Search, Controlled Display, and Linking of Segments in Large Format Material Using JPEG2000

Joe Tavares and Greg Zick
DiMeMa Inc.
Seattle, Washington, USA

Abstract

In this paper we describe how JPEG 2000 can be used to define, search for, display, and link specific regions of interests which are subsets of the much larger image (and hence a file subset of a much smaller data file). The advantage of this approach is that these segments can be individually searched; displayed at predetermined access and resolution limits; and linked to supporting or extended data sources.

Introduction

JPEG 2000 is an ideal image coding system for large format materials such as maps and engineering drawings but these large format materials present their own set of challenges and opportunities. Large format materials usually contain a variable number of significant regions of interest. On maps, there might be cities or special areas of terrain. On engineering drawings, regions of interest might include special parts or specific identification labeling.

The ability to search and retrieve based on region is especially appealing for archived historical maps and engineering drawings. High resolution scans of a collection of 1000s of large format maps can be multiple gigabytes in size uncompressed. Researchers are usually interested in specific regions or neighborhoods within maps rather than the complete map. Unfortunately, the older the material, the less likely it is to have spatial referencing to find and/or relate selected regions to other similar regions on different maps or drawings. Using an image-based search, a researcher would need to retrieve each map, find the neighborhood in a low resolution image and zoom-in to the region of interest. With the region-based approach described here, a researcher would be able to index by city or part name. They could search and retrieve the desired region at an appropriate resolution and then zoom or pan within the image or navigate to defined related regions.

The approval of JPEG 2000 as an imaging standard provides new opportunities for storage and archiving of scanned materials. JPEG 2000 provides for high quality, variable compression; spatial, resolution and quality scalability; and both lossless and lossy compression modes. These features of JPEG 2000 support key functions for digitizing, storing, and retrieving digital content, such as incremental transmission of images for high-efficiency zoom and pan for improved viewing, and highlighted viewing regions for navigation support.

Through the use of a well-defined syntax and wavelet compression, JPEG 2000 supports resolution scalability and spatial progression. JPEG 2000 supports better compression performance through the use of wavelets. Wavelets have an additional benefit of allowing the extraction of lower resolution views from the original image. Spatial progression is implemented in JPEG 2000 through the use of a packet which defines an image window. An image window is a specific resolution level, location, quality layer, and component. Packets can be independently decoded so that a JPEG 2000 server can deliver only the information necessary to decode and display a spatial location at a specified resolution and quality level.

The combination of the JPEG 2000 syntax and a JPEG 2000 server are used to construct a region-based retrieval system. Structural metadata written in XML defines a region or image window. Each region has corresponding metadata which is used by the text based search engine within CONTENTdm to search for and retrieve the region. The XML data for that region is then formulated into a request to the JPEG 2000 server which returns the requested region. From the retrieved region, a software user can connect to other related regions specified in the XML region definition file.

JPEG 2000

JPEG 2000, approved as an international standard in December of 2000, is a next generation follow on to the highly successful JPEG image compression format. JPEG 2000 not only improves the compression performance of the original JPEG but also supplies a host of new features that enable a new range of applications. Such functionalities include region of interest coding, improved error resilience, resolution scalability, random access or spatial scalability and quality scalability. This paper uses the new functionalities to enable its region based approach to image management systems.

JPEG 2000 algorithm can be divided into 5 parts: 1) Pre-processing, 2) Wavelet Transform, 3) Quantization, 4) Tier 1 Coding, 5) Tier 2 Coding.

Pre-Processing

This phase consists of preparing data for the subsequent steps. Pre-processing includes level shifting and color transform. Level shifting and color transforms are designed to improve the efficiency and performance of the compression.

Wavelet Transform

JPEG 2000 is based on a wavelet transform. The wavelet transform encodes the image information in a more efficient manner thereby enabling better compression performance. In addition, the transform supports a lossless mode, scalable multi-resolution image storage and retrieval.

