International Journal on Digital Libraries manuscript No.
(will be inserted by the editor)

A Framework for Scalable Delivery of Digitized Spaces

David Gotz and Ketan Mayer-Patel

University of North Carolina at Chapel Hill, CB #3175 Sitterson Hall, Chapel Hill, NC 27599

Received: date / Revised version: date

Abstract. We present a framework for scalable delivery of digitized environments. Today's digital museums distribute images, text, sounds, and videos. Our work targets the distribution of a more advanced media type which allows users to independently and interactively explore digitized spaces. We describe a multidimensional and multiresolutional representation which maps directly to a set of communication channels. Clients receive data by subscribing and unsubscribing from these channels. Client adaptation to current application and network conditions is performed by managing a working set of channels. This mechanism enables distribution of digitized environments to large groups of independent digital museum visitors.

1 Introduction

Museums are an essential institution in modern society. They serve as a repository for the world's cultural legacy, providing resources for both preservation and dissemination of society's greatest treasures. Traditional museums offer a centralized location for access to these materials, making information more accessible to visitors. Unfortunately, while "brick-and-mortar" museums have many benefits, they are also limited in important ways. In particular, there are two critical limitations to traditional museums that restrict the availability of information.

The first critical limitation of traditional museums is restricted exhibition space. Typically, museums have only a small percentage of their collections on display at any given time. The majority of a museums holdings are often held in storage, hidden from visitors and available only for special exhibits.

The second critical limitation is one of geography. A physical building can only be at one place at a time. This makes any given collection inaccessible to the great majority of the world's population simply because it is located in a remote location.

1.1 *The Digital Museum*

The advent of digital communications - most notably personal computers, digital imaging, and the Internet - has led to the notion of Digital Museums. This new breed of museum promises to dramatically impact the critical physical limitations of both limited exhibition space and geographic location.

Limited exhibition space is already being addressed by the mass digitization of collections via scanning technology, digital photography, and digital audio recordings. As the required technology becomes more affordable and ubiquitous, digital collections are appearing in museums via computer kiosks, movies, and other multimedia presentations. These new tools allow for the exhibition of a much larger portion of a museum's collection.

At the same time, the geographic limitation is being addressed by combining digitized collections with the Internet to form online digital museums. Museums have extended access to their holdings by exposing digital collections to the world via the World Wide Web. Armed with a personal computer and an Internet connection, individuals can explore artifacts housed around the globe. Institutions both large [13] and small [19] have significant online digital museum collections.

1.2 *Continuing Challenges*

Digital museums remain limited despite the major strides they have made over the past few years. In today's digitized collections, shared media is largely limited to small artifacts such as still images, short movie clips, digitized audio, and text. While challenges remain in building interfaces and frameworks for indexing, searching, and sharing the enormous amounts of digital data, these media types have been largely successful at representing a number of artifacts including pictures, sculptures, scientific achievements, and important documents. However, an important class of exhibits remains out of reach. Important spaces and environments, such Notre Dame or the White House, can only be minimally captured using standard techniques.

In recent years, several researchers have explored the digitization techniques needed to create a photorealistic recreation of *space*. These techniques range from image-based rendering (IBR) techniques that reconstruct a scene from a large input set of digital pictures [2,18] to range-scanning devices that capture accurate, high complexity geometric models [17,20]. These new and powerful tools go far beyond traditional media types. They use massive amounts of digitized data to recreate entire environments and allow individualized interactive experiences.

When applied to digital museums, these techniques promise to enable virtual visitors to explore important places as if they were there themselves. Users will experience the digitized location as if they were personally controlling the path of a remote video camera through the space. These tools can be used to create a truly compelling virtual experience by capturing the full complexity and richness of an environment.

Many museums (for example, the Nobel Museum [8]) are already embracing early technologies, such as Quicktime VR [4] or IPIX [6], which provide limited interactivity with panoramic images. However, these technologies are limited to a single still image and restrict user interaction far more than more advanced techniques such as IBR.

Unfortunately, the very complexity and interactivity that make advanced IBR algorithms visually compelling also make these digitized environments extremely large in size and difficult to distribute over the Internet. The amount of data required by IBR algorithms can make video data seem tiny by comparison. Furthermore, the interactive nature of these media types requires that each visitor have control over the order in which data is transmitted. This is in contrast to video where each user views the file in the same way. The individualized data requirements make it a challenge to provide service to any more than a few users at one time.

To date, the use of free viewpoint environments has been largely limited to in-museum kiosks or displays [15, 21]. Without solving the problem of distribution for large user populations, these new media types will remain impractical for online digital museums.

1.3 A New Framework

Throughout this article, we will explore the problem of representing and distributing large digitized environments to digital museum visitors. Specifically, we'll concentrate on IBR applications based on the *Sea Of Images* algorithm [2]. We'll present a framework for representing, communicating, and interacting with IBR datasets as we begin to answer a number of critical questions: How can large digitized scenes be distributed to digital museum visitors across the Internet? How can the required data be represented in a way that facilitates distribution? How can this be accomplished so that it can support large numbers of virtual visitors?

The remainder of this article is organized as follows. In Section 2 we present an overview of related work. In Section 3 we present a more detailed description of the problem. Section 4 describes our proposed framework for solving this problem. In Sections 5 and 6 we present our prototype system and and an evaluation of its performance. Finally, in Section 7, we conclude and discuss areas for future research.

2 Background and Related Work

This section provides background on common media types deployed in digital museums, more recent advances in immersive media including image-based rendering (IBR), and related multimedia techniques for encoding and communication. These concepts provide a general context for our work which concentrates on a framework to support scalable remote access to IBR datasets.

2.1 Media Types in Today's Digital Museums

Today's digital museums utilize a variety of media types in their online collections. Common media types range from simple text to interactive panoramic images. Using digital technologies, museums can offer vast collections online for thousands of virtual visitors.

2.1.1 Text and Still Images

The core holdings in most online museums are formed of both textual descriptions of artifacts and still digital images, captured with digital cameras or digitized from traditional photographs using scanners.

In recent years, museums have undertaken the arduous task of digitizing artifacts such as paintings, sculptures, and documents. Using a variety of imaging techniques, these digitized artifacts are made available over the Internet for users to browse at their leisure. The emergence of digital image collections has been brought about through both existing technologies (such as the World Wide Web, scanners, digital photography, and the JPEG image compression standard) and new techniques developed to simplify and manage the processes of digitization, organization, and distribution.

2.1.2 Video and Audio

Several digital museums have film archives or short video clips as part of their collections. Using standardized video representations such as MPEG or AVI, these museums can digitize artifacts in a way that offers a unique experience for visitors. For certain artifacts, video technology can lead to a richer experience for users.

One drawback of video collections is the added storage and communication cost. For large video files, the time required for a user to download the entire video clip can be excessive, even when using a broadband network connection. To overcome this limitation, video streaming technologies have been developed to *stream* data to the user, allowing them to process data as it arrives rather than waiting for the entire datafile. Several commercial products are available for video streaming, including RealNetworks' Real Player and Microsoft's Windows Media Player.

