A Quick Illustration of JPEG 2000

Fall 2003 ECE533 Final Project Proposal
Department of Electrical and Computer Engineering
University of Wisconsin-Madison

By

Kim-Huei Low **Data Fok**

Submitted to: Professor Hu
Dec 12th 2003

1. Introduction

The international JPEG (Joint Bi-level Image Experts Group) and JBIG (Joint Photographic Experts Group) groups, who represent a wide variety of companies and academic institutions worldwide, have created a new image coding system that uses state-of-the-art compression techniques based on wavelet technology. This standard is called the JPEG 2000, which its architecture should lend itself to a wide range of uses from portable digital cameras through to advanced pre-press, medical imaging and other key sectors.

The JPEG 2000 standard has 11 parts [2], in which part 1, the core coding system is now published as an International Standard. Parts 2-6 are complete or nearly complete, and parts 8-11 are still under development. A Java software implementation of the standard (JJ2000) is also found [1], which implements the entire part 1 of the standard.

Although both the standard and the tools are available to the public, but due to the complexity of the standards, especially for amateurs who just begin to learn about image coding, we feel that there is a need to present a quick tutorial of JPEG 2000. This paper is therefore written to give new users a grasp of JPEG 2000.

2. Approach

The most straightforward way to present a quick tutorial of JPEG 2000 is to illustrate and to examine its features one by one in great details. In order to do that, we have familiarized ourselves with the standards and the tools. At first, we have difficulties understanding some of the algorithms and specifications in the standard. However, after several iterations of reading and with the help of the Java tool set, we are able to Figure out all the features in JPEG 2000.

Similar to the JPEG 2000 final committee draft, we will present our work in the same order. We will briefly explain each section of the standard, illustrate each feature, discuss about its applications, and list the pros and cons. For visualization purposes, most of the time, we encode images with extremely low bit rate to show noticeable differences even with small display images. The encoding bit rates are tuned with very fine granularity, base on the feature that we are illustrating.

3. Experiments, Results & Discussions

Annex A, Annex C and Annex D of the JPEG 2000 standard will not be examined. This is because these sections do not consist of a feature. Instead, Annex A talks about the headers and markers which is simply a bunch of constants that are used to efficiently represent an image. Although it is essential to understand Annex C (Arithmetic Entropy Coding) and Annex D (Coefficient Bit Modeling), however, to a user, these are basically the algorithms. Feature wise, they do not play an important role.

3.1 Annex B: Data Ordering

3.1.1 Tile division

In JPEG 2000, an image offset and a tile offset is often given to specify the upper left corner of the desired cropped image. It's expensive to load a huge image in hardware and try to encode it. Therefore, images are typically broken down into multiple tiles, and encoded independently. The Discrete Wavelet Transformation (DWT) is designed for this purpose.

A tile-component is a tile consists of only one component. For example, a RGB image would be broken down into R tile-component, G tile-component and B tile-component. Each tile-component is then further divided down to different resolutions and sub-bands with the use of DWT. Each resolution is divided into multiple precincts which identify a geometric position of a tile-component of an image. Furthermore, each sub-band at each resolution is divided into multiple code-blocks, which will be coded into individual packets.

Shown below is an illustration of the selection of code-blocks for encoding, assuming only two levels of DWT decomposition. Due to the cropping of a tile-component, not all precinct partitions and code-blocks are included for coding. A precinct is only included if the entire precinct falls within the cropped region of the tile-component. Moreover, only code-blocks that are overlapped with the designated precincts are included for coding as shown in

Figure 3.1.1-4.

