5 times better than the same image compressed with CCITT G4 compression.

JBIG2: This is a more recent standard proposed by the Joint bi-level Experts Group. It uses a soft pattern matching approach to provide a solution to the problem of substitution errors in which an imperfectly scanned symbol is wrongly matched to a different symbol, as frequently observed in Optical Character Recognition (OCR). JBIG2 codes the bitmap of each mark, rather than its matched class index. In case a good match cannot be found for the current mark, it becomes a token for a new class. This new token is then coded using JBIG1 with a fixed template of previous pixels around the current mark. The JBIG2 standard is seen to be 20% more efficient than the JBIG1 standard for lossless compression. [28]

JBIG included arithmetic coding (QM coder), context-based prediction and progressive compression. [29] The compression ratio of JBIG was found to be the best of the available standardized methods back then. Despite this, JBIG did not achieve widespread use. The reason for this is mostly the issue of the patented QM-coder. To alleviate these difficulties JBIG organization developed a new bi-level image compression standard called JBIG2. Since then JBIG was also known as JBIG1. JBIG2 being a successor, offered significant advantages over other compression formats such as [24]:

- Large increases in compression performance (typically 3-5 times smaller than Group 4, 2-4 times smaller than JBIG1)
- Special compression methods for text, halftones, and other binary image content
- Lossy and lossless compression
- Multi-page document compression
- Flexible format
- High-performance decompression

JBIG2 was the first lossy bi-level image compression standard which supported three basic coding modes: Generic, Halftone and Text.

It has been seen that JBIG performs worst with textual images. Hence, in order to increase the performance, it is perceived that the most common symbols need to be coded into separate symbol libraries. [30]

2.5.5 AIC

JPEG provides very good quality of reconstructed images at low or medium compression i.e. high or medium bit rates respectively), but it suffers from blocking artifacts at high compression (low bit rates).

Advanced Image Coding (AIC) [31] is a still image compression system which combines the intra frame block prediction from H.264 with a JPEG-based discrete cosine transform followed by context adaptive binary arithmetic coding (CABAC). It performs much better than JPEG and has the best performance at low bit rates. AIC encodes color images and performs much better than JPEG and close to JPEG 2000. The aim of AIC is to provide better image quality with reduced complexity. It is also faster than existing JPEG 2000 codecs. [32]

For low resolution images, M-AIC performs better than JPEG 2000 and AIC. JPEG-XR and JPEG-LS has similar performance for all the image resolutions. Based on SSIM [33] simulations, it is observed that M-AIC, JPEG 2000 and JPEG-XR give almost the best performance whereas JPEG is closer to the above codecs in performance (visual quality, PSNR etc.) with low complexity and JPEG-LS outperforms the above codecs in the higher bit rate range. [32]

The AIC, shown in Figure 2-11 is developed with the key concern of eliminating the artifacts, thereby increasing quality. The predictor is composed of five parts including IDCT, inverse quantization, Mode Select and Store, Block Predictor and an Adder. The function of the predictor is to predict the current block to be encoded with the previously decoded blocks of the upper row and the left column. AIC uses CABAC entropy coding which uses position of the matrix as the context; while the M-AIC takes up Huffman coding and adaptive arithmetic coding in combination in order to achieve similar performance and also to reduce complexity. [32]

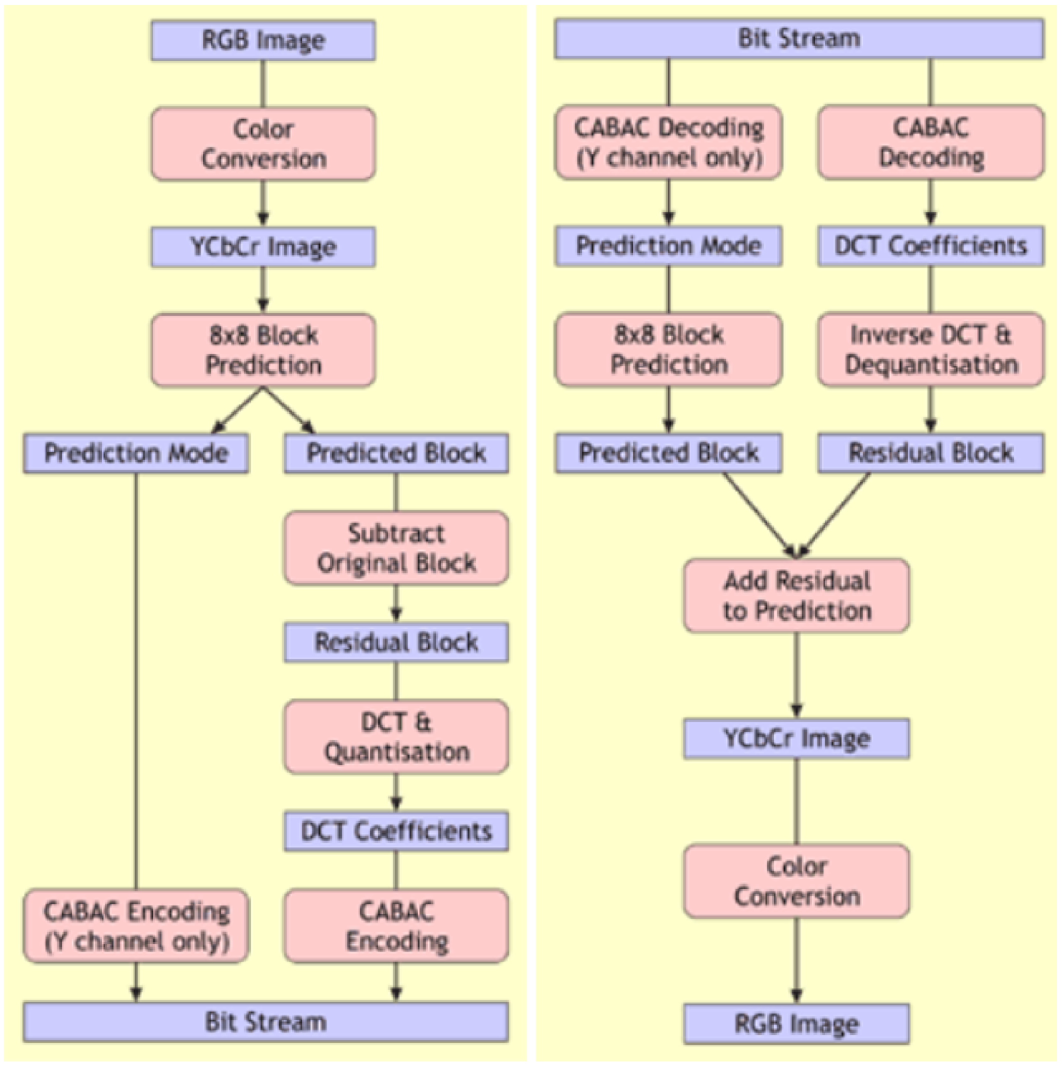


Figure 2-11 The process flow of the AIC encoder (left) and decoder (right) [32]

The JPEG 2000 encoded image has the best perceptual quality, although it tends to smooth and blur the image quite a bit, resulting in loss of detail. The JPEG image is clearly the worst, and AIC is somewhere in between. The objective quality of the AIC image, expressed as the Peak Signal to Noise Ratio (PSNR), is even higher than the JPEG 2000 image. The reason behind this is that AIC preserves the details better. AIC is also simpler than JPEG 2000. The AIC source code is optimized for clarity and readability, and not for speed. [23]

2.5.6 JPSearch

JPSearch [34] (ISO/IEC 24800) is a standardization initiative within the Joint Photographic Experts Group (JPEG) committee. JPSearch aims at providing a standard for interoperability of still image search and retrieval systems. It provides an abstract search framework architecture that contains a set of interfaces and protocols.

The idea of developing a standard for image search and retrieval had become necessary due to the growing camera market. Providing an abstract framework of search architecture that decouples the components of image search and a standard interface between these components became the objectives of the JPSearch standard. The idea of searching images using metadata and developing related functionalities had been exchanged with the MPEG working group. [35]

The need for metadata interoperability is evident within all phases of the multimedia life cycle that constitutes the necessary phases, from media production to storage, retrieval, and consumption. Within these, a special focus lies on the relation between media resources, metadata, and users. The life cycle clearly shows the need for harmonization because the multimedia application domain is way too large to have one application to manage all the facets of the defined phases. [36]

In the JPSearch framework, interoperability can be defined in different ways: between self-contained vertical image search systems providing federated search which is an information retrieval technology, between layers of an image search and retrieval system so that different modules can be supplied by distinct vendors, or at the metadata level such that different systems may add, update, or query metadata. Below is the list of current parts that make up the complete JPSearch suite of standards. [34]

- Part 1, Global architecture
- Part 2, Schema and Ontology
- Part 3, Query Format
- Part 4, File Format
- Part 5, Data Interchange Format
- Part 6, Reference Software

The JPSearch standard aimed at solving three main problems: [35]

- Lack of the ability to reuse metadata
- Lack of a common query format and search semantics
- Lack of a common format for handling context in searching

Figure 2-12 shows the architecture of JPSearch Framework. When a query is composed, it can be represented by the Input Query of JPSearch Query Format defined in Part 3, and is marked as (1) in the figure. If the metadata given in the query is defined by a metadata schema other than the JPSearch Core metadata, it can be translated into an instance of the JPSearch core metadata, shown as (8) in the figure, by following the interpretation of the translation rules defined in Part 2 and represented as (2). The query can be sent directly or after going through the query preprocessor to the database. The database can be a collection of images and metadata as shown as (5) in the figure or a collection of images in which metadata is embedded using the file format defined in Part 4 as shown as (4). The database or the collection of images with metadata can be exchanged using the interchange format defined in Part 5 as shown as (3) in the figure. The image file format defined in Part 4 supports heterogeneous metadata instances and is marked as (12) to be embedded in images in JPEG or JPEG 2000 format. These heterogeneous metadata can have essential interoperability through the translation rules defined in Part 2 as marked by (7)(9)(10)(11). The processed query result (6) is formed into a query output as defined in Part 3 and returned to the user. [35]

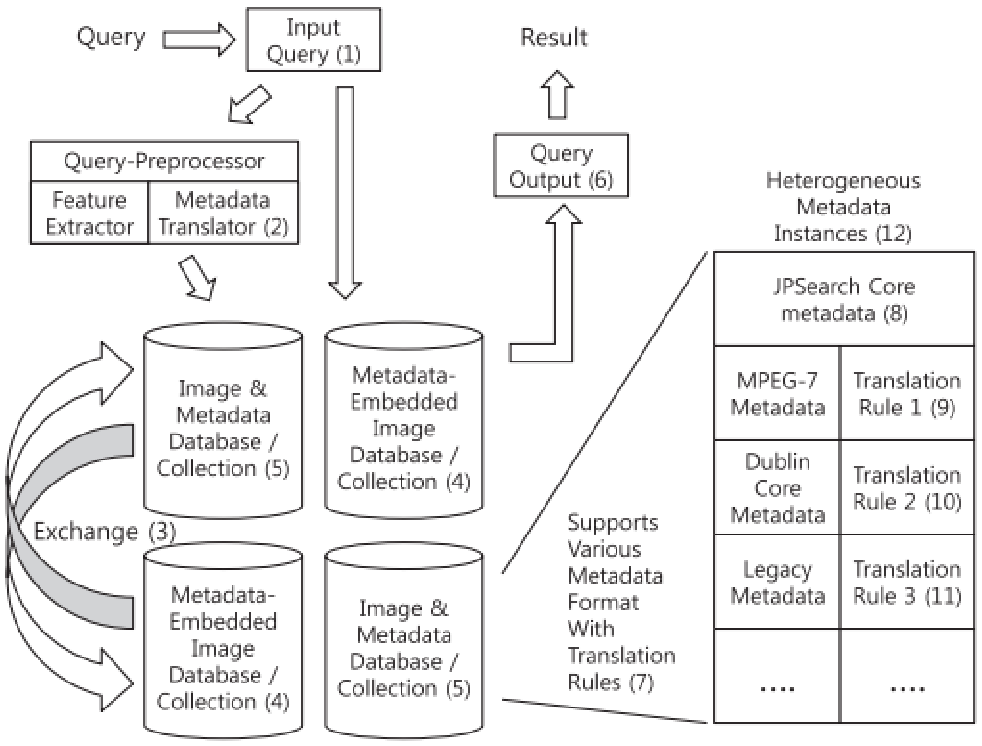

Query
Input Query (1)
Result
Query-Preprocessor
Feature Extractor
Metadata Translator (2)
Query Output (6)
Heterogeneous Metadata Instances (12)
JPSearch Core metadata (8)
Image & Metadata Database / Collection (5)
Metadata-Embedded Image Database / Collection (4)
MPEG-7 Metadata
Translation Rule 1 (9)
Dublin Core Metadata
Translation Rule 2 (10)
Exchange (3)
Supports Various Metadata Format With Translation Rules (7)
Legacy Metadata
Translation Rule 3 (11)
Metadata-Embedded Image Database / Collection (4)
Image & Metadata Database / Collection (5)
....
....

Figure 2-12 Structure of JPSearch framework [35]

The basic processes that are supported by JPSearch are as follows: [36]

- Image collection creation/ maintenance
- Image search/retrieve
- Intrinsic metadata storage

- Repository synchronization

Some of the major advantages of JPSearch are as follows: [35]

- JPSearch provides tools for interoperable interpretation of heterogeneous metadata supporting social tagging. By registering their own schema and translation rules to the authority, based on the JPSearch core metadata and translation rule description language, people can add their own metadata such as key words or text description to the existing images and share the images with metadata;
- JPSearch provides unified messages between client and servers for query and result, called Query Format, to allow independent development of applications and standardized interfaces enabling simple aggregation service of multiple database services;
- JPSearch provides interoperable file format, within which arbitrary number of metadata can be embedded, based on JPEG and JPEG 2000 file format, and provides straightforward binding mechanism of image and corresponding metadata. Using the JPSearch file format, multiple instances of metadata can be embedded inside the image file and transferred between devices without losing information;
- JPSearch defines interchange format for exchange of data between JPSearch image repositories.