In addition to video archives, many digital museum collections contain audio clips. Standardized audio formats, such as MP3 or WAV, can be used to store audio files. Decoders for these formats are usually integrated into web browsers for easy downloading and decoding by museum visitors. Just as with video, audio files can be large in size and streaming technologies have been developed to ease distribution. Again, both Real Player and Windows Media Player support streamed audio.

2.1.3 Interactive Media

More recently, digital museums have begun to incorporate interactive media types into their collections. These media types can enhance the experience of digital museum visitors by allowing users to interact with digital artifacts. Java-based applets, Flash, and Shockwave files have been used to create dynamic online exhibits at a number of digital museums.

In an attempt to represent virtual places, many digital museums have incorporated interactive panoramic images which allow a user to look around a scene as if standing still and turning their head. This technology has developed from Quicktime VR [4] and is now widely deployed using tools from companies such as IPIX [6].

Quicktime VR and its derivatives offer more interactivity than still images, but users still lack the ability to choose a path through the environment. The eyepoint is restricted to a single point in space and the user has control over only the gaze direction.

2.2 Free Viewpoint Environments

Researchers have been working on several methods for digitizing artifacts which allow truly interactive experiences. These techniques create a spatial representation of the artifact using various sensors. Users are then allowed to navigate through space, examining the artifact from any viewpoint and looking in any direction. We refer to these as *free viewpoint environments*. These environments offer significant advantages over more traditional digital media in that users are free to look where they like and explore what interests them most. Users are no longer restricted to fixed viewpoints or static video streams of an environment. Two major areas of research in support of free viewpoint environments are three-dimensional scanners, which digitize geometric information, and image-based rendering techniques, which use large sets of two-dimensional photographs to synthesize novel views of a digitized environment.

2.2.1 Three-Dimensional Scanners

Active sensors have been used to map out the three-dimensional shape of artifacts. These scanners can be used to digitize both individual objects as well as entire spaces or environments. For example, three-dimensional scanners of this type have been used in museum settings to digitize several Michelangelo sculptures [17]. Other researchers have digitized the inside of Thomas Jefferson's historic home of Monticello [20].

These techniques often use laser beams or other active sensors to take precise geometric measurements of an environment or object. From the sensor readings, algorithms develop a full geometric description of the scene. Color images can be combined with the geometric description to form a nearly photorealistic model of the artifact.

2.2.2 Image-Based Rendering

In image-based rendering (IBR), cameras are used to collect sets of photographs from real-world scenes. The captured images are treated as samples from the complete seven-dimensional plenoptic function [1].

IBR algorithms allow a user to navigate through a virtual space, reconstructing the user's view from the original set of input photographs. IBR systems essentially interpolate between nearby image samples to render the view for a user at any point in space. From a signal processing point of view, IBR algorithms attempt to reconstruct the plenoptic function from the captured image samples. There are several working systems of this type including Light Fields [16], Lumigraphs [10], Concentric Mosaics [23], and the Sea Of Images algorithm [2].

2.2.3 Sea of Images

The Sea of Images (SOI) algorithm [2] is the most important IBR algorithm with respect to our research. Our prototype specifically targets SOI datasets and we perform our evaluation using the SOI algorithm. For this reason, we present a more detailed review of this algorithm.

The SOI algorithm is well suited to digital museum applications. Unlike most IBR methods, which are often limited to a third-person perspective of objects, SOI is designed to support first-person, "inside looking out" navigation of digitized spaces. The first-person paradigm is similar to the experience of a person physically walking through a space. It allows users to feel immersed in a digitized environment, as if they were actually present within the space. It is for this reason that we have selected the SOI algorithm for our work.

The input dataset for SOI is a set of panoramic images. Each image contains a full 360° view of the scene. Digitization is typically accomplished by attaching an omnidirectional camera to a motorized cart at some fixed height. The cart is maneuvered through the space while the camera captures a series of photographs. The camera pose for each photograph is computed and stored as an annotation to the image.

The camera positions for all of the input photographs lie along a plane at eye level. A mesh along the plane is formed using a Delaunay triangulation of the camera positions. This mesh is shown in Figure 1(a). Vertices in the mesh represent the position of captured images. The triangulation is used to reconstruct the scene from any arbitrary viewpoint.

Reconstruction begins by locating the virtual viewpoint within the triangulation. The triangle that contains the viewpoint is identified, and the three photographs represented by the vertices of the triangle are used for reconstruction. These are the three closest images in the entire database. Figure 1(a) shows a virtual viewpoint and the associated triangle from the mesh. Once the three photographs have been identified, a novel image is created by interpolating between the three source images as shown in Figure 1(b).

2.2.4 Deployment In Museums

Free viewpoint environments are an exciting new media type. However, deployment in museums has been slow due to some critical drawbacks. They typically require an extremely large input dataset and have particularly high communication requirements making it very difficult to support remote access. Even for a locally running system, dealing with the limited bandwidth of disk operations is a challenge [2]. Accessing data over the far more narrow bandwidth of a network is even more difficult.

For this reason, current experiences with free viewpoint environments in museums have been largely limited to kiosk-style exhibits installed in traditional brick-and-mortar museums. Examples of such exhibits are the Digital Michelangelo kiosk at the Galleria dell'Accademia [15] and the Virtual Monticello exhibit at the New Orleans Museum of Art [21].

The framework we present in Section 4 is designed to overcome the size and bandwidth problems through representation design and communication mechanisms. With a framework in place to support scalable remote access to these databases, we hope to enable the adoption of these powerful media types by digital museums.

2.3 Multimedia Techniques

Our framework has been greatly influenced by developments in the field of multimedia. A number of the specific problems in our research have close parallels in other multimedia areas. Two such topics are media representation and media communication.

2.3.1 Media Representation

Several media representations have been developed to store digital artifacts. A well-designed representation reflects assumptions about the nature of the artifact itself as well as the manner in which an application will make use of the digital artifact. This fact explains why a fairly simple artifact, such as a still image, can be represented in several different ways. The same is true for video formats.

A large number of standards have been defined for still image representation. The need for many of these formats is driven by the nature of the image. Complex real-world images with millions of colors are often stored as JPEG images which use a block-based DCT encoder specifically designed for handling these types of photographs. Simpler images with just a few colors are often better stored as GIF images. The GIF format is specifically designed for images with 256 or fewer colors. Synthetic images, drawn using geometric constructs like lines and shapes, might best be stored as vector-based formats which efficiently store geometric descriptors rather than actual image data.

More recently, the JPEG2000 standard has been released. This format uses a wavelet based encoder to improve the visual fidelity of compressed images. Perhaps more importantly, the JPEG2000 format uses a *progressive codestream* which organizes data in order of importance and enables variable-resolution decoding from a single encoded image file.

Just as with image formats, several video formats have been defined to meet the needs of different applications. The MPEG format is commonly used to represent standard resolution video files. The H.261 format has been defined specifically for low bitrate video files. More sophisticated layered video formats have been developed to allow variable-bitrate decoding from a single encoded video file.

