

Progressive Display of Very High Resolution Images Using Wavelets*

Ya Zhang and James Z. Wang[†], PhD
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802
{yzhang, jwang}@ist.psu.edu

Abstract

Digital or digitized biomedical images often have very high resolutions¹, which make them difficult or impossible to display on computer screens. Therefore, it is desirable to develop a multiresolution display method with which users can freely browse the contents of those high resolution images. In this paper, we present an improved wavelet-based progressive image display algorithm by stressing on the encoding and decoding process. The encoder, which dynamically determines levels of transform and partition of coefficients, is based on a modified Haar wavelet transform. The decoder retrieves the necessary data and reconstructs the requested region at a scale specified by the user. A prototype system, which enables virtually any size of images to be displayed progressively, has been implemented based on this algorithm. The system has low computational complexity for both encoding and decoding process.

1 Introduction

Images are produced and used widely in modern medicine. Medical images such as pathology images usually have very high resolutions. An image, obtained by stitching together images of microscope slides under high power, can have resolutions as high as 20000×20000 pixels, producing several gigabytes of data. Because typical display devices have resolutions below 2000×2000 pixels, it is impossible to display such a high-resolution image on computer screen

in its entirety with all details. Only a portion of the large image can be displayed with high resolution at a time. With the popularity of accessing images over the network, especially with limited communication bandwidth, transmission delay becomes a salient practical problem. For example, transmitting an image of one gigabyte needs more than one minute on a typical intranet.

A user may be interested in any arbitrary region of an image at a certain desired resolution. Thus it is not practical to partition the original image ahead of time in order to process user queries. The method to partition the image after users' demands is also not applicable considering the large size of the data file and slow speed in accessing the secondary storage. It is desirable that the displayed region and resolution of image can be dynamically determined during users' exploration of the image.

Image compression algorithms have been developed to save the storage as well as the transmission time. Many of them are based on Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) [8], or Discrete Wavelet Transform (DWT) [4, 2, 5, 7]. While compression reduces the overall time to transmit an image, access can be made more efficient by exploiting progressive transmission within the compression algorithm. Said and Pearlman proposed in [4] an image multiresolution transform that is suited for both lossless and lossy compression. Their lossless compression ratios are among the best in the literature.

Afework et al. developed a Virtual Microscope system [1, 3], an integrated computer hardware and software system, to stimulate the high-power light microscope. The server program runs on parallel computers with very high processing capability, which makes their system unrealistic for routine image manipulating.

Wang et al. developed a virtual microscope system [6], an earlier version of this work, to display large images progressively. Their system is software-based and suited for real-world applications. The system

*More information about this project can be found at URL: <http://wang.ist.psu.edu/~zhang/wavezoom/>. An earlier work was supported in part by the US National Science Foundation under grant IIS-9817511 and Stanford University. This work is supported by The Pennsylvania State University, the PNC Foundation, and SUN Microsystems under grant EDUD-7824-010456-US. Test images were provided by Keith Cheng, M.D., Ph.D. of The Pennsylvania State University.

[†]Also of Department of Computer Science and Engineering.

¹We refer to spatial resolution in this paper.

uses wavelet transform to dynamically reconstruct the region of interest at a desired resolution and presents to the user in real time. However, the system is unable to handle images of virtually any resolution due to the lack of a process to dynamically determine the wavelet coefficient partitioning strategy.

In this paper, we present an improved algorithm to progressively display images of virtually any resolution. Like our previous work, the algorithm is based on a modified Haar wavelet transform, which maintains lossless reconstruction and permits integer operations. A prototype system with client/server architecture is constructed and can be accessed at <http://wang.ist.psu.edu/~zhang/wavezoom/>. The basic structure of the system is the same as our previous work. For the flowcharts, interested users are referred to [6]. We improved the encoding program and updated the decoding program. The original images are encoded with the modified wavelet transform. The coefficients are partitioned, compressed, and stored for decoding. Compared with the former version of this system, the level of transform and the partition of coefficients are determined *dynamically*. At users' requests, the decoder retrieves the necessary coefficients and reconstructs the queried region of interest at the specified resolution.