One target group of applications that can benefit from adopting JPSearch is online image sharing platforms, such as Flickr or Picasa. These repositories can support the query format to provide third party applications access to their content. The interchange format can be used to facilitate import and export of collections. Adopting the metadata facilities assures preservation of all metadata. Client applications, e.g. mobile applications, can also benefit from adopting JPSearch. By supporting the query format for image search, they can support any compliant repository in a single application. Advanced features such as visual search can be integrated to interact with interoperable servers. [37]

2.5.7 JPEG AR

Augmented Reality (AR) [38] is a view of real world mixed with virtual objects/world. JPEG AR - a newly launched standardization initiative in 2012, aims at ensuring interoperability in the context of augmented reality applications while exchanging still image data. Existing AR frameworks typically provide interoperability within their proprietary ecosystems but exclude interoperability with non-native services. Due to the advent of software and hardware, AR applications have rapidly developed and are distributed to many application domains, such as commerce, education, entertainment, gaming, etc. Despite all of these efforts, the AR market has not grown well due to the absence of standardized metadata, interface, and concepts. JPEG AR is being designed to enable interoperability in distributed heterogeneous AR environments with a particular focus on image-based augmented reality systems. [38]

JPEG-AR has different standardization approaches and scope of standardization to existing standards as follows: - Interface to connects AR application components, - Metadata to describe AR application’s behavior, - File Format to embed above information. [38]

JPEG AR provides the signaling syntax between AR system components, a file format and AR application descriptors. JPEG AR integrates support for earlier JPEG standards such as JPEG, JPEG 2000, JPSearch. [38]

Figure 2-13 depicts the JPEG AR architecture. JPEG AR defines the interfaces between AR components, such as sensor, recognizer, tracker, event handler, and renderer. [39]

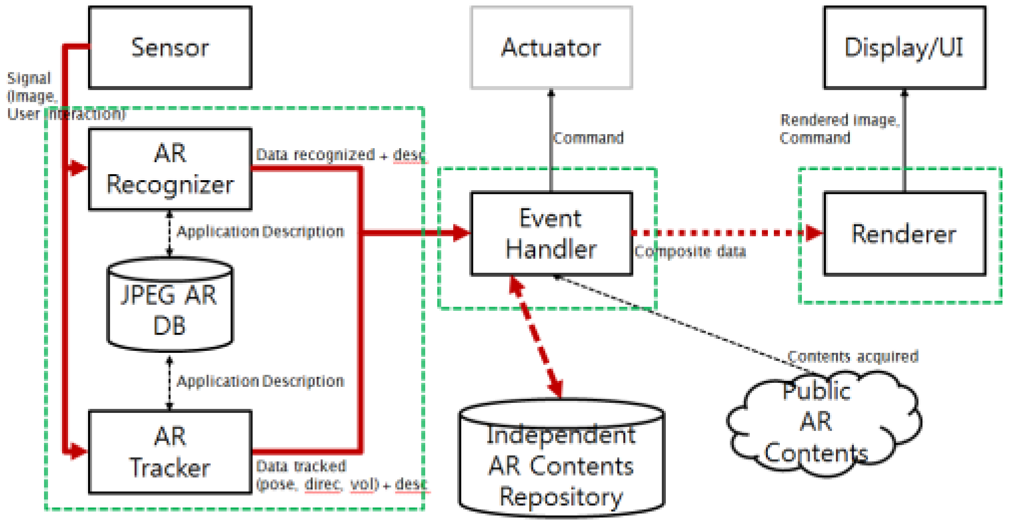


Figure 2-13 JPEG AR Architecture [39]

To construct a JPEG AR application, an image that contains both the real world scene and objects through image registration is required. [40]

Augmented Reality applications are pervasive nowadays caused by dramatically enhanced software and hardware of mobile devices. These applications are opening a new era of immersive experiencing environments to the end user. Augmented Marketing enabled by JPEG Augmented Reality standard is presented. JPEG AR standard provides standard interface, metadata, and file format, not only for the product marketing domain but also for any JPEG related AR domains. We see the fruitful future of AR ecosystem in the JPEG AR compliant environments in which the flexibility of the systems is maximized and current AR market in terms of application point of view is reflected. [39]

2.5.8 JPEG XT

JPEG XT [1] is a new standardization effort which targets the extension of the JPEG features mainly by providing backward and forward compatibility with the JPEG legacy format. It also enables support for High Dynamic Range Imaging, lossless and near-lossless coding, and alpha channel coding. The use of JPEG 2000 and JPEG XR in the market is limited due to lack of backward compatibility to the JPEG legacy. Since the new JPEG extension, JPEG XT is compatible to the 8 bit mode, images of higher bit depth (more than 8 bpp) are used which can also be represented in floating point. The JPEG XT can reduce the file size more than 16 times to store the floating point HDR sample data than lossless OpenEXR while insuring the quality.

JPEG XT will be discussed in detail in Chapter 3.

2.6 Summary

Chapter 2 focuses on the history of JPEG by listing down all the standards developed so far along with their structure, advantages, disadvantages and applications. Chapter 3 concentrates on JPEG XT while describing the need of JPEG XT, its structure, encoding features, the role of JPEG XT in HDR, and advantages and disadvantages.

Chapter 3

Working of JPEG XT

3.1 Why JPEG XT?

The JPEG standard is well known and easily adapted. The extent to which millions of pictures are created, shared and stored, force the JPEG committee to improve and expand its features every time while developing a new standard. With the development in the photography industry and the invention of high-end cameras offering great features such as High Dynamic Range (HDR), JPEG needed to establish a new image processing chain for the photography market.

Digital cameras providing high dynamic range (HDR) images whose color is represented by 12 or 14 bits per color channel are popular. However, most display devices and the de facto international image coding standard, the legacy JPEG (ISO/IEC 10918), still remain in low dynamic range (LDR) images whose color is represented by 8 bits per color channel. [7]

Even though JPEG XR [5] and JPEG 2000 [4] were proposed for the HDR image coding standard, their usage has been limited in the industry mainly due to the lack of backward compatibility with the legacy JPEG. To resolve this problem and modernize the old standard and address the needs of the current market, the JPEG working group (WG) is undertaking a new standardization of backward compatible JPEG image coding for HDR images named JPEG XT. [7] Backwards compatibility here means that legacy decoders are enabled to reconstruct a lossy low dynamic range (LDR), standard color gamut version of the encoded image, while the complete (full range, lossless,...) image data is only available to decoders compliant to the new standard. [41]

JPEG XT is a standardization effort mainly targeting the high dynamic range imaging by providing larger color gamut data. JPEG XT also provides support to JPEG’s features such as lossless and near lossless coding, and alpha channel coding, while also guaranteeing backward and forward compatibility with the JPEG legacy format. [6] Hence, JPEG XT was designed as a multi-part standard, where each subsequent part adds additional features, all integrated into a common framework, allowing addressing individual requirements by selecting the necessary coding features from the parts as necessary. [41] Due to the backward compatibility feature in JPEG XT, all its extensions are derived from the known JPEG algorithms. This led to an easy implementation of the existing hardware: with little more than two readily available JPEG chips, plus pixel-wise post-processing.

It is based on a two-layer design, a base layer which uses the conventional JPEG coding, containing a Low Dynamic Range (LDR) version of HDR image generated by a Tone-Mapping Operator (TMO), and an extension layer providing the full dynamic range.

3.2 Structure of JPEG XT

Out of the multiple coding tools, such as lossless coding, Huffman coding, arithmetic coding or pyramidal coding, defined by JPEG, only the DCT-based Huffman coding is popularly used. It was found that some parts of JPEG are surprisingly not a part of the original standard at all. The YC_bC_r color space was first described in JFIF and was only standardized as ISO/IEC 10918-5 a couple of years ago, and so was the up-sampling procedure for sub-sampled chroma components [41]. What the original JPEG standard does is, it reconstructs the samples to data only, and leaves all the color interpretation to other standards. Thus, the JPEG committee based the new JPEG XT coding technology on the firm grounds to overcome the difference between the "lived JPEG" and the ISO standard in mind.

JPEG XT is the outcome of the first call to a new standard in Paris, 2012. Five proposals were received in the next Shanghai meeting from: EPFL in Switzerland, VUB Brussels, Trellis Management, Dolby and the University of Stuttgart. All the proposals covered different aspects and hence it took almost a year to formulate the new standard with one common architecture.

JPEG XT is structured into nine parts, forming a hierarchy of extensions of JPEG as it is used today. The Dolby proposal was standardized in JPEG XT part 2 with its legacy syntax, or in the generic common syntax of part 3, as profile A of JPEG XT part 7. Profile B of the same part is a slightly extended variant of the Trellis proposal, whereas Stuttgart defined parts 6 and 8, plus profile C of part 7. [41] Figure 3-1 shows the structure of JPEG XT which is described as follows: [6]

- PART 1: It defines the base coding technology (JPEG legacy) and is the core of the JPEG XT Standard.
- PART 2: It describes a backward compatible extension of JPEG for HDR (floating point images).
- PART 3: It embeds the box-based file format into the legacy JPEG syntax.
- PART 4: It defines the mechanisms for reference testing.
- PART 5: It specifies the reference software of the JPEG XT Standard.
- PART 6: It defines the extensions of the JPEG standard for coding the IDR images, i.e. integer samples between 8 and 16 bits precision.

- PART 7: It specifies the coding of HDR images of floating point samples.
- PART 8: It defines the lossless extension of PART 6 and PART 7.
- PART 9: It adds the ability to include an alpha channel along with the image data, to the JPEG legacy.

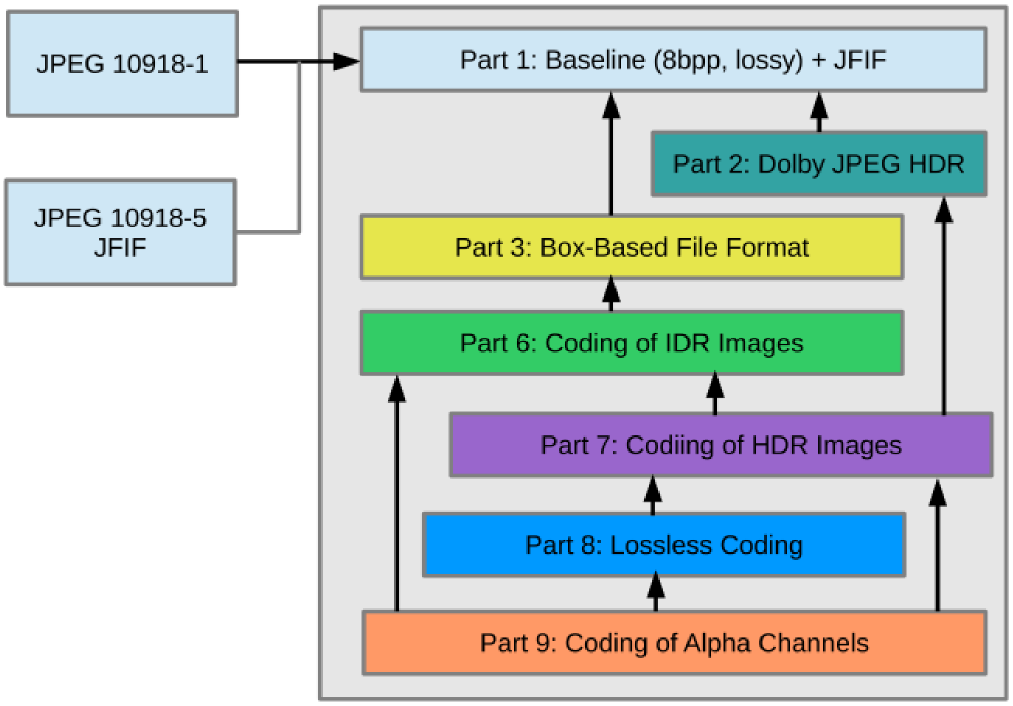

JPEG 10918-1
JPEG 10918-5 JFIF
Part 1: Baseline (8bpp, lossy) + JFIF
Part 2: Dolby JPEG HDR
Part 3: Box-Based File Format
Part 6: Coding of IDR Images
Part 7: Codiing of HDR Images
Part 8: Lossless Coding
Part 9: Coding of Alpha Channels

Figure 3-1 Overview on the parts of JPEG XT and their relation. Parts 4 and 5 define conformance testing and the reference software and are not included in the picture [41]

The JPEG XT Part 1, the core of the JPEG XT standard, describes the 8 bit integer mode as supported by the legacy JPEG decoders.

A backwards compatible signaling mechanism based on the ISO media file format specified in part 3 instructs the decoder how to merge refinement, extension and base layer into one final image; meta data and residual and refinement coded data is here embedded into so-called boxes, a syntax element JPEG XT has in common with JPEG 2000 and MPEG standards, and allowing future extensions of JPEG XT towards applications such as JPIP – interactive image browsing - that depend on such box-structures. The boxes itself are hidden from legacy decoders by encapsulating them in application markers, a generic extension mechanism already defined in the legacy standard. Parts 7 and 8 extend and employ these extension mechanisms of part 6 to enable coding of HDR data and lossless coding. [41]

Part 6 is a combination of two former parts: Part 1 and Part 3. It uses the foundation of part 1 and the extension mechanism of Part 3. This part describes an "intermediate dynamic range (IDR)" image. An IDR image is generally larger than the other traditional JPEG images. Part 6 also supports the encoding of raw data. Part 6 encoding is almost identical to Part 7 profile C encoding, except that it lacks a conversion step from integer to floating-point that is present in the latter [6]. Two orthogonal coding mechanisms to extend the dynamic range, are introduced in JPEG XT part 6:

- Refinement coding: It provides backwards compatibility to the 8-bit mode while it is still quite similar to the legacy 12-bit mode. It works in the DCT domain as it increases the precision of the DCT coefficients.
- Residual coding: This mechanism works in the spatial domain and extends the bit precision of the base layer by including a second, independently coded extension layer. The coding mechanism adopted by the extension layer is closely linked to the legacy JPEG coding modes.

3.3 Architecture of the JPEG XT Standard

The JPEG XT follows a two-layer coding structure:

- Base layer, a legacy JPEG coded LDR
- Enhancement layer, the residual to produce the HDR (This layer uses maximum JPEG legacy coding tools)

The joint decoder architecture (detailed and simplified versions) of JPEG XT is depicted in Figures 3-2 and 3-3 respectively.