2.3.2 Media Communication

The communication and distribution of digital media types is a well studied area of research. From sharing pictures on the World Wide Web to video-on-demand,

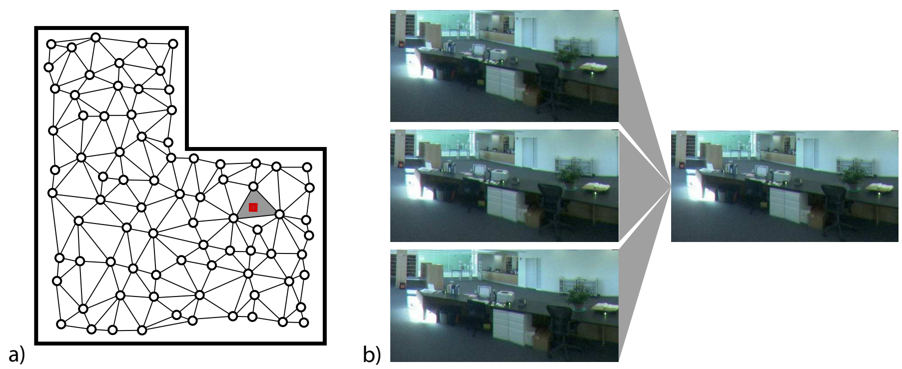

Fig. 1. *The Sea of Images system by Aliaga et al. [2] generates novel views of a scene from a collection of images. (a) The system starts by triangulating the positions of all captured images in a plane. Given the position of a user, represented by a square in the figure, reconstruction begins by finding the triangle containing the user's position. This triangle is shown in gray. (b) The algorithm uses the three images associated with the triangle's vertices to generate the novel view.*

media types of all sizes have been transmitted across digital communication networks using a variety of protocols and mechanisms.

In general, there are three network delivery mechanisms. These are unicast, broadcast, and multicast. Unicast delivery sends out an individual packet to all interested parties. Under unicast, data packets are duplicated for recipients that require the same data. For this reason, unicast is most appropriate when very few recipients are interested in any one piece of data. As the number of recipients with a common interest grows, the high duplication rate makes unicast inefficient.

At the other extreme is broadcast. Broadcast sends out only a single copy of each packet. It is therefore very efficient when many recipients are waiting for the same information. However, broadcast transmits every packet to every recipient, regardless of the recipient's needs. On today's Internet, that would entail the hugely impractical task of sending every data packet to every host in the world.

Multicast delivery mechanisms have been proposed as a middle ground between unicast and broadcast. Conceptually, multicast sends out just one copy of each packet. That single packet is then sent only to the group of interested recipients. A number of techniques have been proposed to support multicast distribution including IP Multicast [7] and several versions of peer-to-peer or overlay multicast [3, 5]. These techniques differ in many ways, but all are designed to support the general multicast concept.

2.3.3 Video-On-Demand

Multicast and broadcast improve the efficiency of transmission, but they limit interactivity because all interested parties receive the same data packets. However, for some applications, sufficient interactivity has been achieved via intelligent data organization and channel structuring. For example, pyramid broadcasting [24] and its derivatives [12, 14] are used in video-on-demand systems. Using these techniques, users have individual control over video streams even though they are transmitted over broadcast networks.

However, the interactive control afforded by these techniques is limited. Users are restricted to navigating only a single dimension: time. Furthermore, users are further restricted in that they must start only at the beginning of the movie and can not fast-forward.

The work presented in this article is inspired in part by these algorithms for scalable support for video-on-demand. However, our work extends these ideas to multiple dimensions of adaptivity. Furthermore, we allow arbitrary navigation patterns through digitized spaces where each user receives an individualized flow of information.

3 Problem Description

The ultimate goal of our research is to enable thousands of remote visitors to simultaneously and independently navigate a three-dimensional recreation of an important space. In this vision, each digital museum visitor will be able to explore a landmark or historic place at their own pace, navigating a virtual environment along their own path, examining what they find most interesting.

To illustrate the research problem, we first present a motivating example. We then discuss the major goals that an adequate solution must meet. Finally, we review the assumptions we've while formulating our solution.

3.1 Motivating Example

Suppose a team of digital museum employees is tasked with creating an online exhibit to feature the pyramids of Egypt. As part of the exhibit, the designers decide to allow users to navigate through a virtual pyramid. The team sets off for the pyramids with a digitization device for capturing SOI datasets. After digitizing the inside of a pyramid, the team uses a computer to process the data and prepare it for distribution via the digital museum's website.

After the exhibit goes online, thousands of visitors gather at the digital museum. They download a viewer that integrates with their web browser and start navigating through the digitized pyramid interior. Some users will connect via a cable modem, others through DSL, and yet others will connect via high-speed corporate networks. All users will see a realistic recreation of the interior, allowing them to explore and learn almost as if they were inside the pyramid themselves.

Unlike video, each user is able to navigate through the pyramid's interior space at their own speed and along their own path. The client software will adapt the incoming data stream to reflect the user's current position and interests. At the same time, adaptation will account for network conditions such as changes in bandwidth or latency in receiving the needed data, making the best use possible of the available network resources.

A number of problems must be solved before this vision can become a reality. These problems can generally be classified into three categories: (1) digitization, (2) distribution, and (3) reconstruction. In Section 2 we discussed some of the work addressing both the digitization and reconstruction portions of the problem. In this paper, we begin to explore the problem of distribution.

3.2 Goals

Before formulating a framework to address the delivery of SOI data, we developed a list of important goals. A successful solution to the distribution problem must have the following properties:

- **Compression:** Typical databases will contain several thousand, or more, high-resolution images. The raw data communication requirements can be enormous. A distribution framework must include a compressed data representation whose properties are congruent with the communication design.
- **Scalable performance for large user populations:** The user population can range from a single user to several thousand simultaneous users, each navigating independently. A distribution framework should only use mechanisms which scale well as the size of the user population increases.
- **Adaptation to client's application needs:** Each individual client is independent and may have unique application requirements. This can result in individual dataflow needs for each client. Individual adaptation must occur to maintain client independence.
- **Adaptation to client's available network resources:** Just as each individual client has unique application requirements, each client has individual network resource allocations. A successful distribution framework must provide the mechanisms for users to adapt their incoming dataflow to make the best use of whatever network resources are available.

It is important to note that some of these requirements conflict with others and that a careful balance must be maintained. For example, individual client adaptation makes scalable distribution difficult. A successful distribution framework needs to balance these competing interests and ensure that all requirements are adequately satisfied.

3.3 Assumptions

We make two simplifying assumptions in our work. These assumptions allow us to concentrate on the core problems of SOI data distribution. First, we assume that the bandwidth bottleneck is the *last mile*, the portion of the network closest to the end user. For most Internet users, this is indeed the case as network backbone capacity far exceeds endpoint capacity.

The second assumption is that the SOI server has a high-capacity connection to the Internet. Clearly, distribution of very large datasets to thousands of users will require a server with a significant amount of available bandwidth. Given an upper bound on the necessary bandwidth, we assume that a server can be provisioned to meet the required capacity.