The rest of the paper is organized as follows. Section 2 describes the algorithm of the encoder. Section 3 presents the algorithm of the decoder. Section 4 illustrates the performance of our prototype system. Section 5 gives the discussion of our algorithm and prototype system.

2 Encoder

The core algorithm of the encoder is the modified Haar wavelet transform, which allows all computation to be done with integer arithmetic. Considering the large size of the images, integer computation is desired because it can save a lot of storage space. The modified Haar wavelet transform is constructed by ignoring the scaling factor of Haar wavelet transform. Thus, the coefficients are sums and differences of nearby pixels.

We assume that the images are in the RGB color space, consisting of the red component, green component and the blue component. Each color component image is encoded by the modified Haar wavelet transform and stored separately. What distinguishes our work is that the level of transform and partition of coefficients are determined *dynamically* base on the spatial resolution of the image to be processed. The level of transform, s , is calculated with $s = \log_2 \frac{\max(nrow, ncol)}{k}$, where $nrow$ is the height of

the image and $ncol$ is the width of the image, both expressed in pixels, and k is a parameter selected manually. By setting the parameter k , we enforce that coefficient files are smaller than a certain size to achieve fast image access. In our experiments, k is set to 100. Depending on the speed of the computer server systems and the desired processing speed, the parameter may vary.

After determining the transform level s , the right and bottom borders of the image need to be padded according to the transform level so that the size of the image is suitable to the level of the transform applied. The way that these borders are padded is not important because the padded borders are never displayed to the users.

After the padding, we apply the s -level transform to each of the three gray scale images of a color image. The encoding process can be approximated with the following description: 1. Reduce the size of an image to half of its width and height by adding the value of each 2×2 blocks of pixels into one single pixel; 2. Record the sums and the differences between the nearby pixels; 3. Repeat step 1 and 2 until the low-frequency file is reduced to a specified minimum size; 4. Store the minimum size low-frequency file; 5. Store the sets of difference information, i.e., the high-frequency information, in reverse order in separate files. The above encoding process is performed only once for each image inserted.

The transform algorithm is lossless in nature. In order to save the storage, the resulting transform coefficients are quantized in the highest frequency when fully lossless is not critical. To make the whole process to be of low loss, only the highest-frequency coefficients are quantized to fit in the scale of -128 to 128, which causes a maximum loss of 2 out of 510 when the image are reconstructed. Because the fourth and higher high-frequency file is too large for fast retrieval of coefficients, each coefficient file is partitioned and stored as several separate files. The partition of the coefficient files is dynamically determined according to the following two equations.

$$div_c = \frac{2^{k-s}ncol}{m}; \quad 3 < k < s, \quad s > 3 \quad (1)$$

$$div_r = \frac{2^{k-s}nrow}{m}; \quad 3 < k < s, \quad s > 3 \quad (2)$$

where div_c and div_r are the numbers of division on the original coefficient file by column and row respectively, k is the level of the coefficient file, s is the transform level, and m is an adjustable parameter. The size of each partitioned coefficient file is determined by m . We constrain that each partitioned coefficient file is no larger than a certain size. In our experiments, m is

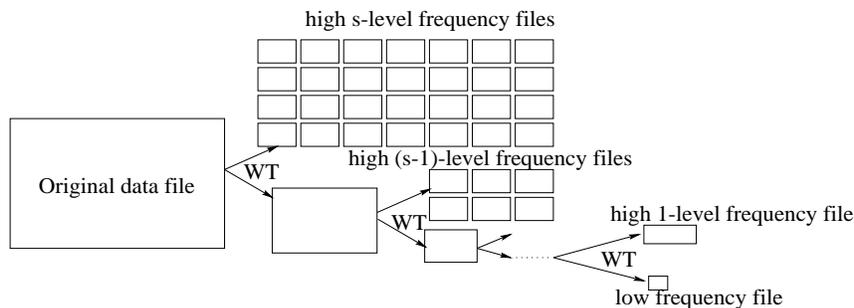


Figure 1: File structure of the coefficients

set to be 1000. Figure 1 illustrates the file structure of an encoded image.