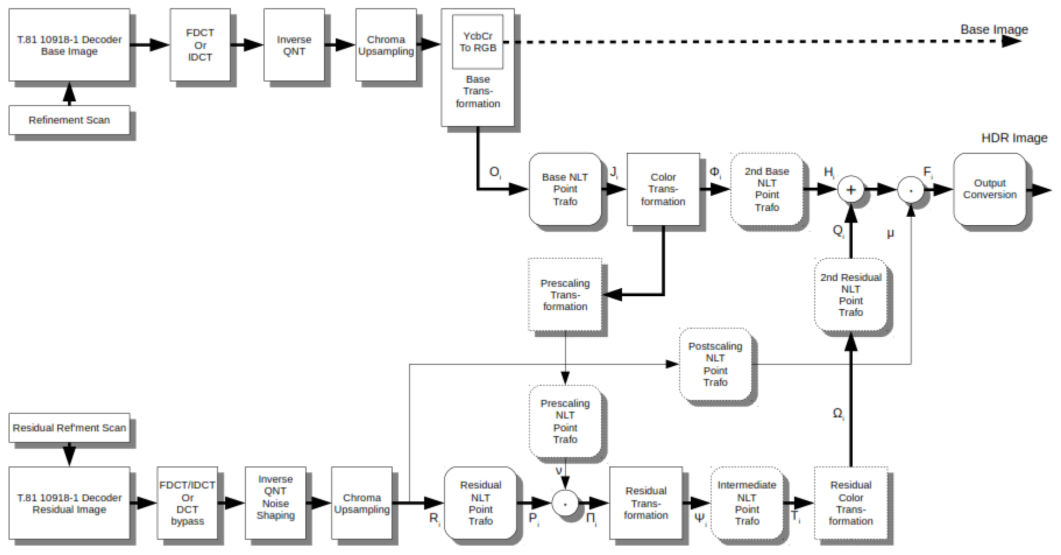


Figure 3-2 Decoder Design of JPEG XT, merging the functionality of all parts. Round boxes are linear matrix transformations, regular boxes apply separately to each component. Thick lines transport three channels, thin lines a single channel. The dotted boxes are not required for part 6, i.e. intermediate dynamic range coding [41]

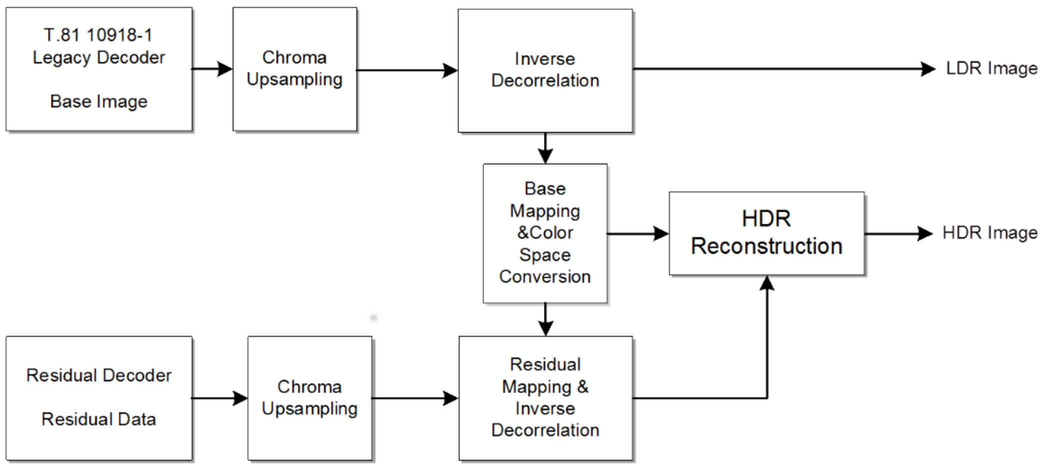


Figure 3-3 JPEG XT Simplified Decoder Block Diagram [42]

By comparing the two versions of the JPEG XT decoder design, a couple of observations are made: First, the legacy JPEG standard only covers the first three left-top boxes of Figure 3-2, denoted as “T.81 10918-1 Decoder”, “FDCT or IDCT” and “Inverse Quantization”. JPEG adds the two additional boxes to the right: Chroma up-sampling and transformation from YC_bC_r to RGB. The lower left box, labeled “Residual Image” represents the decoder for the extension layer, and everything to the right of this box and below the five boxes in the top row implement the merging operations that compute a HDR or IDR image from base layer (top row) and extension layer (bottom row). These operations are all carried out in the spatial domain, on a pixel-by-pixel basis. The only extension that has been made in the DCT domain is Refinement Coding as discussed before for part 6, indicated by the small boxes that point into the base image and residual image decoder. [41] The refinement coding is comparable to the JPEG 2000 Refinement Coding Pass, which also adds least significant bits, but unlike its JPEG 2000 counterpart, it is based on the successive approximation scan pattern of the progressive mode of JPEG. [6]

All boxes on the right of the chroma upsampling steps in Figure 3-2 merge the base image and residual image together to form an IDR or HDR output image. While the depicted set of operations in this figure is complete, each part or each profile of a part uses only a subset of all operations. The full decoder is constructed from four elementary operations performed on a pixel-by-pixel basis: [6]

- Scalar non-linear transformations perform either a table look-up using the input value as index into the table, or scale the input to the range [0; 1], apply a selection of parametrized functions such as linear ramps or gamma corrections, and then scale the output of such functions to the output range. In Figure 3-2, such scalar non-linear transformations are depicted by boxes with rounded edges.
- The second type of transformation is multiplication by 3 x 3 matrices that operate again on a pixel-by-pixel basis, but across components or channels. These transformations either perform inverse component decorrelation transformations such as the transformation from YC_bC_r to RGB, or change the colorspace from one to another. They are depicted in Figure 3-2 by square boxes.
- A vector addition indicated by a circled plus-sign, combining base and residual images.
- A scalar multiplication that optionally scales all components by a common factor, represented by a circled dot.

All parts and profiles of JPEG XT are constructed from these four operations, and they are always applied in the order given in Figure 3-2. A bypassed non-linear transformation will just scale the bit-range of the input to the bit-range of the output, a bypassed linear transformation performs no operation, i.e. uses the identity matrix. Hence, all parts of

JPEG XT can be implemented by one common code, replacing those boxes that are not present by their corresponding default operations. [6]

The JPEG XT standard currently does not allow deviation from the order of operations, i.e. the data flow from the decoded and inversely DCT transformed data to the output is always identical, and always as in Figure 3-2, for all parts and all profiles. It may become necessary for future applications to include additional elements in the common decoder architecture, and hence to include a description of the decoder chain within the JPEG XT chain. [6]

3.4 Encoding

The JPEG standard already defines a mechanism for including extension information by so-called application markers. Sixteen of such markers (APP_1-APP_{16}) are available, and each marker segment can carry up to 64K of payload data. JPEG XT employs application marker 11 (APP_{11}) to both carry the entropy coded data of refinement and residual scans, and the meta-data that defines all the parameters for merging base and residual images together. [6] The JPEG Standard includes extension information by making use of application markers. Figure 3-4 shows how APP_{11} is used in the JPEG XT file and Figure 3-5 describes various formats for some of the application markers.

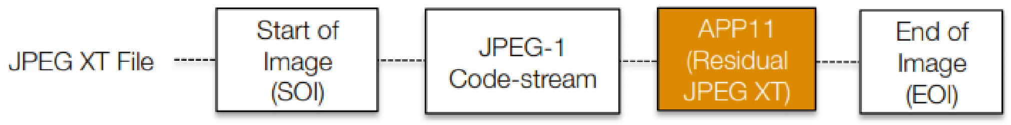
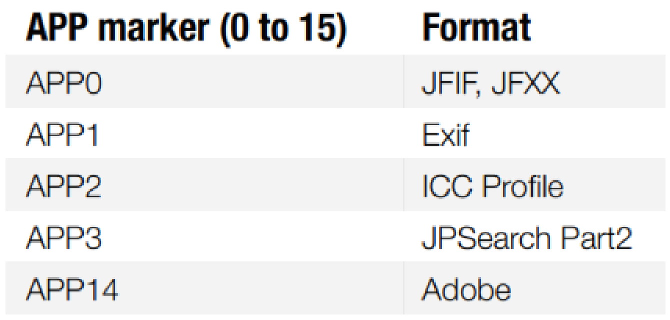


Figure 3-4 Use of Application Marker in JPEG XT [10]



APP marker (0 to 15)	Format
APP0	JFIF, JFXX
APP1	Exif
APP2	ICC Profile
APP3	JPSearch Part2
APP14	Adobe

Figure 3-5 Listings of different Application Markers [10]

The ISO-based media format, also applied for example in the JPEG 2000 file format, is structured around a common syntax element called a box. Boxes have both a length and a type field such that decoders that are unaware of a particular box type and its purpose can skip over them. Boxes contain either structured data, with the structure defined by the standard, entropy coded data, or other boxes. The latter boxes are also referred to as super boxes. [6]

JPEG XT has to use a different transport mechanism for legacy reasons. Within JPEG XT, the legacy codestream carries application markers which again carry the box information. Due to the size constraint of application markers, a single box may, however, extend over several application markers of the same type. The box syntax has been extended by indicators that tell the decoder how to assemble one or several application markers back together into one single box. [6]

3.5 HDR

High Dynamic Range (HDR) [2] Imaging has been found useful in graphics and photography. The reason behind the growing popularity of such images is that they represent a wider range of luminance values which are closer to the luminance range of the Human Visual System (HVS). HDR images are better with high and low illuminated regions as compared to the conventional Low Dynamic Range (LDR) images, thus making HDR images more suitable for capturing richer information from scenes. HDR images are displayed on legacy monitors using Tone-Mapping Operators (TMOs) which map the HDR wider range of contrasts and colors to the ranges available in the displays. Other than JPEG 2000 and JPEG XR, no other image or video coding standards are providing support for HDR content. This lack of support for HDR by the omnipresent JPEG legacy standard can become an issue in its further adoption.

3.6 Role of JPEG XT in HDR

JPEG XT is the standard under development that targets specifically the compression of HDR pictures. Enabling coding of HDR content by a straightforward increase in bit depth support to 16-bit integer (IDR) or floating-point (HDR) would break the backward compatibility with the JPEG legacy implementations. Hence, the JPEG XT standard enables HDR support by splitting the image data in a Low Dynamic Range (LDR) JPEG legacy compliant codestream and a residual stream, allowing to scale-up to an HDR decoded image which is smartly embedded in the JPEG legacy codestream such that it does not jeopardize backward or forward compatibility. Mapping the HDR image onto an LDR image is performed by a tone-mapping algorithm. [43]

3.6.1 Tone- Mapping Operator

The tone mapping operator (TMO) is used to display the HDR image or video on the conventional LDR display device. [7] In order to support backward compatibility with the legacy JPEG, JPEG XT codestream delivers a tone-mapped LDR image of the original HDR image as a base layer. In JPEG XT, the user has the option to choose the TMO. The measure behind tone mapping is shown in the Figure 3-6.

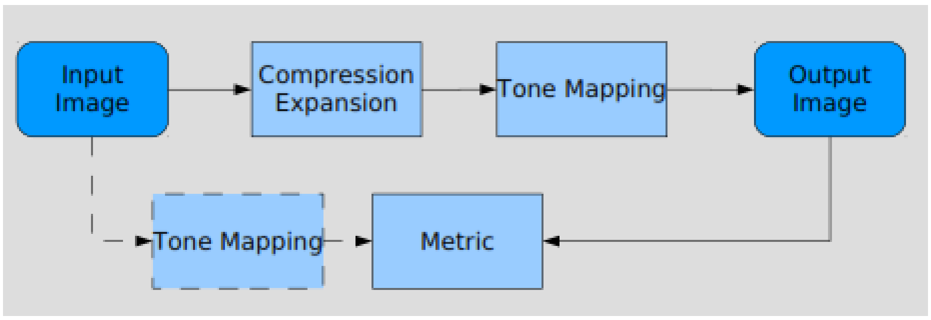


Figure 3-6 Measure behind Tone mapping [44]

3.6.2 Quality Metric

Quality metrics can be classified into two categories: Objective and Subjective quality metrics which evaluate the image coding performance. The objective quality metric includes Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) while Structural quality metric includes Structural Similarity (SSIM) index only.

3.6.3 Encoding Parameter

JPEG XT encoding depends mainly on two parameters:

- Parameter q: controls the base layer coding quality index, which is exactly the same parameter that is used in the conventional JPEG compression tools
- Parameter Q: controls the quality of the residual HDR information.

Part 7 of JPEG XT is used for coding of HDR images. Three profiles have been proposed for JPEG XT part 7 up until now. Since JPEG XT is designed as a backward compatible extension to the JPEG legacy standard, all the additional data has to be encoded non-intrusively for legacy implementations.

3.7 Profiles of JPEG XT Part 7

JPEG XT Part 7 consists of three profiles as shown in the Figures 3-7, 3-8 and 3-9. JPEG XT file contains codestreams of the base and residual layers. The application marker distinguishes them in the legacy JPEG decoder. White blocks are the common functional blocks for three profiles, and grey blocks illustrate the differences among the three profiles. Roughly, JPEG XT uses the Weber-Fechner law [45]. The intensity of a sensation is proportional to the logarithm of the intensity of the stimulus causing it. An HDR image is represented as a sum of base and residual images in the logarithmic domain or a product in the linear domain.

3.7.1 Profile A

In Profile A, the HDR image is first converted from RGB to YC_bC_r color space after which the residual data is obtained by performing operations such as division and subtraction in the YC_bC_r space. The Y component of the residual image is the ratio of the HDR to LDR and the C_bC_r components are color residuals. [7] The decoding algorithm of profile A of part 7 is identical to that of part 2, except that the syntax representing meta-data is different. The guideline of this algorithm proposes that the LDR image is typically encoded in a gamma-corrected space, i.e. sample values are not proportional to radiance, but proportional to a power function of the radiance.