4 Framework

This section outlines our framework to support scalable delivery of SOI datasets. There are three major portions of this framework: (1) representation, (2) transmission, and (3) client adaptation. In general, our framework is a channel-based transmission scheme. The data representation is distributed across a number of transmission channels. Clients retrieve data by subscribing to several channels at any given time. Each client adapts to current network and application conditions by managing their own set of subscribed channels.

Our channel-based design is prompted by the recognition that any per-client work done by the server inherently limits scalability. The overall performance costs for any per-client task will grow linearly with respect to the size of the user population. For this reason, our framework aims to remove all per-client tasks from the server. We make the server's role as simple as possible, pushing work away from the server and toward individual clients. This motivates a static but flexible data organization and a communication framework that supports client-initiated delivery and adaptation.

In this section, we first examine the representational requirements of our framework and the techniques we use to achieve these goals. Second, we discuss our proposed transmission mechanisms under both low and heavy usage conditions. Finally, we outline our approach to client adaptation, which allows users to individually control their incoming flow of data.

4.1 Representation

Our representation is designed to provide efficient data compression, allow multiresolutional access, and map easily to our transmission framework (see Section 4.2). This mapping allows clients to efficiently navigate through the dataset and adapt to local resource requirements in a scalable fashion.

The representation must also be aware that the communication framework, to improve system scalability, will use UDP, a network protocol that does not guarantee successful delivery of data. This factor must be taken into account in the representation design so that an individual packet loss will have only limited effect on the data stream.

Finally, the representation should recognize that the communication path is most likely to be the system bottleneck. Therefore, effort should be made to compress the SOI dataset, even at the expense of added decoding time.

For our application, the input data set consists of a set of cylindrical panoramic images. Each image has an associated position along a 2D plane at eye level. These images are encoded using a multi-stage process. The two major stages are segmentation and multiresolution encoding.

4.1.1 Segmentation

The first step is to begin segmenting the data based on three spatial dimensions. The data is first partitioned based on the $2D$ planar position of each image. Conceptually, we are creating this partitioning structure to allow a client to navigate the database by jumping from one partition to the next as the position of interest changes. Typically, a client will be interested in a neighborhood of partitions surrounding the viewpoint.

In our prototype, we use a regular grid partitioning algorithm as illustrated in Figure 2(b). However, variable resolution partitioning mechanisms can be used as well. For example, irregular partitioning schemes can be used to account for the non-uniform distribution of clients that might occur in a real-world application. Consider a virtual Louvre recreation. An irregularly high concentration of users will gather near the famous Mona Lisa painting. Smaller partitions near such high-demand locations would allow more fine-grained control for clients in popular regions.

Following partitioning, the segmentation process continues by splitting the dataset along view direction. Each image is captured as a 360° panorama. The images are broken up into four pieces, each with a 90° field-of-view. At this stage, we refer to each group of images as a *partition.* Each partition contains a collection of images, all oriented in the same view direction.

Partitions are further segmented along the dimension of spatial density, the first multiresolutional dimension. This step first requires that images be arranged by order of importance as measured by spatial density. We build a progressive ordering of images via quadtree decomposition, storing the image closest to the center of each region in the corresponding node of the quadtree. A breadth-first traversal of the quadtree determines the order. Figures 2(c) and 2(d) show this process. We refer to this order as the *streaming order.*

The partition is then segmented by grouping the images along the streaming order. This step creates segments of the database at various spatial resolutions. The split locations are chosen so that the low resolution data (i.e., sparse images) is contained in units of smaller size, while high resolution data (i.e., dense images) is contained in larger size units. This allows faster access to the sparse data which is of greater importance.

4.1.2 Multiresolution Encoding

Following segmentation, each individual image is encoded at multiple resolutions. We use the JPEG2000 image compression standard which uses a wavelet-based encoding process. Wavelet-based coders inherently encode an image at multiple resolutions. We split the progressive JPEG2000 codestream into individual parts, just as we did with the progressive streaming order.

Each image is encoded as either an index frame or a delta frame. Index frames are fully encoded as standard images. Delta frames are stored as differences from previously stored frames. For images with a closely matching predictor, delta encoding takes less storage space.

Images are encoded in streaming order. For each image, we first search for the best possible predictor. The search only examines images located earlier in the streaming order. If no good predictor is found, the image is encoded as a index frame. Otherwise, the difference is computed and encoded as a delta frame. The compressed images are split into multiple resolution layers as each image is encoded.

After all partitions have been encoded, the dataset can be viewed as a number of semi-independent units embedded within a $5D$ *channel space.* The channel space is defined by the two planar dimensions (x, y), an orientation dimension (θ), spatial density (δ), and image resolution (ρ). Each unit can therefore be specified by the quintuplet $(x, y, \theta, \rho, \delta)$.

While each partition is encoded independently, there are inter-dependencies between certain units. Within each partition, there is a two-dimensional collection of units corresponding to combinations of spatial density and image resolution.

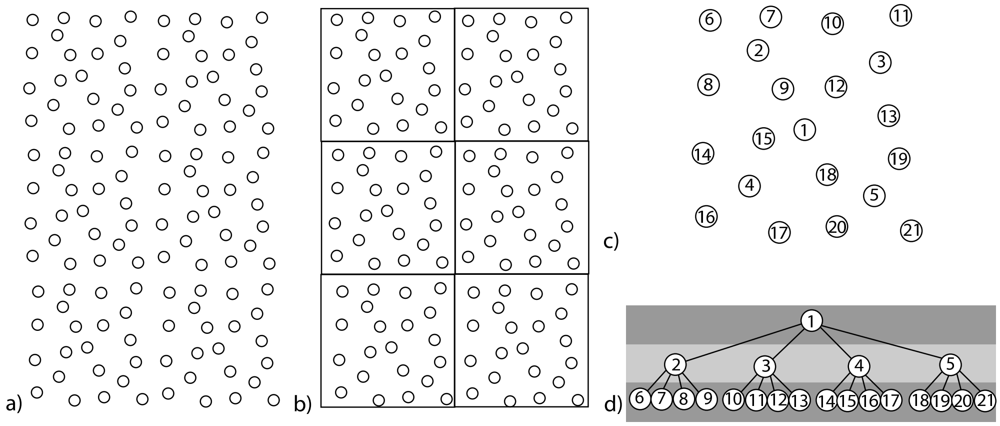

Fig. 2. *The dataset is organized across multiple dimensions. (a) The data consists of a collection of images arranged in a plane. (b) The first step in building the representation is to partition the data into spatial regions. This figure shows a regular grid partitioning. (c) The next step is to order the individual images by spatial density. (d) The ordering is determined by a breadth-first traversal of a quadtree built using the image positions in the plane.*

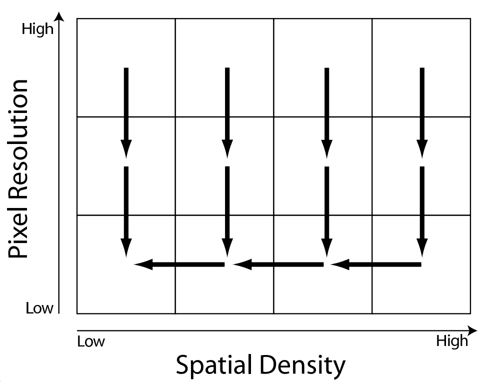

Fig. 3. *Individual units of the representation may depend on other units. In this example, higher pixel resolution units depend on lower pixel resolution units to form a chain from the highest to lowest resolutions. In the spatial dimension, dependencies exist only at the lowest pixel resolution. This dependency structure reflects the expected data access pattern for SOI.*

Dependencies between these units are required for our compression scheme. Unfortunately, dependencies can limit the ability of a client to utilize the tradeoffs provided by the multidimensional representation. To maintain flexibility, the dependencies are driven by the expected data access pattern and balance the needs for both flexible data access and effective compression rates.