The wavelet transform divides an image into 4 subbands LL, HL, LH and HH. The transform is then reapplied to the lower left subband iteratively in order to obtain the number of desired resolution levels. The figure 1 below shows an image divided into 4 subbands with the LL subband further subdivided.

Quantization

After the wavelet transform, quantization is performed. By quantizing the wavelet coefficients compression is greatly improved. By adjusting the quantization scalar one can obtain a target bit rate. Part 1 of the standard referred to as the baseline specifies quantization using a scalar with a deadzone.

Figure 1. Wavelet Decomposition

Tier 1

The first step of Tier 1 coding is to divide the quantized coefficients into rectangular regions called codeblocks. Codeblock size is determined by the encoding process. Codeblock sizes must be the same for all subbands unless they are further constrained by precinct size (precincts will be discussed in the section on Tier 2 coding)

After the subband is divided into codeblocks it is further divided into bitplanes. The bitplanes go through three coding passes. The resultant signal from each pass is coded using a context based adaptive arithmetic coder. The arithmetic coder is the same one as the MQ coder in JBIG2.

Tier 2

Tier 1 coding generates a collection of bitstreams that are combined and ordered in a code stream using Tier 2 Coding. Tier 2 coding enables quality, resolution and spatial progression and scalability.

Quality scalability is achieved using layers. Each layer is a collection of bit planes for every codeblock at all resolutions. Each codeblock may contribute a different number of bit planes to a layer. The encoder determines the number of layers and the bit planes of each codeblock that are included in a layer.

A precinct is a collection of codeblocks. They divide a resolution level into rectangles of size 2^{ppx}, 2^{ppy}. A resolution level is divided into precincts by dividing the LL subband of the parent level. In the case where the code block size is larger then the precinct size then the codeblock is shrunk to fit the precinct dimensions. Precincts can be used to limit packet size, codeblock size at certain resolutions and to enhance spatial progression. In Figure 2 below, an image is divided into codeblocks and precincts; each precinct contains 4 codeblocks.

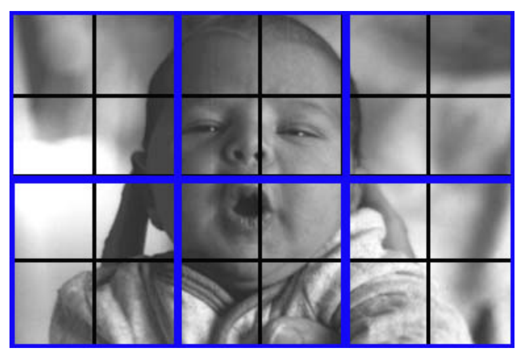

Figure 2. Shows the portioning of an image into codeblocks and precincts. The black lines delineate codeblocks and the gray lines precincts

Finally, the data is organized in to packets. Each packet is defined by its precinct, resolution level, layer and component. Every packet can be independently decoded. Every (precinct, resolution, layer, component) has an associated packet. In certain instances the packet may contain no code stream data.

Using the JPEG 2000 well-ordered code stream above it is possible to identify the data necessary to decode a section of an image at a desired resolution. Utilizing this property, a JPEG 2000 server can extract and deliver an image window (resolution, quality layer, spatial location and component) to a client. Thus, one can view portions of large images over the network.

JPEG 2000 Client/Server Interactive Image Browser

To maximize the power of JPEG 2000 for networked environments there must be a server that is capable of delivering partial code streams and a client capable of displaying them.

The JPEG 2000 server used in this paper was introduced by David Taubman.[1] Taubman's approach is the basis of a working draft of the new ISO/IEC/JTC1/SC29/WG1 work item known as JPIP (JPEG 2000 Internet Protocol) described in document WG1N2607 . JPIP actually consists of two new protocols: jpip-h and jpip-ht. The protocols are based on HTTP 1.1 (RFC2616) and use it for all of its signaling. The difference between jpip-h and jpip-ht is that in jpip-h data is transmitted over the HTTP connection while in jpip-ht only the control data is transferred over the HTTP connection. A separate tcp connection is opened to transmit the data. Therefore, one can discuss JPIP in general since the two protocols only differ in their data delivery mechanisms. In the JPIP analysis we will look at the server in two parts: definitions and communications.