The decoding algorithm of profile A of part 7 is identical to that of part 2, except that the syntax representing meta-data is different. The HDR image is here reconstructed as

$$\mathrm{HDR} = \mu\left(\Phi(\mathrm{LDR}) + \chi\right)$$

where HDR is the reconstructed HDR image, LDR is the base image in the legacy codestream, Φ is an inverse gamma-correction, χ a color-residual and μ a position-dependent scale factor. The guideline of this algorithm proposes that the LDR image is typically encoded in a gamma-corrected space, i.e. sample values are not proportional to radiance, but proportional to a power function of the radiance. [6]

In the common decoder design, Φ is represented by the Base NLT point trafo, and μ is computed through an exponential ramp from the luminance of the residual image RES_0, i.e.

$$\mu = \exp\left(a\text{RES}_0 + b\right)$$

This exponential function is represented by the Postscaling NLT point trafo in Figure 3-2. [6]

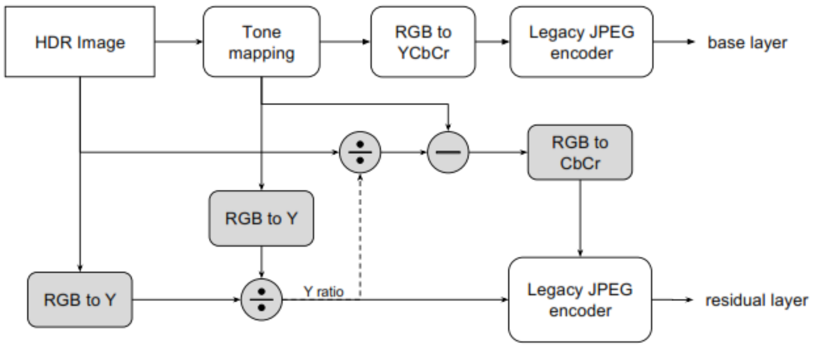


Figure 3-7 Simplified Encoding process of JPEG XT Profile A [7]

3.7.2 Profile C

At this point, it is more convenient to discuss profile C first and proceed to profile B later: While profile A scales all components by one single factor μ, profile C can be understood as a component wise scaling. That is, [6]

$$\mathrm{HDR}_i = \Phi\left(\mathrm{LDR}\right)_i \mathrm{RES}_i \quad (i = 0, 1, 2)$$

Profile C however expresses this relation in the logarithmic space: [6]

$$\mathrm{HDR}_i = \exp\left(\log\left(\Phi\left(\mathrm{LDR}_i\right)\right) + \left(\log\left(\mathrm{RES}_i\right)\right)\right)$$

Thus, it is possible to merge log *Φ* into a single non-linear function that is represented as the Base NLT Transformation in the common decoder architecture. This gives the following simple reconstruction algorithm: [6]

$$\mathrm{HDR}_i = \psi\exp\left(\hat{\Phi}\left(\mathrm{LDR}_i\right) + \mathrm{RES}_i - o\right)$$

Here, $\psi\exp$ is the pseudo-exponential, a piece-wise linear approximation of the exponential function which is represented by the Output Conversion in Figure 3-2, and O is an offset that ensures that the residual image is non-negative. Post or pre-scaling non-linearities are not required in this profile. [6]

Thus, it is observed that Profile C represents the HDR image as a sum of the predicted HDR image and the final residual errors. As shown in Figure 3-8, the refinement scan is performed to increase the bit precision up to 12 bits in the DCT domain. [7]

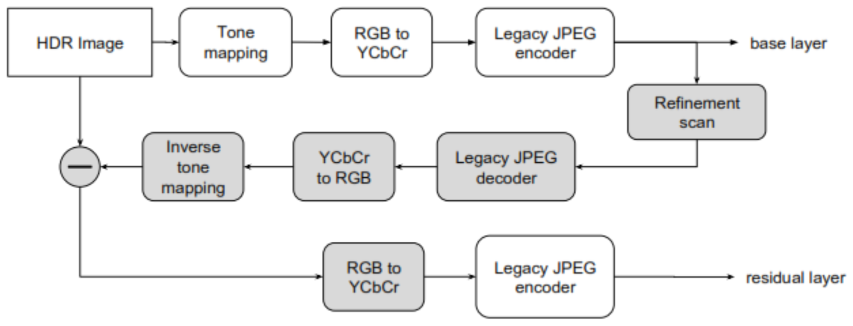


Figure 3-8 Simplified Encoding process of JPEG XT Profile C [7]

3.7.3 Profile B

Profile B provides a tool that allows the use of HDR imaging in a simpler way. The base layer is either an exposed and clamped or tone-mapped version of the original HDR image. The HDR residual layer contains the fractional part of the tone-mapped LDR image divided by the original HDR image for each RGB channel. [7]

Profile B operates similar to Profile C except that it expresses the HDR image as a component-wise quotient instead of a component-wise product. Similar to the working of profile C, the standard operates in a logarithmic representation: [6]

$$\mathrm{HDR}_i = \sigma \frac{\Phi\left(\mathrm{LDR}_i\right)}{\Psi\left(\mathrm{RES}_i\right) + \epsilon} \quad (i = 0, 1, 2)$$

Similar to the working of profile C, the standard operates in a logarithmic representation, using the exact logarithm and not a piece-wise linear approximation: [6]

$$\mathrm{HDR}_i = \sigma \exp\left(\log\left(\Phi\left(\mathrm{HDR}_i\right)\right) - \log\left(\Psi\left(\mathrm{Res}_i\right)\right)\right)$$

The Φ function is, as in profile A, an inverse gamma correction and represented by the Base NLT Transformation. The outmost exponential function together with the scale σ is defined by the Output Conversion, and the logarithms within the exponential by the Secondary Base and Secondary Residual NLT Trafo. Finally, Ψ is the Intermediate Residual NLT. It is typically selected to be a gamma correction whose exponent is image dependent and derived by the encoder. [6]

Similar to profile C, profile B does not require the pre- and post-scaling maps, but it is the only profile that depends on the Secondary Base and Secondary Residual NLT Trafo. A practical decoder implementation would, of course, never go through the exponential and logarithmic functions but would evaluate the quotient directly. [6] Figure 3-9 shows the simplified encoding process of JPEG XT Profile B.

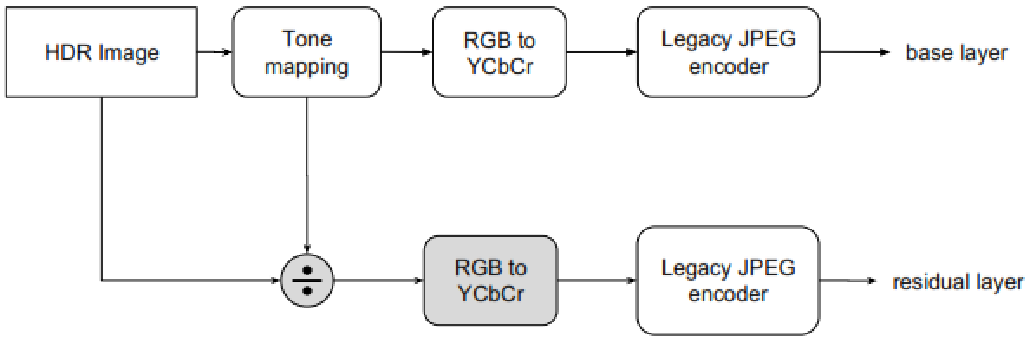


Figure 3-9 Simplified Encoding process of JPEG XT Profile B [7]

Figure 3-10 shows how each profile work in the JPEG XT decoder by assigning a color to each profile individually.

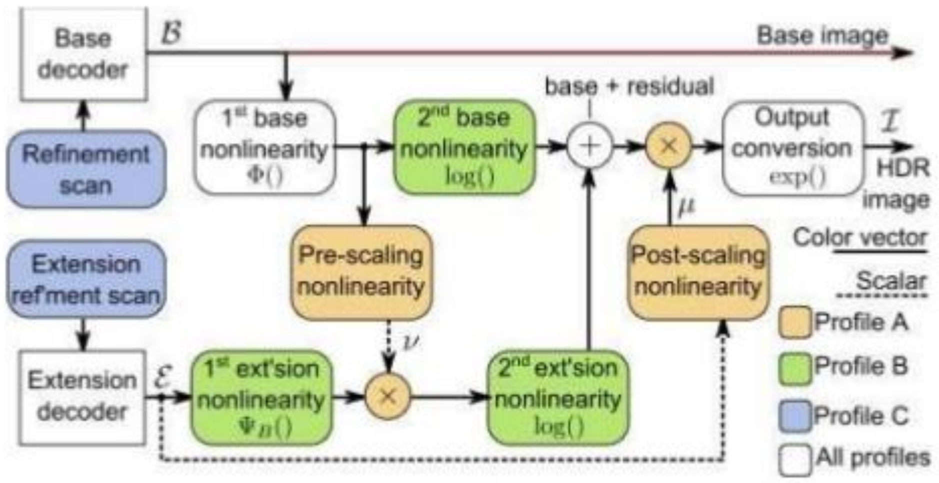


Figure 3-10 JPEG XT Profiles [66]

3.8 Advantages and Disadvantages

JPEG XT is a very flexible and rich image compression framework, allowing lossy and lossless compression of IDR and HDR data, including coding of opacity information. JPEG XT addresses the needs of a “modernized JPEG", filling the gaps the legacy JPEG standard left: Scalable lossy to lossless compression, support for bit-depths between 8 and 16 bits, support for floating point samples and coding of opacity data. Unlike other standardization initiatives, JPEG XT is fully backwards compatible to legacy applications and always includes a lossy 8-bit version of the image in its codestream, visible to legacy applications. Clearly, carrying the legacy of the over twenty-year-old JPEG coding scheme costs performance, and JPEG XT cannot match with more modern compression algorithms such as JPEG 2000 or JPEG XR. However, JPEG XT can be easily implemented on the basis of two readily available JPEG IP cores and a single DSP for post-processing. Unlike more modern architectures, JPEG XT keeps the legacy JPEG tool-chain working, allowing existing software and hardware architectures to access JPEG XT images without the compatibility problems of other image compression formats. [41]

3.9 Summary

In chapter 3, the new JPEG XT standard is introduced. It focuses on the importance of JPEG XT along with explaining the architecture and the encoding profiles. Chapter 4 outlines the need for security and privacy. It also states the proposed algorithm and the software used.

Chapter 4

Implementation

4.1 Need for security

Picture-related applications are extremely popular because pictures present attractive and vivid information. Nowadays, people record everyday life, communicate with one another, and enjoy entertainment using various interesting imaging applications. In many cases, processed images need to be recovered to their original versions. However, most approaches require storage or transmission of both original and processed images separately, which result in increased bandwidth and storage resources to be used. [46]

With the popularization of high-quality digital cameras, smart mobile devices with high-resolution cameras, as well as user-friendly imaging and social networking applications, taking pictures, then editing and sharing, have become part of everyday life for many. Popular imaging applications include Photoshop (professional graphics editor), Picasa (integrated image organizer), Instagram (photo sharing), Snapchat (photo messaging), and tons of image-related games. These photo sharing service providers (PSPs) now have a large user base, to the point where PSP photo storage subsystems have motivated interesting research. However, this development has generated privacy concerns. [46][47]

JPEG also offers a solution to tag images. JPEG/Exif (Exchangeable image file format) is a popular way for digital cameras and other photographic capture devices to tag and capture location related metadata about photos. JPEG/JFIF (JPEG File Interchange Format) is the most popular format for storage and transmission of images on the World Wide Web (www). The two formats are often not distinguished from each other and are simply referred to as JPEG each with their own application segments (APP0 for JFIF, APP1 for Exif) in the header of a JPEG file. Recently, a new standardization activity called JPEG XT has been initiated, addressing the needs of photographers for higher dynamic range (HDR) images in both lossy and lossless coding while retaining backward compatibility to the established legacy JPEG decoders. The central idea underlying the backward compatible coding of HDR content is to encode a low dynamic range version of the HDR image generated by a tone-mapping operator using a conventional JPEG encoder and to insert the extra encoded information for HDR in an application marker. Adopting this idea, any useful information can be embedded in the application markers of a JPEG file. [46]

Private photos have been leaked from a prominent photo sharing site [48]. Furthermore, widespread concerns have been raised about the application of face recognition technologies in Facebook [49][50]. Despite these privacy threats, it is not clear that the usage of photo sharing services will diminish in the near future. This is because photo sharing services provide several useful functions that, together, make for a seamless photo browsing experience. In addition to providing photo storage, PSPs also perform several server-side image transformations (like cropping, resizing and color space conversions) designed to improve user perceived latency of photo downloads and, incidentally, bandwidth usage (an important consideration when browsing photos on a mobile device). [47]But for the sake of avoiding further privacy issues, it is always recommended to embed security in the JPEG standards.

Figure 4-1 Top companies with privacy issues [58]

The Figure 4-2 depicts the process of including the privacy in JPEG XT application marker 11 (App11). The different protection methods that could be used are: cipher, digital signature, hash function, encryption tool registration authority (like JPSEC-RA). The protected image data contains partial DCT coefficients to overwrite the non-protected data (minimum region= 8 x 8 DCT block). [10]

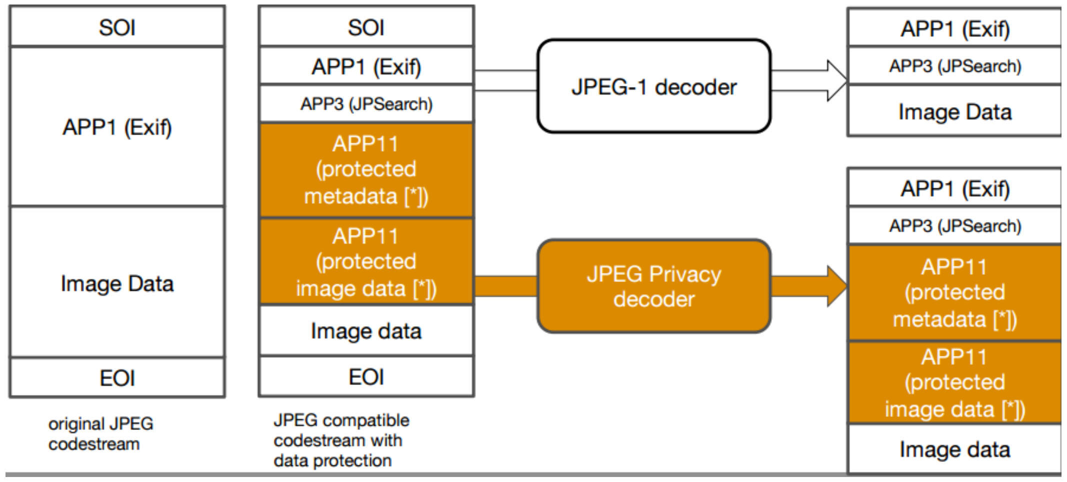


Figure 4-2 JPEG Privacy App11 [10]

4.2 Proposed Algorithm

4.2.1 Basic Algorithm

With the growing demand for identity protection, it is necessary to obtain a way, to secure images uploaded on the internet especially on the social networking websites. To provide one possible solution to this privacy problem, an algorithm that will help the users to protect their pictures is proposed using OpenCV[67] and C++.

The algorithm is split into two parts:

I. Face Detection

Face detection is done by loading the Haar cascade classifier from the OpenCV libraries. This classifier helps in detecting the number of faces in the input image by using the in-built detectMultiScale() function of the cascade classifier. Using additional functions of OpenCV, each face is embossed by a shape (rectangle (as done in this algorithm), ellipse, etc.) and is saved.

II. Gaussian Blur

To enable privacy, blurring is incorporated by applying the in-built GaussianBlur() function from the imgproc (image processing) module of OpenCV. The masking kernel is taken from the user and passed in this function. The size of the kernel is proportional to the amount of blur. The output image consisting of the blurred faces is then saved.

This algorithm works for color images as well as grayscale images.

4.2.2 Haar Cascade Classifier 0 [51][52]

The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection. This algorithm is implemented in OpenCV as cvHaarDetectObjects(). [51] This classifier is namely a cascade of boosted classifiers working like Haar-like features.