It can be assumed that low-resolution data will be required before any high-resolution data. Following this assumption, higher-resolution layers rely on lower-resolution layers, and images from more dense units are encoded using predictors from more sparse units. A dependency diagram showing this relationship is shown in Figure 3.

As each unit is encoded, the compressed data is written to disk using both forward error correction (FEC) and application-level framing (ALF) [9]. Both techniques are used to counter the effects of lost packets. We must protect against loss because transmission will be based upon the unreliable UDP network protocol. FEC adds redundancy to the representation and allows for the recovery of information in the presence of low loss levels. ALF organizes data so that the damage caused by lost packets is as localized as possible.

4.2 Transmission

The second major component of our framework is the transmission of individualized data streams to the client pool. A suitable transmission mechanism must be scalable to support large user populations and provide each client with the ability to control its incoming data flow based on their own independent requirements. We accomplish this goal by proposing a novel channel-based communication design.

We support individualized client adaptation through channel management. Our design leverages techniques typically applied as bandwidth adaptation tools in static transmission systems with predictable user behavior, such as layered audio and video streaming. In our work, we extend these methods beyond simple bandwidth adaptation to support scalable adaptation to application-level content requirements such as a client's moving viewpoint in our SOI application. We discuss the process of adaptation in more detail in Section 4.3. In this section, we concentrate on the transmission mechanisms that make adaptation possible.

We begin by recognizing that our data representation can be easily mapped to a set of transmission channels. Each unit of encoded data is assigned to a unique channel. Each channel transmits the data assigned to it at a relatively low bitrate, allowing clients to receive multiple channels at the same time.

In our prototype, we transmit every channel at the same bitrate. However, there is no restriction that this be the case. The specific bitrate at which each channel transmits can be adjusted to reflect the popularity of the data contained in the channel. Transmitting popular data at a higher bitrate can support faster access to more popular regions of the digitized space. For example, this could be used to improve data access times for virtual Louvre visitors gathered in front of the Mona Lisa.

A client initiates a new session by contacting the server and obtaining a channel set description. The description specifies how data is mapped to the channel set, as well as any other global information needed by the client. The client then decides to which set of channels to listen based on the user's current viewpoint, motion, and local network conditions.

The number of channels that can be listened to at a given time depends on the bandwidth available to the individual client. As network conditions deteriorate, a client can reduce the number of channels until loss and latency characteristics improve. As network conditions improve, additional channels can be added to increase the reconstruction fidelity.

Our framework is designed to work with either unicast, broadcast, or multicast. The subscription metaphor is applicable to all three transmission methods, allowing the most appropriate to be used for a given user population.

4.2.1 Unicast

When the user population is relatively small, unicast is likely the best transmission method. In this case, users send subscription requests directly to the server. The server responds by transmitting the data associated with the requested channel via a TCP packet stream. The server transmits until either all data associated with the channel has been sent or an unsubscribe request is received. The server is responsible for handling subscribe and unsubscribe requests from all clients.

The unicast case is very simple and easy to implement. However, it does not perform well as the user population grows in size. Each client sends subscribe and unsubscribe requests to the server. Therefore, the management overhead grows linearly with respect to the population size. More importantly, the server-side bandwidth requirements grow linearly as well. The total bandwidth requirement is unbounded and, under heavy loads, can quickly lead to saturation of the server's available bandwidth. This point is discussed in more detail in Section 6.6.

4.2.2 Multicast and Broadcast

The communication framework is highly scalable when used with multicast or broadcast techniques. The server is no longer responsible for any per-client work. Instead, it simply transmits statically allocated streams. Each channel transmits data from a single unit of the representation via UDP. The data is transmitted over and over repeatedly. The total server bandwidth requirement is constant because there are a fixed number of streams transmitting at a fixed bitrate.

In the multicast case, the work of group management is handled by the multicast infrastructure and does not affect the server's load. In the broadcast scenario, there is no group management and the server's performance is not affected by the user population size. In either case, clients simply subscribe to a subset of channels and manage that set to control the incoming dataflow.

Both multicast and broadcast techniques rely on unreliable UDP transmission, so each stream should be encoded using FEC and ALF techniques. This will provide protection against low loss levels. When loss levels exceed the protected threshold, the client will be forced to continue listening to the channel until the lost data repeats. For this reason, data of high importance (i.e., low resolution) is stored in smaller sized units and repetition occurs with greater frequency.

Ideally, we would like to use multicast to transmit each channel only to interested parties. There are several competing multicast technologies and the amount of overhead from group management responsibilities varies greatly between them. On local area networks, broadcast can be used. One can imagine broadcast as the ideal multicast technology as far as group management is concerned. It transmits data to all participants, regardless of their interest. It therefore has no group management overhead and performs equally well with both small and large user populations.

4.3 Client Adaptation

Adaptation is an essential operation for clients. It is the mechanism through which clients can respond to application needs and changes in available resources. Through adaptation, clients control which portions of the dataset are received and at what rate the data arrives.

In our framework, clients adapt their incoming data flow by managing a working set of active channels. Each channel contains a portion of the dataset and is transmitted at a relatively low bitrate. Therefore, clients can control which parts of the dataset are received by subscribing and unsubscribing from individual channels.

The same subscription operations can be used to manage the rate of incoming data. A decrease in the incoming data rate can be achieved by unsubscribing from a channel and reducing the active channel set size. An increase in rate can be achieved by increasing the active channel set size.

4.3.1 Content Adaptation

At the start of a new session, clients obtain a description of the channel structure from the server. This description specifies the mapping between the data representation and the available communication channels. It also specifies the interdependencies between channels.

At run time, each client maintains a *point of interest* within the dataset based on the user's position in the virtual space. Given the point of interest, a rough prediction of the user's future position, and other application and network measurements, the client software decides which channels are most useful and subscribes to that set of channels. As the data associated with a specific channel is resolved, the client unsubscribes from the channel and subscribes to the best unresolved channel based on the current application conditions. We note that with both multicast and broadcast transmission, these subscription operations occur without ever directly contacting the server, allowing a single server to support a large population of simultaneous users.