Definitions

Using the JPIP server one can request a group of components of a spatial region at a given resolution for an image. This paper will refer to this region as an image window. Recall from the discussion above on JPEG 2000 packets, this definition fits logically with the four parameters of a packet. The implementation described in this paper always requests all of the image components and all of the quality layers.

Communication

Clients obtain data from the server by making a request using a valid URI (universal resource identifier) to the JPIP server for a particular image window. Requests are made using the HTTP Get method.

Requests must specify a valid URI which determines the JPEG 2000 image filename and the requested image window.

The JPIP Server May Modify the Client's Request. A request can be modified for two reasons: the properties of the JPEG 2000 file or system resources. The server follows a principle of diminishing scope and when modifying a request must never increase the scope of a request. The lone exception is the case where the requested resolution is lower than the lowest resolution available.

System Overview

The goal of a region based system is to allow data providers with a greater granularity of control over their content. Instead of using the image file as a base unit for searching and retrieval, the entire image is simply a special case of the region. The region is defined by its spatial location within the image canvas and by its resolution.

An additional requirement for the design under consideration is to allow content providers to define relationships between regions. The obvious example is that regions within an image have a relationship to each other; however, once one lifts the boundaries of the image file it is easy to imagine many instances where regions would have relationships to regions within other image files for example.

Now we can apply the JPEG2000 principles to a prototype system that allows a user to search across multiple regions regardless of which image they come from; retrieve that region and allow the user to navigate within related regions defined by the content provider.

Regions

The base unit of the system is the region. Regions are the entity that will be searched and referred to as the region of interest (ROI). They must be defined in such a way that they can be indexed for fast searching and support a scheme to define such relationships between regions.

In JPEG 2000, a ROI is a defined as an area that is coded with more fidelity than the background. A region of interest is generally indicated by prioritizing the coefficients of the ROI so that the region is decoded before the background. Two approaches are defined in the JPEG 2000 standard: *The max shift method* and a *general scaling method*. Both of these methods are limited for the use of JPEG2000 describe in this paper because they are static ROI schemes. These static schemes require the region of interest information to be embedded within the code stream. Such an approach limits flexibility and makes it difficult to index the regions for searching and retrieval

A region based retrieval system requires dynamic creation, modification and deletion of regions. Having dynamic regions allows content providers to make content available sooner and grow their data and granularity incrementally. Therefore, the static ROI schemes are not appropriate.

A dynamic approach was proposed by Rosenbaum and Schmann.[3] Their approach uses a transcoder to insert a layer and move all non-important layers up. Therefore, one can effectively split a layer into two with the first one containing the information for the ROI and the second containing the background information. The transcoder does not have to decompress the image, it simply needs to redo the Tier-2 coding. Such an approach does introduce some overheard.

The design described herein depends on a JPEG 2000 server that is capable of delivering regions of an image to a client. Regions are defined externally using XML, the XML can then be parsed and used to request a region from the server. By describing the regions in XML the scheme is flexible and expandable to help meet future needs and enables one to easily add, delete or modify regions. In addition, the approach provides an entity for indexing which will allow one to efficiently search for regions.

When a region is retrieved, the XML will be parsed and then a request will be made to the JPEG 2000 server. Since, the server can not serve anything smaller then a precinct, the granularity of the region is dependent on the precinct size.

Each searchable region is represented by an XML index file. The index file defines the image window that fully defines the location of the region in x, y, resolution space. Only rectangular regions are defined, however, it is trivial to extend the XML syntax to support arbitrary shaped regions.

In addition to the fully specified image window, each index file contains one or more parents. Each of these parents, described in the next section, define relationships between the current region and other regions.