A classifier, in general, consists of hundreds of training data. This data helps in detecting objects. The positive examples, i.e. the object to be detected are scaled to the same size whereas the negative examples, i.e. the remaining arbitrary image is scaled to the same size but different than the positive example size. Once the classifier is trained, it is applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs a “1” if the region is likely to show the object of interest (i.e. face), or “0” otherwise. While searching for the desired object in the whole image, the search window is moved across the image and every location is checked, using the classifier. The classifier is designed such that it can be easily “resized” in order to find the objects of interest at different sizes, which is more efficient than resizing the image itself. Hence, in order to find an object of an unknown size in the image, the scan procedure is done several times at different scales. [52]

The word “cascade” in the classifier name means that the resultant classifier consists of several simpler classifiers (*stages*) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. Currently Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost are supported. The basic classifiers are decision-tree classifiers with at least 2 leaves. Haar-like features are the input to the basic classifiers, and are calculated as described below. The current algorithm uses the Haar-like features which are shown in Figure 4-3. [52]

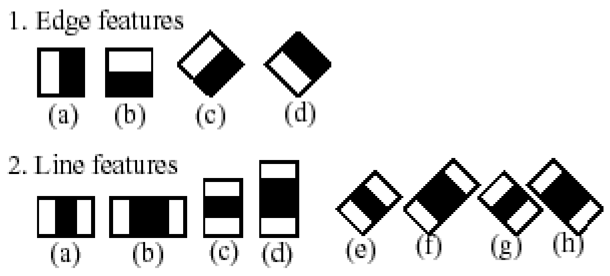


Figure 4-3 Haar-like features [52]

4.2.3 Gaussian Blur

A Gaussian blur (also known as Gaussian smoothing) is developed by using a Gaussian function to blur an image. It helps to reduce the image noise and details. Gaussian filtering is done by convolving each point in the input array with a *Gaussian kernel* and then summing them all to produce the output array. A 1D Gaussian kernel is shown in Figure 4-4 [53]

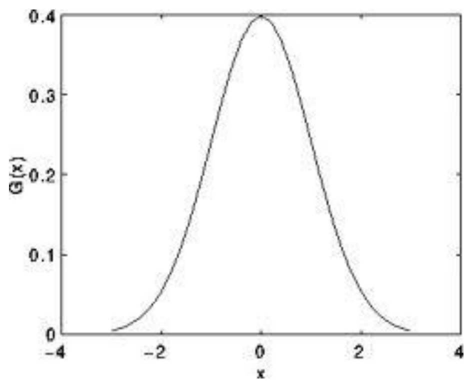


Figure 4-4 Gaussian kernel [53]

Assuming that an image is 1D, you can notice that the pixel located in the middle would have the biggest weight. The weight of its neighbors decreases as the spatial distance between them and the center pixel increases. [53]

4.3 Software

The analysis of the JPEG XT software is explained as follows:

1. Download the JPEG XT software from the website link provided in [56].
2. Run the solution code in Visual studio [57].
3. The jpeg.exe file will be saved in the project folder named debug under jpeg.
4. Open the command prompt from the debug folder. Run jpeg.exe to obtain the syntax and different options for using the software.
5. For choosing the test image, either save the image in the debug folder or give the entire path while running the software. It is advisable to save the image in the same folder location as the jpeg.exe file. The test image must be a jpeg file.
6. The syntax for encoding is: jpeg.exe [options] infile.ppm outfile.jpg
   This is the standard JPEG compression with 4:4:4 subsampling. This software only accepts the image in .ppm file format for encoding.
7. The syntax for decoding is: jpeg.exe infile.jpg out.ppm
8. To work on a jpeg image, decode the jpeg image into a .ppm file first and then encode it with suitable parameters.
9. The results of the encoding and decoding are saved in the same debug folder.

4.4 Experimental Setup

4.4.1 System

CPU: Intel® Core™ i7-4510U CPU @ 2.00GHz

4.4.2 Software

JPEG XT Reference Software

4.4.3 Library

OpenCV Version 2.11.4

4.4.4 Tools

Microsoft Visual Studio 2013

MATLAB R2015a

4.4.5 Test Images

4.4.5.1 Microsoft Visual Studio 2013

1. lena.jpg [59] (512x512)
2. friends.jpg [60] (1300x866)
3. steripack.jpg [61] (962x369)
4. unicefUSA.jpg [62] (1024x768)
5. uta.jpg [63] (1073x618)

4.4.5.2 MATLAB R2015a

1. hdr.jpg [64] (3264x2248)
2. *newyork_hdr.jpg [65] (1920x1080)*

4.5 Application

For a social media application such as Facebook which is the biggest platform worldwide, has been facing some serious privacy issues. Even though their accounts are private, the profile picture is still visible for every account irrespective of being friends or not. Due to this, users have been facing identity issues. By using the non-encrypted profile pictures, fake accounts are created with a different name, thus disrupting the original user's privacy. Hence it is vital for such applications to provide full privacy support to the users. One way of doing this can be by enabling the algorithm, i.e. blurring the faces by default for those who are not friends. The ‘friend request’ option can be the key to deblurring the image. In this way, the group photos can also be protected by blurring all the faces in the picture and every face can only be deblurred if they are friends.

4.6 Summary

Chapter 4 shows how security and privacy can be a part of the JPEG XT standard. It also explains the proposed algorithm in detail. Chapter 5 tracks the results of the proposed algorithm for five different test sequences while describing the steps in detail.

Chapter 5

Results

Due to limited resources, time constraints and compatibility issues, the proposed algorithm could not fit well in the JPEG XT software. Hence, the test images were first passed through the JPEG XT software to encode them and then the encoded images were encrypted using the proposed algorithm.

The performance of this algorithm is observed visually. Five different test sequences [59-63] were first decoded to the .ppm format and then encoded using the JPEG XT software [56]. The compressed output file is then passed through the proposed algorithm for face detection and blurring, thus providing privacy.

5.1 Execution steps

STEP 1: Command line argument is passed through the command prompt for decoding the infile.jpg image to out.ppm. The output file is saved at a user chosen destination.

STEP 2: Another command line argument is passed through the command prompt but with additional options for encoding. In this thesis, the quality parameter '-q' is passed for different values.

Step 3: The output.jpg file obtained in Step 2 is then called in the proposed algorithm to perform additional functions such as face detection and blurring.

The above steps show the functionality of the JPEG XT software and the proposed algorithm separately.

5.2 Test Sequences and Output

5.2.1 lena.jpg

STEP 1

The original lena.jpg image as shown in Figure 5-1 is passed through the JPEG XT software and decoded to lena.ppm.

Command line argument: jpeg.exe lena.jpg lena.ppm

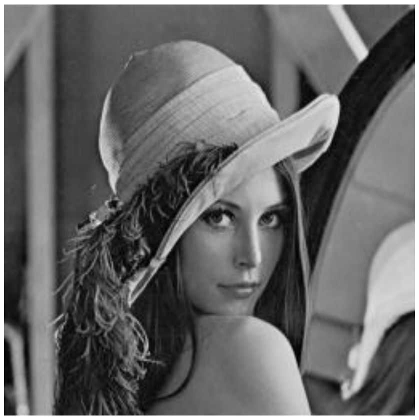


Figure 5-1 Original Image (lena.jpg)

STEP 2

The decoded image is then encoded by setting the base layer quality factor to 25.

Figure 5-2 shows the encoded lena_out.jpg image.

Command line argument: jpeg.exe –q 25 lena.ppm lena_out.jpg

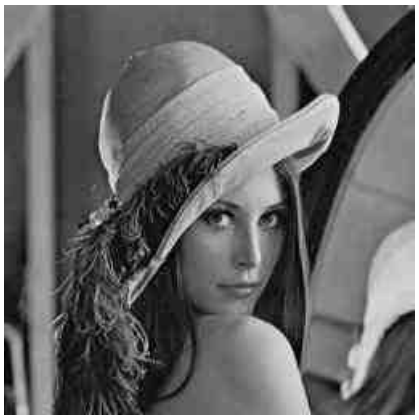


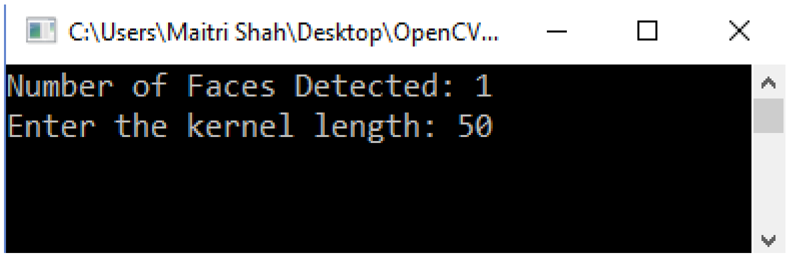
Figure 5-2 Compressed Image (lena_out.jpg ‘q=25’)

STEP 3

For encrypting the image, the encoded lena_out.jpg image is passed through the proposed algorithm to obtain lena_protect.jpg.

Figure 5-3 shows the command prompt which displays the number of detected faces and asks the user to enter the kernel length.

Figure 5-4 shows the encrypted image and the original cropped face region.



```
Number of Faces Detected: 1
Enter the kernel length: 50
```

Figure 5-3 Command Prompt Input and Output Arguments

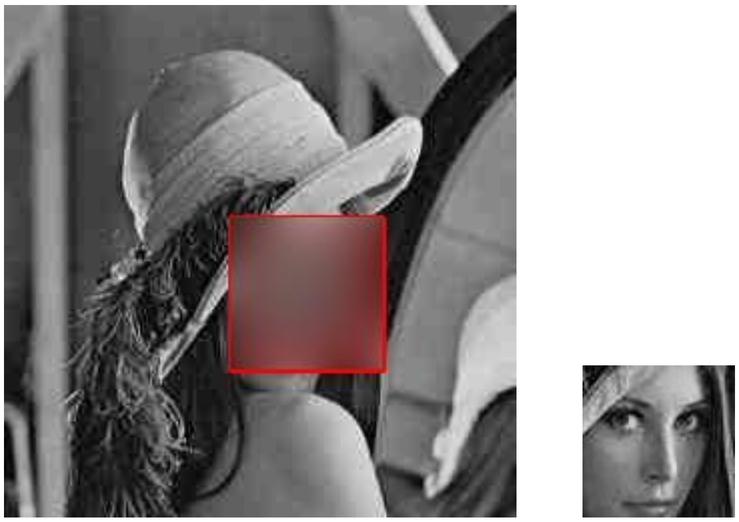


Figure 5-4 Protected Image (lena_protect.jpg) and the cropped face

5.2.2 friends.jpg

STEP1

The original friends.jpg image as shown in Figure 5-5 is passed through the JPEG XT software and decoded to friends.ppm.

Command line argument: jpeg.exe friends.jpg friends.ppm

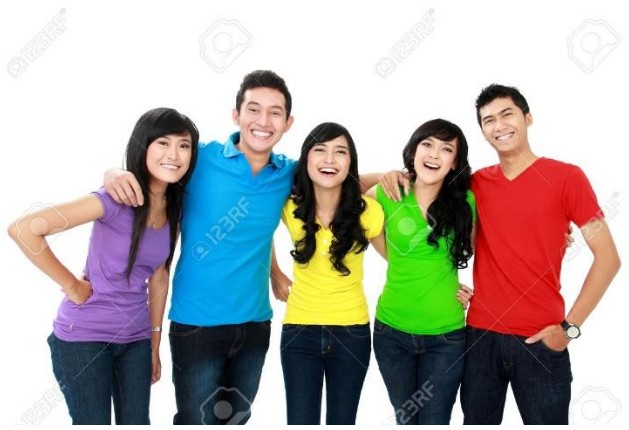


Figure 5-5 Original Image (friends.jpg)

STEP 2

The decoded image is then encoded by setting the base layer quality factor to 35.

Figure 5-6 shows the encoded friends_out.jpg image.

Command line argument: jpeg.exe –q 35 friends.ppm friends_out.jpg

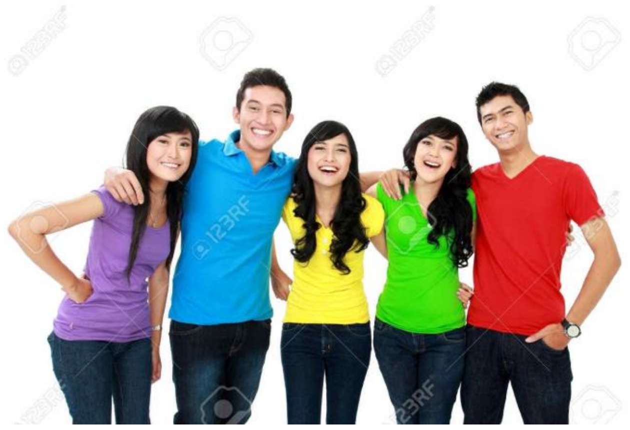


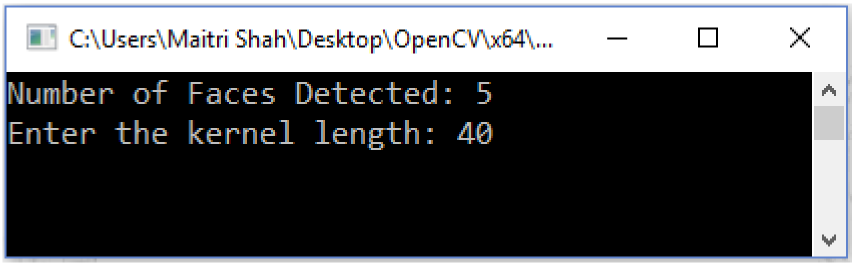
Figure 5-6 Compressed Image (friends_out.jpg ‘q=35’)

STEP 3

For encrypting the image, the encoded friends_out.jpg image is passed through the proposed algorithm to obtain friends_protect.jpg.

Figure 5-7 shows the command prompt which displays the number of detected faces and asks the user to enter the kernel length.