The process of finding the best unresolved channel uses a utility metric that evaluates the usefulness of candidate streams. Our prototype uses a metric based on Euclidean distances within the $5D$ channel space. The utility metric also incorporates network and application conditions. Only channels with resolved dependencies are considered candidates for evaluation using the metric. Our general adaptation algorithms are more fully described in some of our other work [11].

Combined with our representation, this adaptation policy is an extremely powerful mechanism for data flow control. Through channel management, clients can intelligently allocate the available bandwidth for current application priorities. Clients can manage tradeoffs between any of the five dimensions that define the channel space: pixel resolution, image density, image position on the eye plane, and image orientation.

For example, clients alter their adaptation policy based on navigation speed. Fast moving clients must prefetch data from far ahead of the point of interest. Such a client will dedicate bandwidth to low resolution, low density channels located ahead of the point of interest. Slow moving clients can afford to prefetch less data from far ahead and can instead subscribe to channels containing more dense or higher resolution image information.

Subscription latency can be overcome by using these same content adaptation mechanisms. At high latencies, subscribe operations take longer to satisfy. Clients must therefore subscribe to data further ahead in the spatial dimensions at the expense of both density and resolution. At low latencies, adaptation can occur more quickly allowing more bandwidth to be allocated for high density and high resolution channels.

Our content adaptation mechanisms depend on the coherent data access pattern typical of navigational environments to intelligently prefetch data before it is needed. Therefore, users must move from one location to another by traveling through the environment much as a real visitor to a physical location would walk from one geographic spot to another.

Fig. 4. Each digital photo in the *Desk* dataset is a 360° field-of-view panoramic image with a resolution of 2048 × 512 pixels.

Random jumps within the space, such as teleportation from one spot to another, break the coherency of movement required for proper adaptation and make prefetching impossible. This implies that in systems that allow such teleportation events, drastic and abrupt changes in user viewpoint may require the system to pause as new data is loaded to reflect the new position.

4.3.2 Rate Adaptation

Throughout the session, an estimate of current network loss rates is maintained as a measure of network bandwidth conditions. Our work assumes that packet loss is caused by bandwidth saturation. We therefore decrease the channel set size when loss rates are above a threshold L_{shrink} for a longer than time T_{shrink}. Conversely, when loss rates remain below a threshold L_{grow} for longer than time T_{grow}, the channel set size grows.

Typically, T_{shrink} is significantly smaller than T_{grow} so that the system reacts quickly to excessive loss rates and grows slowly in low-loss situations. For reasons of stability, the value of L_{shrink} must be at least as large as L_{grow}. Without this restriction, it would be possible for same network conditions to simultaneously qualify for both an increase and decrease in channel set size.

Rate adaptation, as with content adaptation, is accomplished via channel subscriptions operations. Therefore there is no per-client work by the server when transmission is performed via multicast or broadcast and a single server can support a large population of users.

5 Prototype

We developed a prototype based on our framework to evaluate the effectiveness of our ideas and to guide future work. Our prototype enables remote visualization of SOI data over a wide range of network conditions.

Our server and client were both developed on Linux platforms using off-the-shelf personal computers. We used the *Desk* dataset for our experiments, provided to us courtesy of Lucent Technologies and the Sea of Images Project under Daniel Aliaga [2]. This dataset consists of 1,947 panoramic images. Each image has a full 360° field-of-view and a resolution of 2048 × 512. The raw size of this database is approximately six gigabytes. A sample image from the dataset is shown in Figure 4.

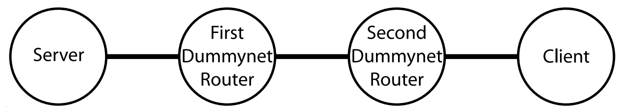

Fig. 5. Our experimental network consisted of four machines arranged in a simple dumbbell network. A server was connected to a client via two intermediate hosts. The intermediate machines acted as routers and ran the Dummynet network emulation software.

We ran our experiments on a simple dumbbell network as shown in Figure 5. Located between the client and server, two hosts running Dummynet [22] were used for network emulation. These hosts emulated a number of different network bandwidth and latency configurations during our evaluation.

6 Evaluation

We performed several experiments to evaluate the performance properties of our proposed framework. Using our prototype and the Dummynet network emulator, we tested system performance under a variety of network conditions. For all experiments, we emulated a single user navigating a predefined path through the Desk dataset described in Section 5.

Each experimental session lasted for 330 seconds and followed identical paths though the digitized environment. All experiments used a fixed channel bandwidth of 10,000 Bytes/sec. The values for T_{shrink}, L_{shrink}, T_{grow}, and L_{grow} were held constant across all experiments. All experiments used UDP transmission and emulated broadcast.

In this section we discuss the Area Factor metric, present results from our experiments, and draw some analytic conclusions regarding our framework's behavior.

6.1 The Area Factor

The Area Factor (AF) is a quality metric used in several of our experiments. The AF metric is derived from the SOI algorithm's use of image triplets for reconstruction. For any synthesized viewpoint, we can identify two triplets: the *ideal triplet* and the *available triplet*.

The ideal triplet consists of the three images closest to the viewpoint using the entire database as input. However, at runtime, the client does not typically have access to the entire database at once. Rather it has access to only the subset of images that have been retrieved to that point in time. We define the available triplet as the three images closest to the viewpoint using the only the available images as input.

We measure the quality of reconstruction by comparing the area of the triangles formed by the triplets

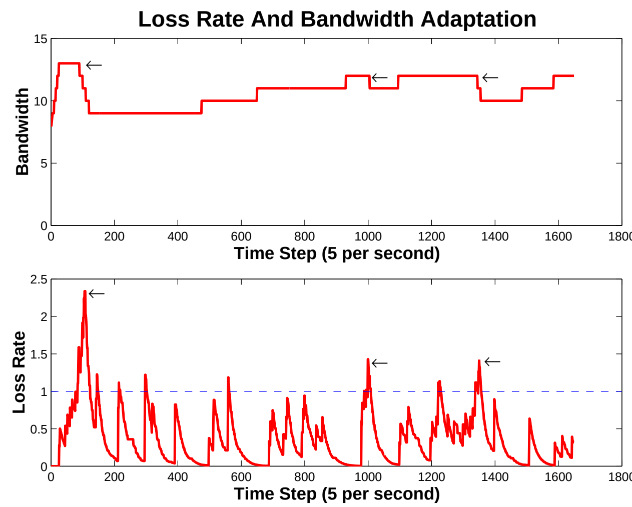

Fig. 6. The top graph shows the utilized bandwidth, measured in channels, over the course of a single session. The bottom graph shows the measured loss rate during the same session. The dotted line shows the value of L_{shrink}. After extended periods above this threshold, the channel set size is decreased. Three such events are illustrated by matching arrows in the two graphs.

of images. There are two triangles of interest when comparing the available triplet to the ideal triplet. We refer to these as $\triangle_{avail}$ and $\triangle_{ideal}$, respectively.