Defining Relationships

One can look at an image as a collection of regions with defined spatial relationships. A region is defined because it is significant. Further, it is often the case that a region's relationship to other regions contributes added meaning. An example of this is shown in figure 3. It shows a collage from the University of Washington Libraries WTO Collection (http://content.lib.washington.edu/WTOweb/index.html). While, each segmented region shows a snapshot of the protest and is a valuable entity on its own, the juxtaposition of the regions to each other conveys additional meaning.

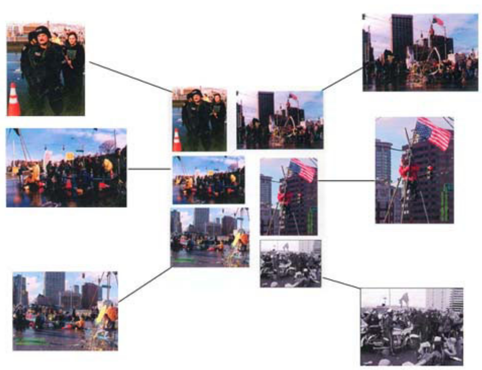

Figure 3. A collage from WTO protest is composed of 6 different regions

Whole images define relationships between pixels by storing pixels in a certain order. It is the order of the pixels that tells one the relationship between one pixel and another. Since the image has been segmented into regions, if one is to study such regions, it is important to define a structure to describe such relationships. Relationships do not only exist between regions within the same image, but also may exist between regions within different images.

A natural choice to represent these data relationships is XML. XML is a markup language designed to create a structural model of data such as the structure that is needed to represent the region relationships. These region relationships are defined using XML in a parent file. The parent contains pointers to the index files for the related regions. Since both the parent and the child (index file) maintain links to each other one can navigate from the child to the parent and the parent to the child.

System

The above sections were primarily concerned with describing the underlying infrastructure designed to support the system developed in this paper. This section describes a prototype that was developed to demonstrate the system.

The database chosen for the prototype is CONTENTdm™. CONTENTdm by DiMeMa (contentdm.com) is a popular database for special collections in the library market. Many of the envisioned application areas for the application described in this paper fall into the realm of special collections such as maps and newspapers.

CONTENTdm performs text based searching on user entered metadata. It can retrieve one record amongst millions in less than a second. It has a web based search client that makes client deployment trivial. Each object in CONTENTdm will be a URL to the region definition file. By default a thumbnail is precomputed and stored with each object. Such an approach allows a user to quickly browse search results. Once a user has found the desired region double clicking on the thumbnail will retrieve and display the region.

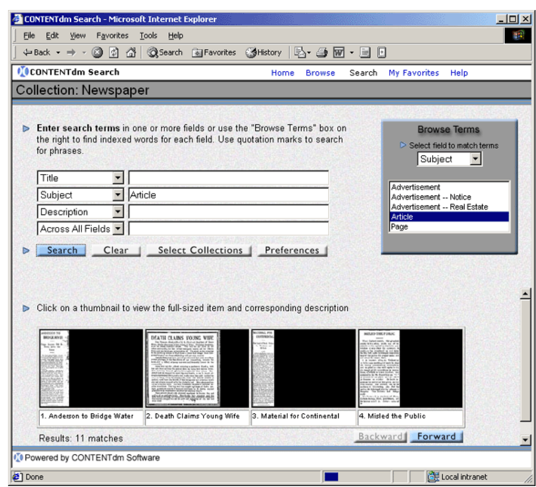

Figure 4. CONTENTdm Search Client

The JPIP client is based on Kakadu's kdu_show application. Kakadu, developed by David Taubman, is a JPEG 2000 software toolkit (http://www.kakadusoftware.com). When the database passes the URL to the client the client downloads the region definition file. From the region definition file the client downloads each of the parents and the region definition files referenced by those parents. The information is then presented in a hierarchal menu to the user (Figure 5).

Once all of the regions and relationship information is downloaded, the client requests the region from the JPIP server, displays the region at the appropriate coordinates and resolution within the image canvas and centers the display on the region.