Figure 5-8 shows the encrypted image and all the cropped original faces.



```
Number of Faces Detected: 5
Enter the kernel length: 40
```

Figure 5-7 Command Prompt Input and Output Arguments

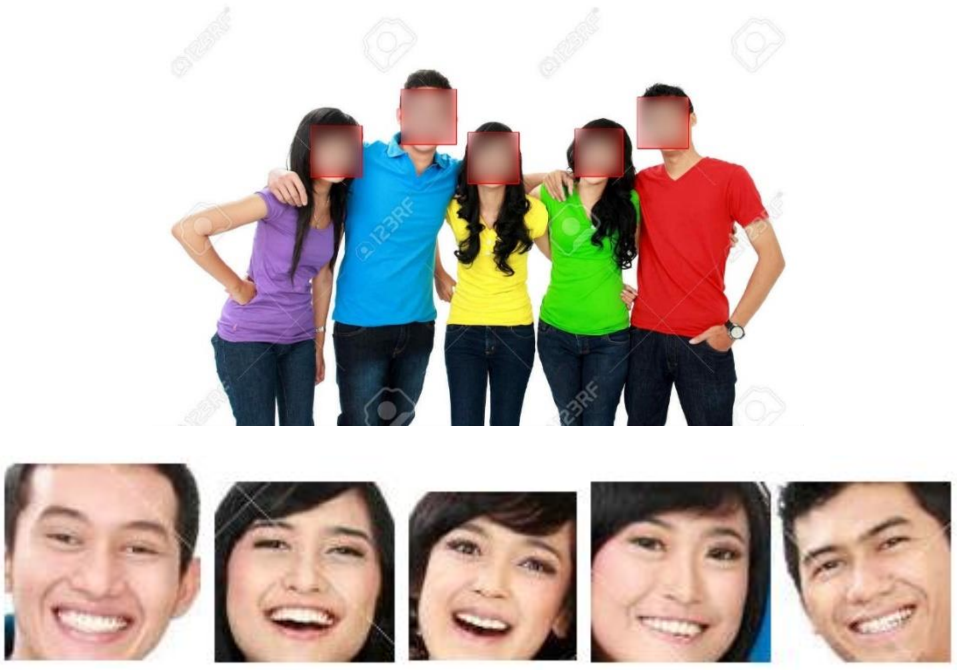


Figure 5-8 Protected Image (friends_protect.jpg) and the cropped faces

5.2.3 steripack.jpg

STEP 1

The original steripack.jpg image as shown in Figure 5-9 is passed through the JPEG XT software and decoded to steripack.ppm.

Command line argument: jpeg.exe steripack.jpg steripack.ppm

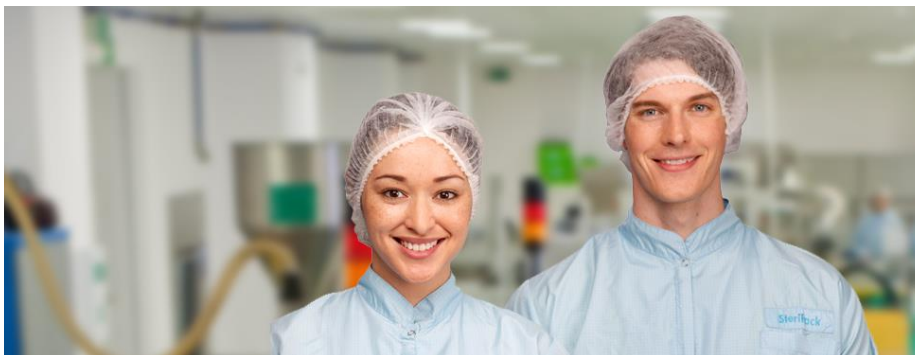


Figure 5-9 Original Image (steripack.jpg)

STEP 2

The decoded image is then encoded by setting the base layer quality factor to 60.

Figure 5-10 shows the encoded steripack_out.jpg image.

Command line argument: jpeg.exe –q 60 steripack.ppm steripack_out.jpg

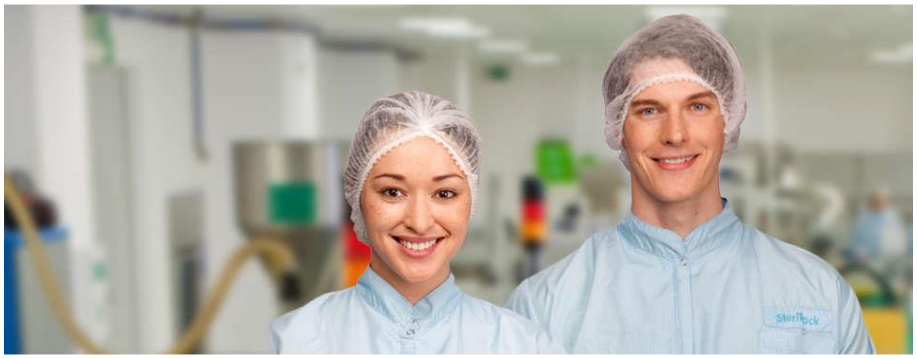


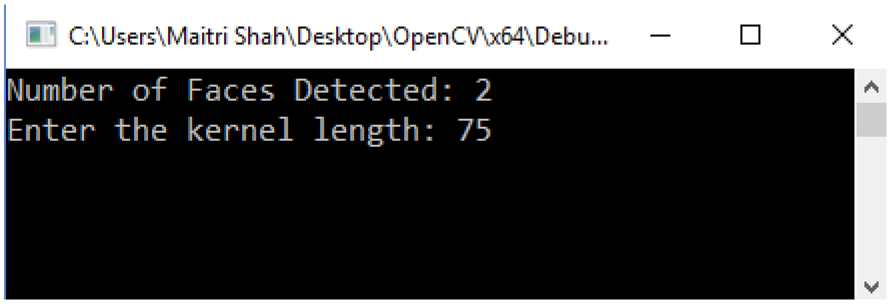
Figure 5-10 Compressed Image (steripack_out.jpg ‘q=60’)

STEP 3

For encrypting the image, the encoded steripack_out.jpg image is passed through the proposed algorithm to obtain steripack_protect.jpg.

Figure 5-11 shows the command prompt which displays the number of detected faces and asks the user to enter the kernel length.

Figure 5-12 shows the encrypted image and all the cropped original faces.



```
Number of Faces Detected: 2
Enter the kernel length: 75
```

Figure 5-11 Command Prompt Input and Output Arguments

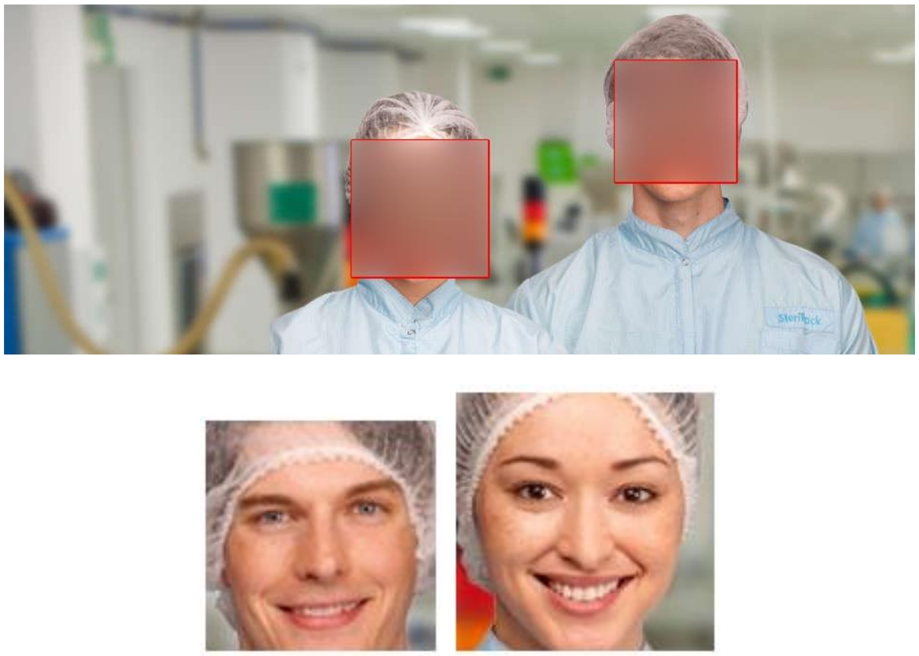


Figure 5-12 Protected Image (steripack_protect.jpg) and the cropped faces

5.2.4 unicefUSA.jpg

STEP 1

The original unicefUSA.jpg image as shown in Figure 5-13 is passed through the JPEG XT software and decoded to unicefUSA.ppm.

Command line argument: jpeg.exe unicefUSA.jpg unicefUSA.ppm

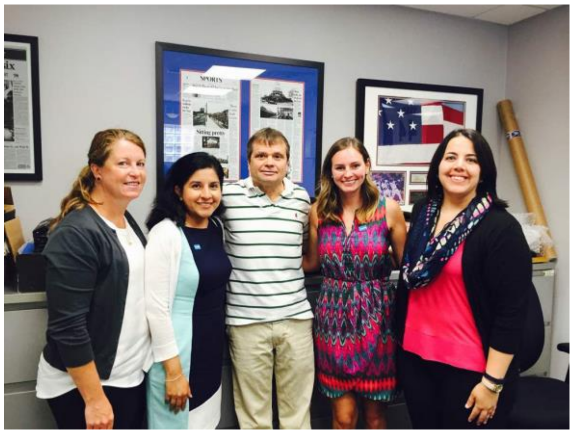


Figure 5-13 Original Image (unicefUSA.jpg)

STEP 2

The decoded image is then encoded by setting the base layer quality factor to 50.

Figure 5-14 shows the encoded unicefUSA_out.jpg image.

Command line argument: jpeg.exe –q 50 unicefUSA.ppm unicefUSA_out.jpg

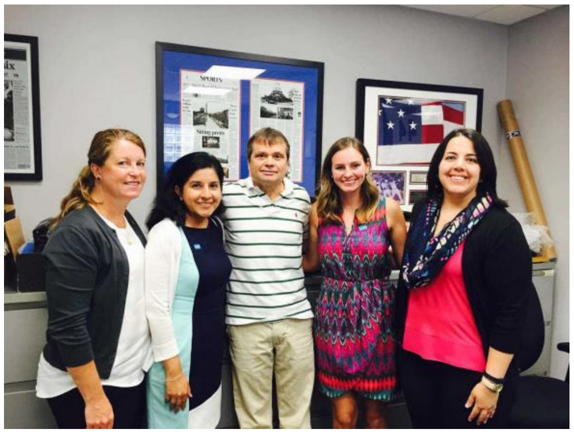


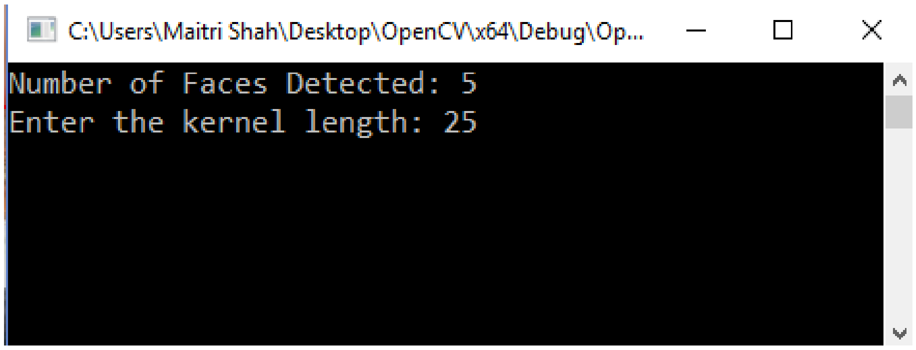
Figure 5-14 Compressed Image (unicefUSA_out.jpg ‘q=50’)

STEP 3

For encrypting the image, the encoded unicefUSA_out.jpg image is passed through the proposed algorithm to obtain unicefUSA_protect.jpg.

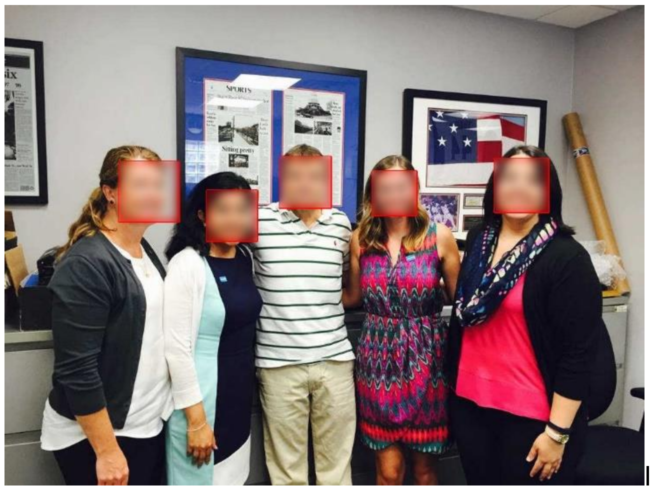
Figure 5-15 shows the command prompt which displays the number of detected faces and asks the user to enter the kernel length.

Figure 5-16 shows the encrypted image and all the cropped original faces.



```
C:\Users\Maitri Shah\Desktop\OpenCV\x64\Debug\Op...
Number of Faces Detected: 5
Enter the kernel length: 25
```

Figure 5-15 Command Prompt Input and Output Arguments



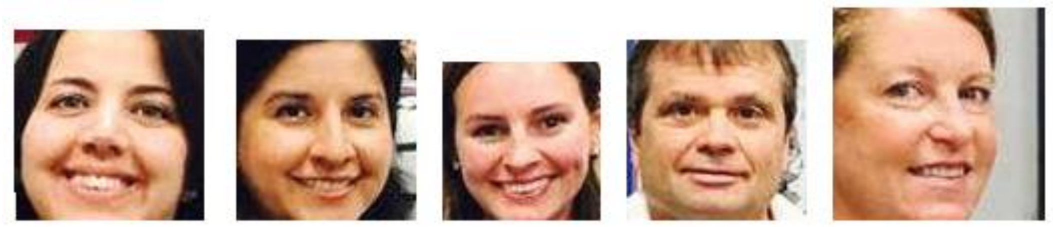


Figure 5-16 Protected Image (unicefUSA_protect.jpg) and the cropped faces

5.2.5 uta.jpg

STEP 1

The original uta.jpg image as shown in Figure 5-17 is passed through the JPEG XT software and decoded to uta.ppm.

Command line argument: jpeg.exe uta.jpg uta.ppm

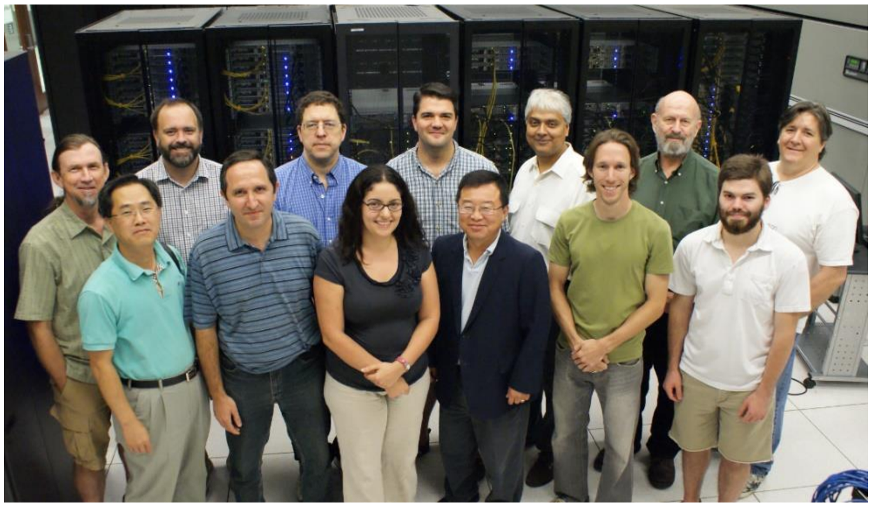


Figure 5-17 Original Image (uta.jpg)

STEP 2

The decoded image is then encoded by setting the base layer quality factor to 75.