Wavelet transform decomposes an image into sums and differences of nearby pixels. For smooth areas, the difference elements are near 0. Therefore, the coefficient files of a quantized image have a rather peculiar structure. Not only are large areas of the transform image zero, but also the non-zero values are strongly concentrated in the lower-order coefficients. Based on the above characteristic, Huffman coding is employed to further compress the coefficient files and then the compressed files are stored. When a query comes in, the decoder retrieves all necessary files and reconstructs the image.

3 Decoder

Wavelet transform preserves the localization of data, which means that if a user queries a small contiguous portion of a whole image, the decoder only need to retrieve the coefficients corresponding to the queried region. Given a query with x-location, y-location and scale variable, the decoder searches the stored representation files of the image for the related data files, decompresses them and finds the necessary coefficients to reconstruct the block of pixels at the specified scale. The decoding process is the exact reverse of the encoding process. Because of the quantization of the highest-level coefficients, the reconstructed image will have a maximum error of 2 out of a scale of 255.

We have implemented a web-based user interface that allows users to magnify any portion of the images in different resolution scales(as a power of 2). Fig 2 shows some examples of the web interface. The reconstructed image is converted to a JPEG image of very high fidelity to the original and then shipped to the client/user through the Internet.

4 Results

The algorithm is implemented in C language on a Pentium III PC installed with Linux². The prototype system employs a client/server architecture where the server program consists of the encoder and the decoder. The original images are encoded by the encoder program and stored as a set of partitioned coefficient files. When the image query comes in, the decoder program searches for required information to reconstruct the block of pixels at the user specified location and resolution with minimum loss of quality and low computational complexity. Then the constructed image is converted to JPEG and transmitted to the client/user via the Internet.

The major challenge in the implementation is how to hold the entire data in the computer's main memory and conduct the transform. Because of the localization property of the wavelet transform, we can partition the images into blocks and apply the transform block by block without producing any boundary artifact. This strategy greatly saves the computer memory for the transform computation and speeds the encoding process. Our experiments indicate that encoding an image with 5532×2235 pixels takes only 34 seconds, and 17 minutes for an image with 18055×18038 pixels. Generally, for an image of M columns and N rows, the computational complexity for the encoding process is $O(M \times N)$, which is linear to the image size.

Since the encoding is only performed once for each image when it is inserted into the database, the encoding time will not effect the query time. We are more concerned with the retrieval time and reconstruction time. It is desired that the system can perform real-time transmission. Our experiments show that the query process is very fast. It takes less than one-second CPU time to reconstruct an image of 256×256 pixels including the time to convert the image to JPEG format. The reconstruction time varies with the size

²The source code may be obtained by email request to the authors.

Wavelet-based Virtual Microscope Demo

Click in the image below to select the region of interest.



Restart



Go back to project page

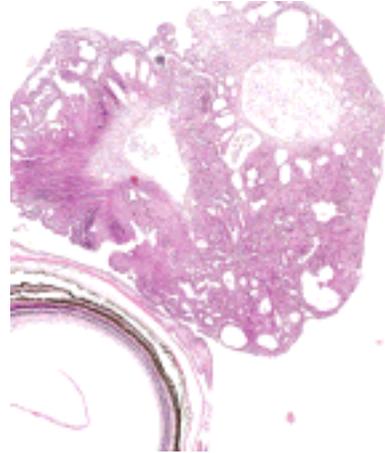
2001 A. Y. Zhang IST, Penn State University

Wavelet-based Virtual Microscope Demo

Click in the image below to select the region of interest.



Restart



Go back to project page

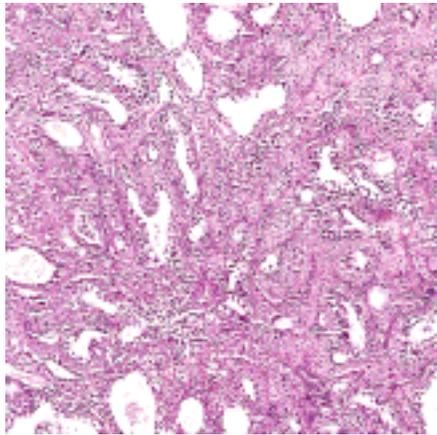
2001 A. Y. Zhang IST, Penn State University

Wavelet-based Virtual Microscope Demo

Click in the image below to select the region of interest.