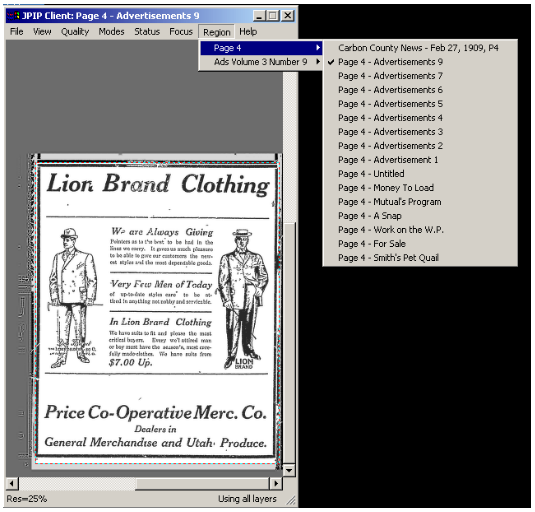

Figure 5. JPIP client showing region relationships in region menu

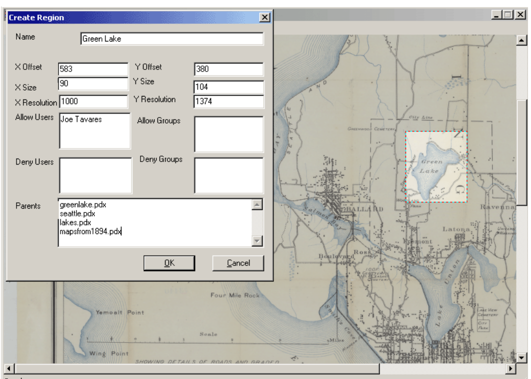

Figure 6. Example of creating a region using the developed region creation tool

The server used is Kakadu's JPIP server with only one slight modification. The simple modification allows the region definition and parent files to be downloaded from the server.

Allowing download of the region definition and parent files is trivial. Simply by checking for appropriate file extensions in the resource portion of the URI and sending those files over the network the requirement was met.

For the prototype to effectively illustrate the region based system described herein regions within images must be defined. To help construct the necessary XML files for region definition a simple tool was developed. Using the tool, users click and drag the mouse to draw a rectangle over the region of interest. Once the region is boxed, they simply choose Define Region from the menu. A dialog will then be displayed (see Figure 6) with the coordinates of the image window already defined (x, y, width, height, x resolution, y resolution). The image window can be modified in the dialog and users can enter the xml files containing related region. Once a user hits okay the appropriate files are created and/or modified.

The prototype combines and extends several existing components to demonstrate a system based upon the design defined in the previous section. It shows the potential for how a system can be successfully built to address several application areas.

Conclusion

The approach for region indexing described in this paper is based on region definition and processing using XML and JPEG 2000. Relationships are defined between regions allowing a much more complex data representation. This enables the use and representation of visual data from large format material in an effective way. The extending software developed provides powerful search functionality based on common name, type or theme searching and is easily supported. The approach described in this paper extends the capabilities of modern image databases to retrieve in one step a specific region from millions of images, each with multiple regions.

References

1. David Taubman, "Proposal and Implementation of JPIP (JPEG2000 Internet Protocol) in Kakdu v3.3)", Aug. 9, 2002.
2. Majid Rabbani and Diego Santa-Cruz, "The JPEG2000 Still-Image Compression Standard" 2001 ICIP, October 11, 2001.
3. Rene Rosenbaum, Heidrun Schumann, "Flexible, Dynamic and Compliant Region of Interest Coding in JPEG2000".

Biographies

Joe Tavares received his B.E. degree in Electrical Engineering from the University of Washington in 1996 and a M.S. in Electrical Engineering in 2003. Joe is currently a software developer at DiMeMa Inc. He has been working with CONTENTdm since 1998. His work has primarily focused on the storage, acquisition and indexing of images.

Greg Zick is a Professor of Electrical Engineering at the University of Washington. His interests are in digital archive management systems. He has worked in the areas of medical imaging, information technology and digital Libraries. He currently is President of DiMeMa Inc.

info@contentdm.com