The AF value is computed by solving for the ratio between the areas of these triangles:

$$AF = \frac{Area(\triangle_{avail})}{Area(\triangle_{ideal})}$$

Given this formulation, the range of AF is restricted to $AF \geq 1$. Ideal reconstruction, where the areas of the two triangles are equal, is represented by $AF = 1$. Larger values of AF signify greater disparity between the ideal and available triangles, implying poorer reconstruction quality.

6.2 Adaptation to Available Bandwidth

An important design goal for our framework is the ability to adapt to a client's available network resources. Our framework is designed to support this goal by adjusting the size of the active channel set in response to measured loss rates under the assumption that loss rates are tied to network saturation.

To test our framework's adaptive performance, we measured both the size of the active channel set and the estimated loss rate for a complete session. During this session, we set the available bandwidth to 100,000 Bytes/sec.

The results are plotted in Figure 6. Given the per channel bandwidth of 10,000 Bytes/sec, client bandwidth hovered at about 10 channels. During the session, there were three loss events that caused decreases in the channel set size. A decrease is triggered by loss rates over

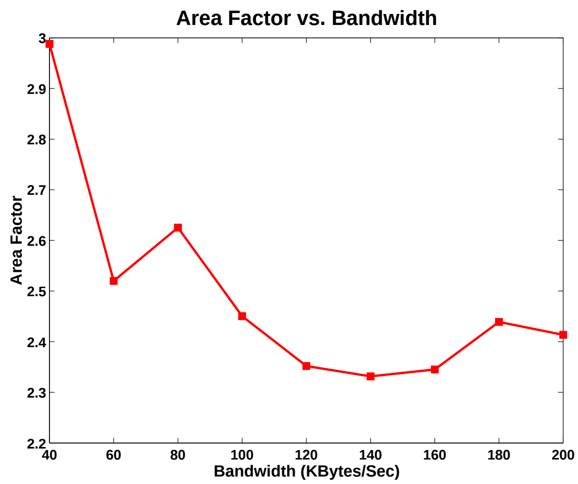

Fig. 7. The Area Factor (AF) metric is a measure of quality where lower values signify improved reconstruction. As the available bandwidth increases, the AF value drops. Low speed clients receive enough data to create low fidelity reconstructions, while higher speed clients use the additional bandwidth to improve quality.

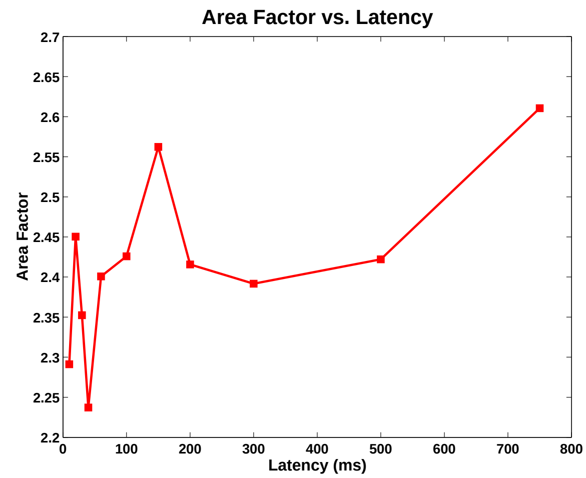

Fig. 8. The Area Factor (AF) trends upward as subscription latency grows, signifying a drop in reconstruction quality. Longer subscription latencies make adaptation harder by increasing the cost of subscribe requests. Longer latencies also require the client to look further ahead when evaluating the utility of each channel, negatively impacting AF.

L_{shrink} (shown by the dotted line) for a period of at least T_{shrink}. The three loss events are highlighted in the figure by arrows. Note that extended spikes in the loss rate were matched by drops in bandwidth. Following extended periods of relatively low loss, the network is probed for additional bandwidth by increasing the channel set size.

6.3 Quality and Bandwidth

The target user population is heterogeneous in nature. Not only will users navigate the digitized space independently, they will also have unique resource requirements. This is especially true for network connectivity. The user pool will contain both high and low bandwidth clients. Our framework must have the flexibility to supply high quality data over fast network connections while degrading gracefully for low speed connections.

We tested our framework's performance over a number of different bandwidth connections. We computed the average AF value for each session and plotted it against the available bandwidth. As bandwidth increases, AF decreases, signifying improved quality at higher network capacities. This trend is evident in Figure 7. The observed behavior shows that our framework behaves as desired, yielding improved quality for higher-speed connections.

6.4 Quality and Latency

Channel subscription latency is an important factor in end-user performance. This factor is especially important because it will be a key parameter in our future work when we begin experimentation using multicast.

We simulated various levels of subscription latency by adding a fixed delay to all subscribe requests. We varied this additional delay from between 20ms to 750ms. The delay values are in addition to any other network delays such as queue time. At each configuration, we computed the average AF for the session. The results are shown in Figure 8.

As the figure shows, AF trends upward as subscription latency grows, signifying a drop in reconstruction quality. There are two primary reasons for this change in quality. First, longer subscription latencies make adaptation harder by increasing the cost of subscribe requests. As a result, it takes longer for data to arrive following a request which directly impacts the AF value. Second, longer latencies require the client to look further ahead when evaluating the utility of each channel. Because a client knows that latency is high, it anticipates application needs for further in the future. This has a negative impact on quality.

6.5 Representation Design and Channel Size

There are several decisions that must be made when encoding the dataset and formulating the actual representation. Perhaps the biggest question is the degree to which data is partitioned. Recall that each unit of the dataset is mapped to an individual channel. Therefore, a highly partitioned dataset would allow relatively fine-grained access at the cost of high frequency channel switches. Alternatively, a minimally partitioned dataset would reduce the frequency of channel switches but would allow only course-grained access.

To measure the impact of partitioning on quality, we measured the average AF for several sessions with

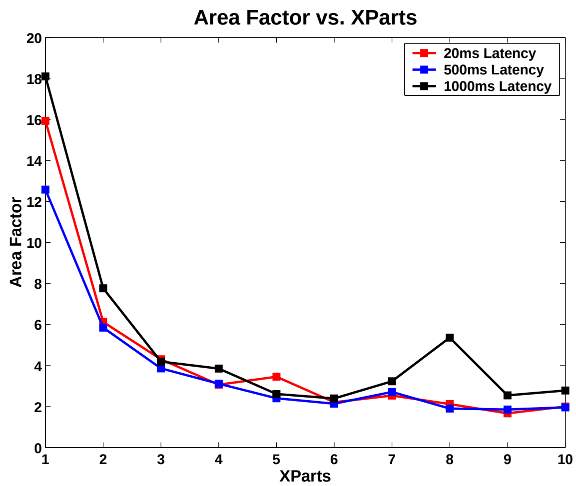

Fig. 9. The Area Factor (AF) is highest at low degrees of partitioning. As the dataset is partitioned into greater pieces, indicated by a larger X_{Parts} value, AF is improved. The improvement in quality for high X_{Parts} values is due to finer-grained access to the SOI dataset.

a variety of partition configurations. For each session, we changed the X_{Parts} value, which is the number of partitions in the X dimension of the capture plane. By adjusting this value, we can control the degree of partitioning that takes place in our regular grid partitioning process.