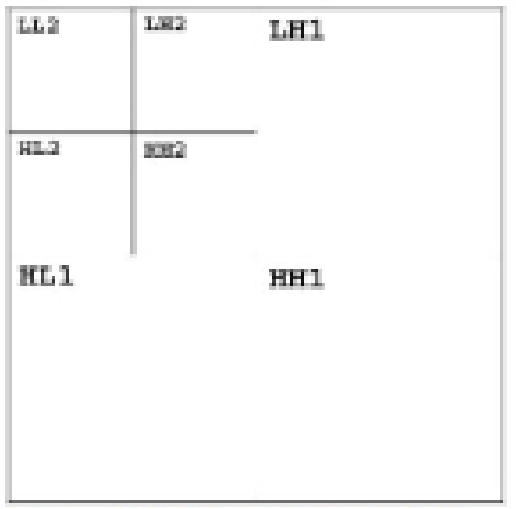

Figure 3.1.1-1: Original DWT

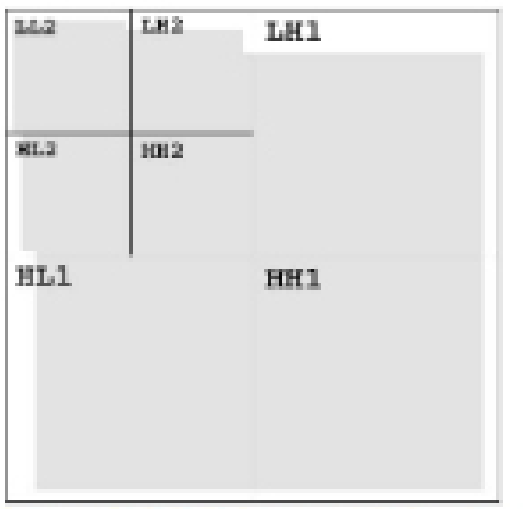

Figure 3.1.1-2: Precinct Selection

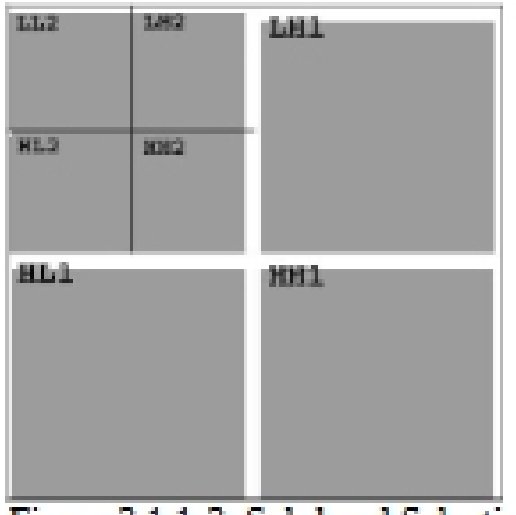

Figure 3.1.1-3: Sub-band Selection

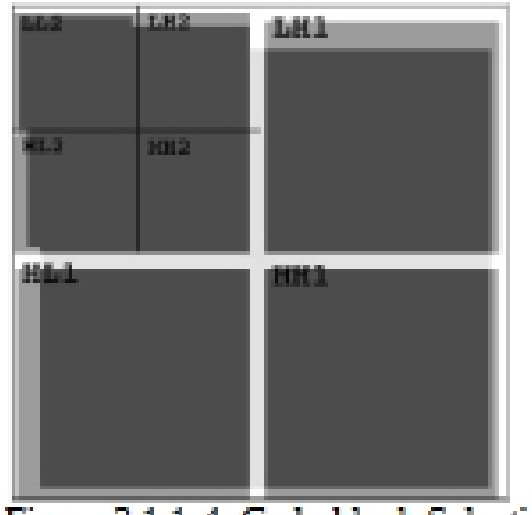

Figure 3.1.1-4: Code-block Selection

3.1.2 Progression Order

For a given tile, the packets contain data from a specific layer, a specific component, a specific resolution, and a specific precinct. The order in which these packets are interleaved is called the progression order. The interleaving of the packets can progress along four axes: layer, component, resolution and precinct.

As shown below, progression in layer and resolution results in sharper image when bit rate increases, and progression in precinct results in overall clearer image, as more precincts/portions of the image are decoded. We also show that progression in component results in less color distortion and increasing contrast as more components are decoded.

There are altogether 5 progression types defined in the JPEG 2000 standard. They are listed below:

1) Layer-Resolution-Component-Position Progressive
   - All positions are encoded before all components before all resolutions before all layers.
   - Image quality is reduced before any color components or parts of the image being thrown away.

2) Resolution-Layer-Component-Position Progressive
   - Visually same effects as Layer-Resolution-Component-Position Progressive.

3) Resolution-Position-Component-Layer Progressive
   - In general, same effects as Layer-Resolution-Component-Position Progressive.
   - Better quality since layer has the highest priority to be coded, trading off some portions of the image.

4) Position-Component-Resolution-Layer Progressive
   - All layers are encoded before all resolutions before all components before all positions.