Restart



Go back to project page

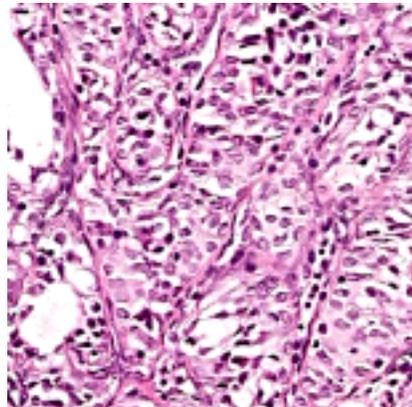
2001 A. Y. Zhang IST, Penn State University

Wavelet-based Virtual Microscope Demo

Click in the image below to select the region of interest.



Restart



Go back to project page

2001 A. Y. Zhang IST, Penn State University

Figure 2: Web client interface

of the retrieved image that is defined by users.

Due to the quantization of the high-frequency coefficients of the first level transform, the maximum error is 2 out of a scale of 255 for the reconstructed image at the original resolution. After the JPEG compression, the error may get slightly higher. Users can adjust the JPEG compression ratio to lessen or increase the loss according to their needs. Since human eyes are not sensitive to such slight variations in color, according to our experiment results, the distortion in color cannot be perceived by human eyes.

5 Discussion

In this paper, we have demonstrated an improved wavelet-based progressive image resolution refining algorithm for display of very high-resolution images. This algorithm is efficient and practical for real-world applications. Because the algorithm is based on only integer arithmetic, it encodes images losslessly or with minimum loss. This property is crucial in many applications such as medical diagnosis. On the other hand, since we dynamically reconstruct images on the fly, any arbitrary portions of the image can be retrieved on users' demands free of boundary effect, which makes this algorithm superior to the ones based on DCT.

The transmission process is in real-time and reconstructing the image with 256×256 pixels generally need less than 1 second. Therefore, users can quickly browse and focus on their regions of interest. Furthermore, the algorithm makes no assumption about the characteristics of images. Thus, it may be extensible to any applications domains that require high quality and flexibility in large image manipulation.

References

- [1] A. Afework, M. Beynon, F. Bustamante, A. Demarzo, R. Ferreira, R. Miller, M. Silberman, J. Saltz, A. Sussman, and H. Tsang, "Digital Dynamic Telepathology - The Virtual Microscope," *Proceedings of the 1998 AMIA Symposium*, Orlando, Florida, November 1998.
- [2] C. Chrysafis and A. Ortega. "Efficient context-based entropy coding for lossy wavelet image compression," *Data Compression Conference*, IEEE Computer Society Press, Snowbird, March 1997.
- [3] R. Ferreira, B. Moon, J. Humphries, A. Sussman, J. Saltz, R. Miller, and A. Demarzo. "The Virtual Microscope," *Proceedings of the 1997*

AMIA Symposium, pp. 449-453, Nashville, Tennessee, 1997.

- [4] A. Said and W.A. Pearlman. "An Image Multiresolution Representation for Lossless and Lossy Image Compression," *IEEE Transactions on Image Processing*, vol. 5, pp. 1303-1310, 1996.
- [5] C. Taswell, "Wavelet Transform Compression of Functional Magnetic Resonance Image Sequences," *Proceedings of the IASTED International Conference Signal and Image Processing (SIP'98)*. Las Vegas, Nevada, 1998.
- [6] J. Z. Wang, J. Nguyen, K. Lo, C. Law, and D. Regula, "Multiresolution Browsing of Pathology Images using Wavelets," *Proceedings of the 1999 AMIA Symposium*, pp. 340-344, November 1999.
- [7] J. Z. Wang, "Wavelets and Imaging Informatics," *Journal of Biomedical Informatics*, vol.34, no.2, pp. 129-141, Academic Press, 2001.
- [8] A. B. Watson, "Image Compression Using the Discrete Cosine Transform," *The Mathematical Journal*, vol. 4, no. 1, 1994.