The results of this experiment are shown in Figure 9. As the graph shows, the area factor was worst with low values of X_{Parts} and gradually improved as the X_{Parts} value increased. This improvement was common across a variety of latency parameters. We caution that although the results indicate that additional partitions improves AF, this trend is not expected to continue indefinitely. At very high degrees of partitioning, we believe that the overhead costs of switching between channels will dominate the benefit of fine-grained access to data, leading to higher AF values.

6.6 Scaling Behavior for Large Numbers of Users

Our framework is designed to be scalable for large user populations. Unicast delivery, which replicates data for all interested participants, is not a reasonable mechanism for large user groups with high bandwidth demands. For this reason, our design is tailored toward multicast and broadcast delivery.

We have already shown how our framework, through channel set management, provides enough data flow control to support adaptation. We now argue that our framework, under both multicast and broadcast, is inherently scalable.

In the worst case, a server using our framework will be transmitting all channels at the same time. Our representation is static and has a fixed number of channels. In addition, our framework calls for a fixed bandwidth allocation to each channel. These properties define an upper bound on required bandwidth, regardless of the number of simultaneous users. In this scenario, the true limit on user population derives from the multicast group management mechanisms which become increasingly complex as user populations grow. Under the ideal of broadcast transmission, there is no fundamental limit to scale.

Conversely, under unicast delivery the required bandwidth grows linearly with the size of the user population. For this reason, a server will quickly become saturated as the user population grows. For any more than a few simultaneous users, the performance under unicast will be worse than under broadcast. The exact crossover point depends on the number of channels, the bandwidth allocated to each channel, the amount of bandwidth allocated to the server, and the average bandwidth demand of clients.

7 Conclusions and Future Work

We have presented a framework for scalable delivery of SOI datasets. This framework enables a large population of digital museum visitors to independently navigate a digitized environment. This new media type promises to provide a richer experience to museum users.

Our framework allows each user to maintain individual control over their incoming data stream. Each client uses channel subscription operations to manage a working set of active channels. In this way, clients can make the best use of available networking resources to meet their individual application needs.

There are two major contributions in our framework. First, we presented a multidimensional, multiresolutional representation for SOI data that allows access at a variety of fidelity levels. Second, we discussed a channel-based transmission design that is both scalable to large user populations and allows individualized control over a client's datastream through channel management.

The framework allows data transmission using unicast for low loads and broadcast/multicast for larger user populations. The framework is designed specifically to scale up to large numbers of clients. The effectiveness under ideal conditions was shown through experimentation and analysis.

While our initial results are promising, there are many avenues for future research. Most importantly, we must undertake deployment and measurement of the framework using real-world multicast protocols and measure the true impact of group management overheads. While we have roughly simulated this overhead in our subscription latency experiments, a more thorough investigation is needed. We would like to determine how closely multicast can come to matching the ideal performance of broadcast for digital museum environments. To this end, we have initiated a full evaluation of our latest prototype using Emulab [25]. This ongoing evaluation will provide a more thorough analysis of our approach under unicast, broadcast, and a variety of multicast infrastructures.

We must also address a number of application level problems. First, we must develop improved channel set management metrics. Perhaps the most fundamental evaluation in our system is the decision regarding which channel is best at any given point in time. While we implemented a rough heuristic for our prototype, a more scientific approach is needed.

Second, a more detailed analysis of the data representation design space is required. There are several decisions regarding partitioning, mapping data to channels, and inter-channel dependencies that must be more carefully explored.

Finally, and perhaps most importantly, the true value of delivering digitized spaces as part of a virtual museum can only be fully understood with the deployment of a fully functional prototype in a working virtual museum. Once the remaining roadblocks have been solved, thorough evaluation of a widely deployed prototype and a realistic, museum-quality dataset is essential.

References

1. E. Adelson and J. Bergen. The plenoptic function and the elements of early vision. In *Computational Modelsof Visual Processing*, pages 3–20. MIT Press, 1991.
2. Daniel G. Aliaga, Thomas Funkhouser, Dimah Yanovsky, and Ingrid Carlbom. Sea of images. In *Proc. of IEEE Visualization*, 2002.
3. S. Banerjee, B Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. of ACM Sigcomm*, 2002.
4. Shenchang Eric Chen. Quicktime VR - an image-based approach to virtual environment navigation. In *Proc. of ACM SIGGRAPH*, pages 29–38, 1995.
5. Y.-H. Chu, S. G. Rao, S. Seshan, and H.Zhang. Enabling conferencing applications on the internet using on overlay multicast architecture. In *Proc. of ACM SIGCOMM*, 2001.
6. Internet Pictures Corporation. Ipix. http://www.ipix.com/.
7. S. E. Deering. *Multicast Routing in a Datagram Internetwork*. Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, CA, 1991.
8. Nobel E-Museum. http://www.nobel.se/.
9. S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5:784–803, 1997.
10. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *Proc. of ACM SIGGRAPH*, pages 43–54, 1996.
11. David Gotz and Ketan Mayer-Patel. A General Framework for Multidimensional Adaptation. In *Proc. of ACM Multimedia*, 2004.
12. K. A. Hua and S. Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In *Proc. of SIGCOMM*, pages 89–100, 1997.
13. Smithsonian Institution. http://www.si.edu/.
14. Li-Shen Juhn and Li-Mend Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcasting*, 43(3):268–271, September 1997.
15. Stanford Computer Graphics Laboratory. Digital michelangelo kiosk. http://graphics.stanford.edu/projects/mich/kiosk/kiosk.html.
16. Marc Levoy and Pat Hanrahan. Light field rendering. In *Proc. of ACM SIGGRAPH*, pages 31–42, 1996.
17. Mark Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, Dave Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Funk. The digital michelangleo project: 3d scanning of large statues. In *Proc. of ACM SIGGRAPH*, pages 131–144, 2000.
18. L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Proc. of ACM SIGGRAPH*, pages 39–46, 1995.
19. Ackland Art Museum. http://www.ackland.org/.
20. Lars Nyland, Anselmo Lastra, David K. McAllister, Voicu Popescu, and Chris McCue. Capturing, processing and rendering real-world scenes. In *Videometrics and Optical Methods for 3D Shape Measurement, Electronic Imaging, Photonics West, SPIE Vol. 4309*, 2001.
21. University of North Carolina Graphics Laboratory. Virtual Monticello at the New Orleans Museum of Art. http://www.cs.unc.edu/ ibr/projects/NOMA/.
22. Luigi Rizzo. Dummynet: A Simple Approach to the Evaluation of Network Protocols. *ACM Computer Communication Review*, 27(1):31–41, 1997.
23. H. Shum and L. He. Rendering with concentric mosaics. *Proc. of ACM SIGGRAPH*, pages 299–306, 1999.
24. S. Viswanathan and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *Multimedia Systems*, 4:197–208, 1996.
25. Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, and Abhijeet Joglekar. An integrated experimental environment for distributed systems and networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255–270, Boston, MA, December 2002. USENIX Association.