Figure 5-18 shows the encoded uta_out.jpg image.

Command line argument: jpeg.exe –q 75 uta.ppm uta_out.jpg

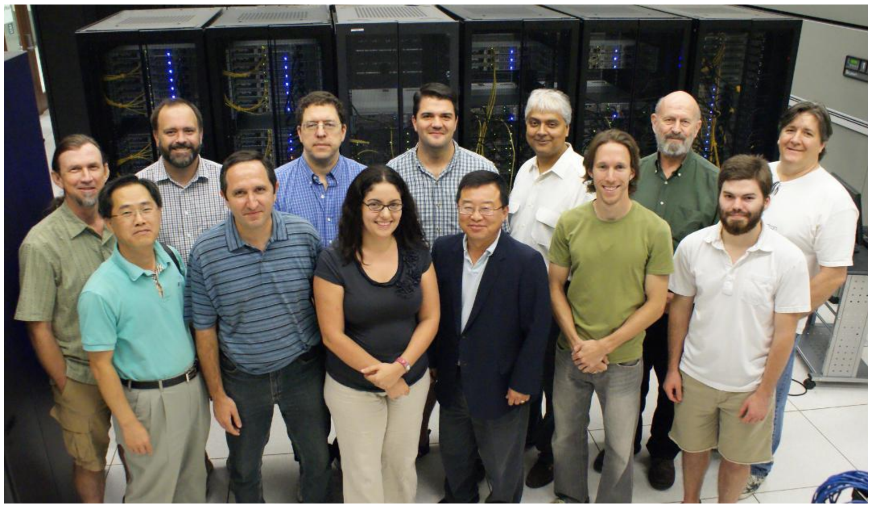


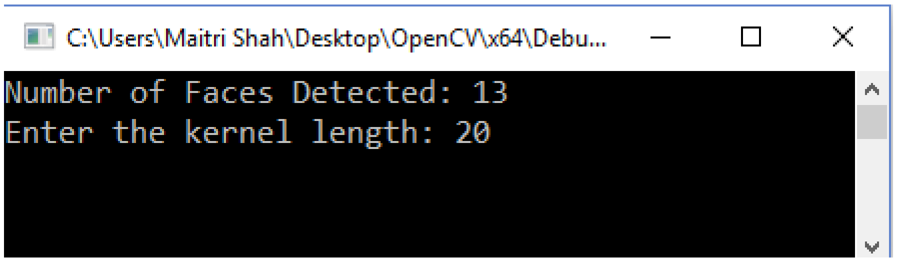
Figure 5-18 Compressed Image (uta_out.jpg ‘q=75’)

STEP 3

For encrypting the image, the encoded uta_out.jpg image is passed through the proposed algorithm to obtain uta_protect.jpg.

Figure 5-19 shows the command prompt which displays the number of detected faces and asks the user to enter the kernel length.

Figure 5-20 shows the encrypted image and all the cropped original faces.



```
Number of Faces Detected: 13
Enter the kernel length: 20
```

Figure 5-19 Command Prompt Input and Output Arguments

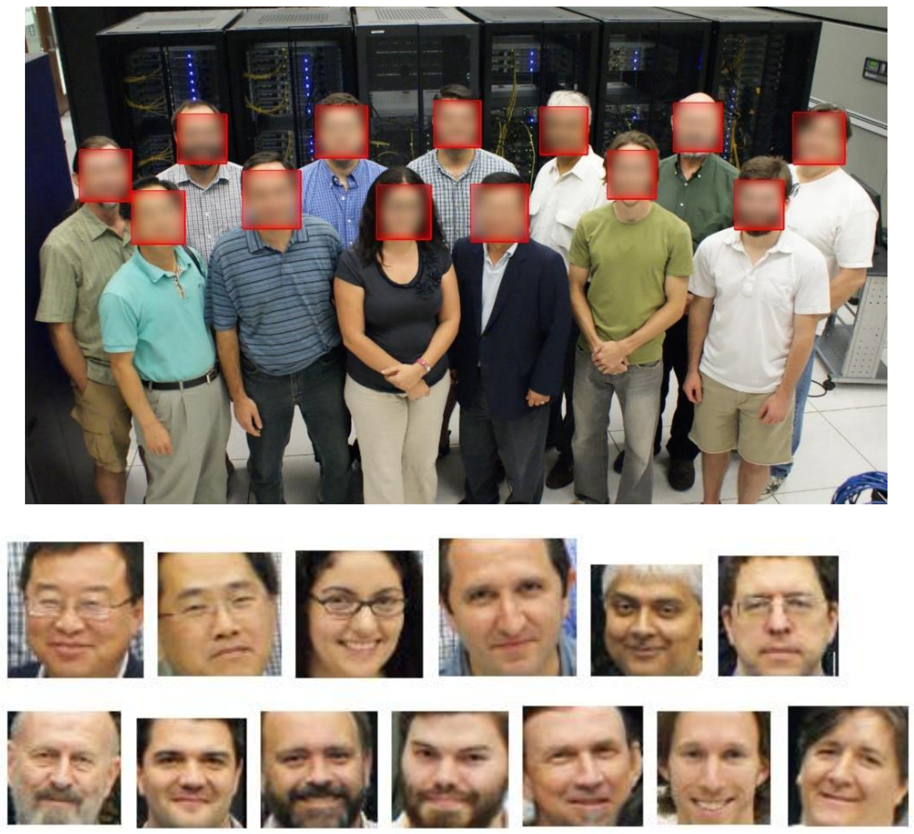


Figure 5-20 Protected Image (uta_protect.jpg) and the cropped faces

5.3 Summary

In chapter 5, outputs of the proposed algorithm are gathered and its performance is observed visually. Chapter 6 talks about the conclusions derived from chapter 5 and lists down different areas which can be focused on, as a part of the future work.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The algorithm has been developed using optimal C++ coding techniques and using the Open CV libraries in Microsoft Visual Studio. C++ has been used, keeping in mind that the JPEG XT code has been written in the same language.

The proposed algorithm performs well for protecting an individual's identity by blurring the image. This will help in avoiding the identity theft which has become a major threat. By resolving the compatibility issues between the Open CV Libraries and the JPEG XT Code, the algorithm can be incorporated in the JPEG XT Code to provide the security feature automatically.

Also, the amount of blurring can be changed according to the user preference by changing the kernel size. This gives the user full control over how the image should be seen by others.

6.2 Future Work

While it is possible to encrypt arbitrary data - text and images - such that only authorized persons can get access to the content, it is also desirable to enable a finer graded control especially on image content. As said, typical applications for extending this algorithm would be social networks where, for example, the general public would only receive scrambled or blurred images, while the entire image would be accessible by only a restricted audience with appropriate credentials. A second application would be that of image databases offering a lower quality, lower resolution preview of images for free, while by buying an additional authorization token, the full image can become available without requiring the client to download the image a second time. [6]

Also, to increase the efficiency of this algorithm, human detection can be incorporated. For a social media application such as Facebook which has been facing some privacy issues, the scrambled image (entire image or a user-defined area) is viewed by those who are not friends, by default. Depending on the data to be hidden, context-aware privacy encryption can also be implemented. Adaption of this algorithm for image privacy applications will help to enhance and protect every individual's identity and privacy.

This algorithm can also be extended to encrypt the videos frame-by-frame. While scrambling every frame individually, the image, sound and text can also be encrypted simultaneously.

Another possible extension of this algorithm is to scramble the detected region or the entire image. The scrambling algorithm for the entire image is developed and tested in MATLAB R2015a for different images. An initial seed is required for this algorithm, which is taken from the user. Based on the seed, the pixels are scrambled by a concept of permutation. While decrypting, the initial seed is required and compared with the seed obtained during encryption. If the seeds match, the decryption of the image is performed, or else the encrypted image is displayed again.

The working and the output of this code is shown in Appendix A

Figures 6-1 and 6-2 defines a method on how scrambling and descrambling can be embedded in the JPEG XT encoder and decoder respectively.

JPEG
Scramble
Descramble
JPEG

Figure 6-1 Secure JPEG Scrambling [54]

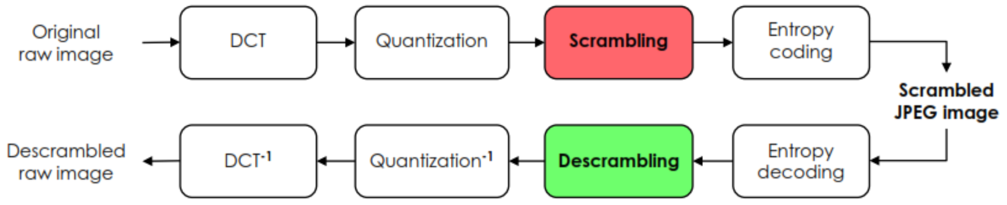


Figure 6-2 JPEG XT encoding and decoding [54]

Appendix A

Scrambling Algorithm-MATLAB

An additional algorithm for protecting the entire image was developed in MATLAB. This algorithm focuses on the method of scrambling the image pixels based on permutation. This type of encryption requires an integer input from the user in the form of a seed. Depending on the value of the seed, the code scrambles all the pixels and provides an encrypted image. In order to decrypt and obtain the original image, same seed is required. Difference in the encryption and decryption seed will output the encrypted image again. To provide double protection, an additional key can also be set along with the seed. The steps for this algorithm are defined as follows along with their results.

Encryption algorithm outline

Step 1: Define an encryption type

Pixels play a very important role in hiding the image data. Therefore, a strong reversible encryption algorithm is needed to rearrange these pixels such that the original pixel positions are recovered which decrypting the image. An algorithm for scrambling the pixels in an image is developed. Scrambling is a process wherein the pixels are randomly moved to a different location depending on the user input seed.

Step 2: Take the user input seed

The users are prompted to enter seed ranging between 1 to 100.

Step 3: Take the user input key

For double protection, an additional key is required from the user.

Step 4: Output the encrypted scrambled image

Decryption algorithm outline

Step 1: Define the decryption type

The decryption algorithm should be the reverse of the encryption algorithm. Hence, descrambling of the pixels will provide the original image.

Step 2: Take the user input seed

This seed is first compared to the initial seed obtained during encryption. If they are same, then the user is prompted for the key, if not then the algorithm stops and the encrypted image is displayed again.

Step 3: Take the user input key

This key is compared to the initial key obtained during encryption. If they are same, then the decryption algorithm is performed as described in Step1 and the decrypted output image is displayed. If the key is not same, then the algorithm stops and the encrypted image is displayed again.

The outputs obtained from test image 1 show successful encryption and decryption while the outputs obtained from test image 2 differ in the user defined seeds, and hence the decryption is not performed.

Outputs from MATLAB

Test Image 1: hdr.jpg

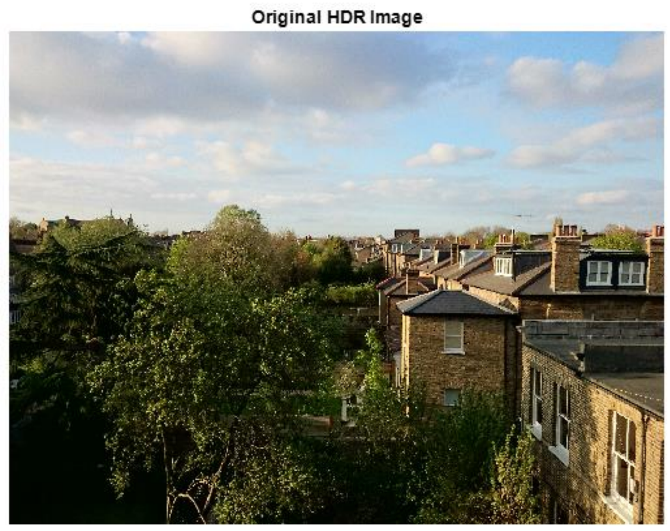


Figure A-1 Original HDR Image

Enter initial seed for encryption: 10

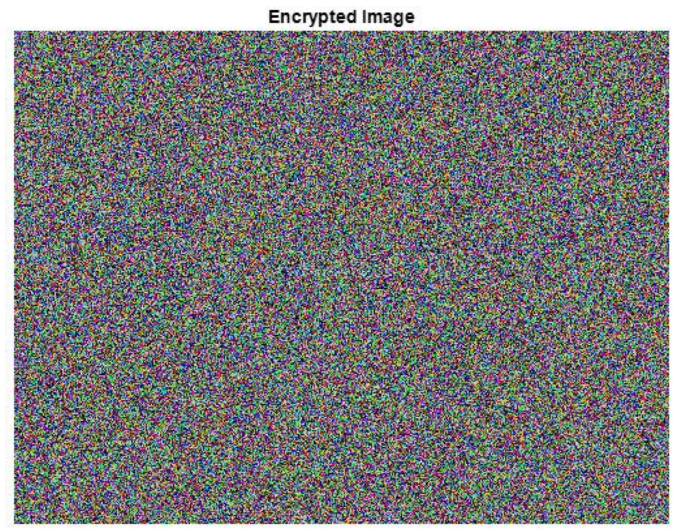


Figure A-2 Encrypted Image

Enter initial seed for decryption: 10

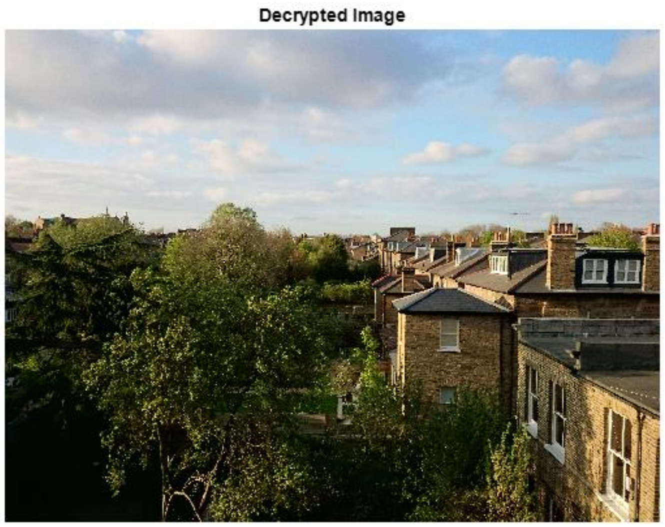


Figure A-3 Decrypted Image

Test Image 2: newyork_hdr.jpg

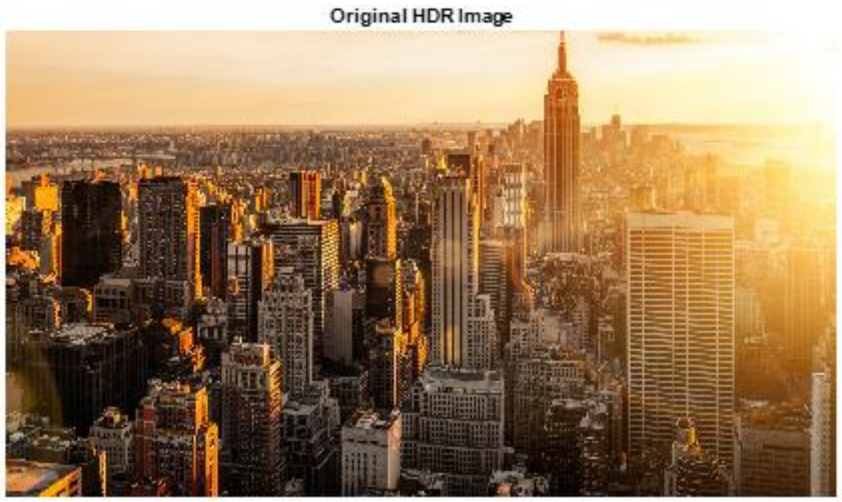


Figure A-4 Original HDR Image

Enter initial seed for encryption: 50

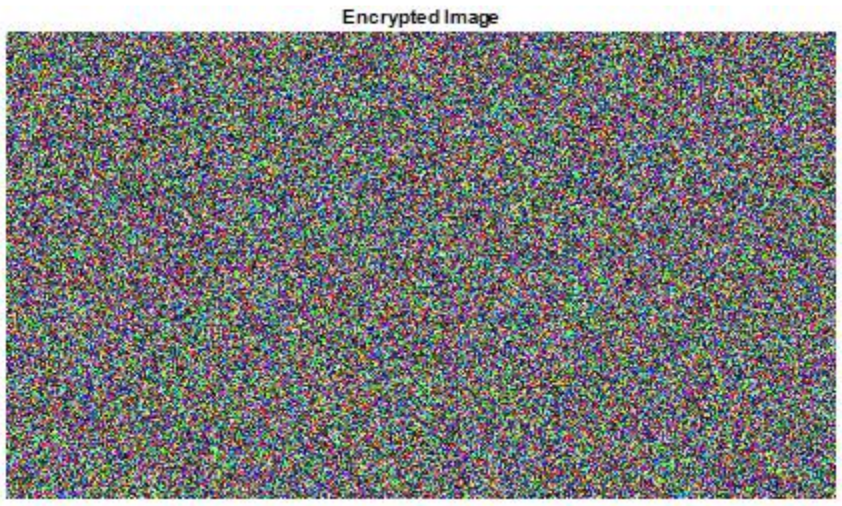


Figure A-5 Encrypted Image

Enter initial seed for decryption: 45

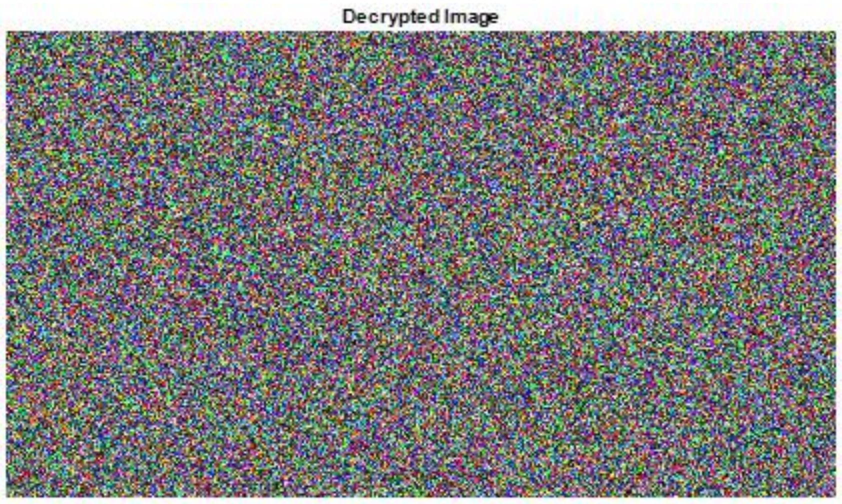


Figure A-6 Decrypted Image

Appendix B

JPEG XT code

Those interested in the code can contact Maitri Shah at maitri.shah@mavs.uta.edu.

The code is developed in Microsoft Visual Studio using C++ and OpenCV libraries. The scrambling algorithm is developed in MATLAB.

References

[1] Overview of JPEG XT

http://jpeg.org/jpegxt/

[2] E. Francois et al, “High Dynamic Range and Wide Color Gamut Video Coding in HEVC: Status and Potential Future Enhancements,” IEEE Trans. CSVT. vol.26, pp.63-75, Jan. 2016.

[3] Overview of JPEG

http://jpeg.org/jpeg/index.html\

[4] Overview of JPEG 2000

http://jpeg.org/jpeg2000/index.html

[5] Overview of JPEG XR

http://jpeg.org/jpegxr/index.html

[6] T. Richter et al, “The JPEG XT suite of standards: status and future plans,” SPIE. Optics + photonics, San Diego, California, USA, vol. 9599, Aug. 2015.

[7] S. Choi et al, “ Performance Evaluation of JPEG XT Standard for High Dynamic Range Image Coding,” in Proceedings of the World Congress on Engineering (WCE), London, U.K., vol. 1, 2015

[8] M. Marcus, “JPEG Image Compression,” June 2014

http://www.cs.dartmouth.edu/~mgm/Final%20Report.pdf

[9] Introduction to JPEG

http://users.ecs.soton.ac.uk/srk/jpeg/JPEG.html

[10] JPEG Family of Standards:
https://iptc.org/download/events/pmdc2015/61_PeterSchelken_JPEG-Standards.pdf

[11] G.K. Wallace, “The JPEG still picture compression standard,” Commum. ACM, vol. 34, no. 4, pp. 30-44, Apr. 1991

[12] Computer Science Rutgers University, Li and Drew 2003 Chapter 9 Fundamentals of Multimedia,
http://www.cs.rutgers.edu/~elgammal/classes/cs334/slide9_short.pdf

[13] M. Rabbani and R. Joshi, “An overview of the JPEG2000 still image compression standard,” Signal Processing: Image Communication, vol. 17, pp. 3-48, Jan. 2002

[14] JPEG 2000 Wikipedia
https://en.wikipedia.org/wiki/JPEG_2000

[15] D. Taubman and M. Marcellin, “JPEG 2000: Standard for Interactive Imaging”
http://www.astro.cornell.edu/~carcich/LRO/jp2/PDS_JPEG2000/Taubman_and_Marcellin.pdf

[16] Benefits of JPEG
http://www.digitizationguidelines.gov/still-image/documents/Buckley.pdf

[17] Overview of JPEG-LS
http://jpeg.org/jpegls/index.html

[18] Lossless JPEG Wikipedia
https://en.wikipedia.org/wiki/Lossless_JPEG

[19] M. J. Weinberger, G. Seroussi and G. Sapiro, “ The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS,” IEEE Trans. on Image Processing, vol.9, pp. 1309-1324, Aug. 2000

[20] Lossless XR Wikipedia

https://en.wikipedia.org/wiki/JPEG_XR

[21] F. Dufaux, G. J. Sullivan, and T.Ebrahimi, “The JPEG XR image coding standard [Standards in Nutshell],” IEEE Signal Process. Magazine, vol. 26, no. 6, pp. 195-199 and 204, Nov. 2009

[22] S. S. Jadhav and S. K. Jadhav, “JPEG XR an Image Coding Standard,” International Journal of Computer and Electrical Engineering, vol. 4, no. 2, April 2012

[23] Cnet discussion forum

http://www.cnet.com/news/better-jpeg-standard-due-in-2009/

[24] Overview of JBIG

http://jpeg.org/jbig/index.html

[25] V.dai,” Binary Lossless Layout Compression Algorithms and Architectures for Direct-write Lithography Systems,” University of California, Berkeley

[26] JBIG Compression Blog

http://www.fileformat.info/mirror/egff/ch09_07.htm

[27] JBIG and JBIG2 Image Compression SDK Technology

https://www.leadtools.com/sdk/compression/jbig

[28] “Module 6: Still Image Compression Standards,” ECE IIT, Version 2, Kharagpur, India

http://www.nptel.ac.in/courses/117105083/pdf/ssg_m6l16.pdf

[29] D. Tompkins “Bi-Level Image Compression”, The University of British Colombia

V. Kyrki “JBIG image compression standard”, March 9, 1999

[30] Overview of AIC

http://jpeg.org/aic/index.html

[31] R. Veerla, Z. Zhang and K. R. Rao, " Advanced image coding and its comparison with various codecs," American Journal of Signal Processing, vol. 2, no. 5, pp. 113-121, 2012.

[32] Z. Wang and A. C. Bovik, H. R. Sheikh, and E.P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600-612, Apr. 2004

[33] Overview of JPSearch

http://jpeg.org/jpsearch/index.html

[34] K. Yoon et al, "JP Search: New international standard providing interoperable framework for image search and sharing," Signal Processing: Image Communication, vol. 27, pp. 709-721, 2012

[35] M. Doller et al,"JPEG's JPSearch Standard: Harmonizing Image Management and Search," IEEE Computer Society, 2013

[36] F. Temmermans et al, "JPSearch: Metadat Interoperability During Image Exchange,"

http://biblio.telecom-paristech.fr/cgi-bin/download.cgi?id=12689

[37] Overview of JPEG AR

http://jpeg.org/jpegar/index.html

[38] M. Kim and K. Yoon, "JPEG-AR Standard Enabling Augmented Marketing," in IT Convergence and Security (ICITCS), 2014.

[39] Call for Contribution on JPEG AR

https://jpeg.org/items/20140301_cfp_jpegar.html

[40] T. Richter, Book chapter on JPEG XT (unpublished)

[41] A. Pinheiro et al, “Performance evaluation of the emerging JPEG XT image compression standard,” in IEEE 16th International Workshop on Multimedia Signal Processing (MMSP), Jakarta, Indonesia, 2014

[42] JPEG-HDR Wikipedia
https://en.wikipedia.org/wiki/JPEG-HDR

[43] HDR Image Quality in the Evolution of JPEG XT
ftp://vqeg.its.bldrdoc.gov/Documents/VQEG_Ghent_Jul13/MeetingFiles/VQEG_HDR_2013__026_JPEGXT_Stuttgart_Richter.pdf

[44] Weber-Fechner’s Law
http://www.cns.nyu.edu/~msl/courses/0044/handouts/Weber.pdf

[45] L. Yuan and T. Ebrahimi, “Image Transmorphing With JPEG,” Multimedia Signal Processing Group, EPFL, Lausanne, Switzerland, Aug. 2015

[46] M. Ra, R. Govindan and A. Ortega, “P3: Toward Privacy-Preserving Photo Sharing,” 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’13)

[47] CNN: Photobucket leaves users exposed.
http://www.cnn.com/2012/08/09/tech/photobucket-privacy-breach/

[48] Facebook Shuts Down Face Recognition APIs After All,
http://www.theregister.co.uk/2012/07/09/facebook_face_apis_dead/

[49] Facebook Face Recognition concerns
http://www.bloomberg.com/news/articles/2015-07-28/facebook-s-use-of-facial-recognition-tool-draws-privacy-concerns

[50] Viola–Jones object detection framework

https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

[51] Cascade Classification

http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html

[52] Gaussian Blurring

http://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html

[53] L. Yuan, P. Korshunov and T. Ebrahimi, "Secure JPEG Scrambling Enabling Privacy in Photo Sharing," Multimedia Signal Processing Group, EPFL, De-ID workshop, Ljubljana, Slovenia

[54] LinkedIn SlideShare

http://www.slideshare.net/ramamurthyaashish/image-encryption-and-decryption

[55] JPEG XT Reference Software

https://jpeg.org/jpegxt/software.html

[56] Microsoft Visual Studio 2013

https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx

[57] L. Yuan and T. Ebrahimi, "Privacy- Preserving Photo Sharing based on Secure JPEG," Multimedia Signal Processing Group, EPFL, Lausanne, Switzerland, Nov. 2015

[58] Test Sequence 1: lena.jpg

http://people.sc.fsu.edu/~jburkardt/data/jpg/jpg.html

[59] Test Sequence 2: friends.jpg

http://www.123rf.com/photo_16035518_group-of-smiling-teenagers-isolated-over-white-background.html

[60] Test Sequence 3: steripack.jpg

http://steripackgroup.com/assets/2_people.jpg

[61] Test Sequence 4: unicefUSA.jpg

https://www.unicefusa.org/sites/default/files/field-images/story-banner/2015/Team%20Quigley%20resize%202.jpg

[62] Test Sequence 5: uta.jpg

http://www-hep.uta.edu/uta-hep-group.jpg

[63] Test Sequence 6: hdr.jpg

http://cnet4.cbsistatic.com/hub/i/2014/04/14/97ca3a75-5f3f-437e-919a-5aa9bfd79c09/ealing-hdr-test-sony-xperia-z2.jpg

[64] Test Sequence 7: newyork_hdr.jpg

http://archwall.xyz/wp-content/uploads/2015/12/skyscrapers-new-york-roofs-skyscrapers-sky-nyc-sun-houses-city-streets-usa-buildings-manhatten-morning-view-background-images-1920x1080.jpg

[65] JPEG XT profiles:

http://image.slidesharecdn.com/jpegbusinessplan-150627125542-lva1-app6892/95/overview-of-jpeg-standardization-committee-activities-16-638.jpg?cb=1435418164

[66] OpenCV

http://opencv.org/downloads.html

[67] A. Bradley and F. Stentiford,”JPEG 2000 and Region of Interest Coding,” presented at Digital Image Computing Techniques and Applications (DICTA), Melbourne, Australia, Jan. 2002

[68] Photo core transform functions http://scc.ustc.edu.cn/zlsc/sugon/intel/ipp/ipp_manual/IPPI/ippi_ch15/ch15_PhotoCT_functions.htm

Biographical Information

Maitri Shah was born in Mumbai, Maharashtra, India in 1992. After completing her schooling from S. M. Shetty School in 2010, she joined K. J. Somaiya College of Engineering to complete her Bachelor of Engineering in Electronics Engineering from 2010-2014.

She enrolled in The University of Texas at Arlington to pursue her M.S. in Electrical Engineering. Maitri’s main areas of interest are Digital Signal and Image Processing. She joined Multimedia Processing Lab under the supervision of Dr. K. R. Rao in Summer 2015.