

# A Collaborative Digital Pathology System for Multi-Touch Mobile and Desktop Computing Platforms

W. Jeong<sup>1</sup>, J. Schneider<sup>2</sup>, A. Hansen<sup>3</sup>, M. Lee<sup>4</sup>, S. G. Turney<sup>3</sup>, B. E. Faulkner-Jones<sup>5,6</sup>,  
J. Hecht<sup>5,6</sup>, R. Najarian<sup>5,6</sup>, E. Yee<sup>5,6</sup>, J. Lichtman<sup>3</sup>, and H. Pfister<sup>3</sup>

<sup>1</sup> Ulsan National Institute of Science and Technology, Korea

<sup>2</sup> King Abdullah University of Science and Technology, Saudi Arabia

<sup>3</sup> Harvard University, USA

<sup>4</sup> Inha University, Korea

<sup>5</sup> Beth Israel Deaconess Medical Center, USA

<sup>6</sup> Harvard Medical School, USA

---

## Abstract

*Collaborative slide image viewing systems are becoming increasingly important in pathology applications such as telepathology and E-learning. Despite rapid advances in computing and imaging technology, current digital pathology systems have limited performance with respect to remote viewing of whole slide images on desktop or mobile computing devices. In this paper we present a novel digital pathology client-server systems that supports collaborative viewing of multi-plane whole slide images over standard networks using multi-touch enabled clients. Our system is built upon a standard HTTP web server and a MySQL database to allow multiple clients to exchange image and metadata concurrently. We introduce a domain-specific image-stack compression method that leverages real-time hardware decoding on mobile devices. It adaptively encodes image stacks in a decorrelated color space to achieve extremely low bitrates (0.8 bpp) with very low loss of image quality. We evaluate the image quality of our compression method and the performance of our system for diagnosis with an in-depth user study.*

---

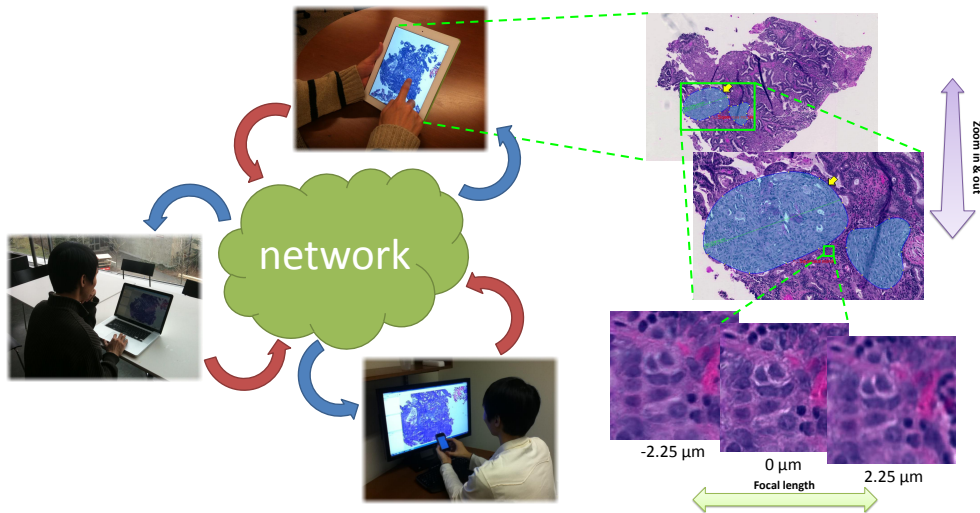
## 1. Introduction

Traditional anatomic or surgical pathology involves the review of thin tissue sections mounted on glass slides using a conventional light microscope. An experienced surgical pathologist views a large number of slides each working day, generated from a variety of different tissue and specimen types. Small biopsy-type specimens are generally represented on only a few slides (less than 10), whilst larger, complex multi-part excisions may generate tens to a hundred or more slides. In general, assigned cases are prioritized ("rush", biopsy or excision) and reviewed as they become available through the day. Additional slides or special stains are ordered as necessary, the findings integrated with those on the initial slides and a diagnosis rendered. In this way a large number of diagnostic decisions can be made quickly and efficiently enabling the pathologist to handle a large and varied caseload.

Typically, the majority of the case review is "solo", how-

ever, there are several common situations that require collaborative review. A general pathologist may need the opinion of a more experienced colleague or an expert opinion from a subspecialist. If both are at the same geographic location, the pathologists can view the slides simultaneously by using a double-headed microscope. One pathologist "drives" the slide on the microscope while the other views it, an intrinsically more passive experience. Consensus review of difficult cases by a pathology group, or group of subspecialists utilizes a multi-headed microscope for collaborative review. Finally, solo review followed by collaborative review with the attending pathologist is crucial for training of pathology residents and fellows.

Digital pathology has many potential advantages over the current manual processing, archiving and retrieval of glass slides. For example, slide delivery and subsequent archiving can be fast and simple and much less labor intensive. Slide images can be incorporated into the electronic medi-



**Figure 1:** Our collaborative digital pathology system. Mobile and desktop clients are connected through networks for collaborative remote diagnosis. Our client systems provide a multi-touch user interface for fast and intuitive view manipulation that mimics the viewing glass slides on a real microscope. Our novel image compression method leverages hardware decompression to allow fast switching of focal planes and zoom levels on mobile devices.

cal records for easy referencing. Removing geographic constraints allows greater access to subspecialty pathologists and a more efficient distribution of cases. Despite the apparent benefits, many pathologists are reluctant to transition to use of digital methods. To gain widespread acceptance, a digital pathology system needs to be as fast and efficient as the glass slide approach. Although whole slide images can be generated quickly (approximately 1 min per slide) using automated scanners from companies such as Aperio, Hamamatsu and Olympus, existing software solutions for viewing whole slide images have three significant drawbacks. First, reliable and fast access of the data for subsequent viewing and analysis is difficult, especially if the desire is to access the data from a remote site. Current systems are not optimized for remote collaborative viewing. Second, modern tools for visualization (e.g., rapid advance through focal planes or successive slide images) and measurement (e.g., length or area) cannot be brought to bear on the diagnosis challenge. Finally, annotations as a record of diagnosis or for the purposes of education cannot be affixed directly to the data, for example, to circle, highlight or indicate points of interest in the image. The overall goal of the work proposed here is to address these challenges.

Our approach has been to develop a server responsible for data and parameter storage and exchange while a client handles necessary computation locally using only a small subset of the data. The server manages large image data efficiently using a database and quickly provides small subsets of the data requested by the client. The server also stores all the non-image *meta-data* that are generated during the diagnosis process, including session and user identifiers, image opera-

tors, and annotations, and coordinates clients to share this information with each other. We used a standard HTTP-based server and a MySQL database to implement a scalable system while reducing development effort. The client has a dedicated image viewer that can display extremely large image data efficiently. The core idea is to reduce the data transfer overhead by using a novel data compression method and to achieve interactive performance by processing only visible data using on-the-fly hardware decompression. Our system supports multi-touch enabled mobile client devices, namely the iPad and the iPhone, and high-performance workstations equipped with fast graphics processors (GPUs) (Figure 1).

The primary contributions of our work are twofold. First we introduce a novel digital pathology system based on a client-server model that supports remote viewing of three-dimensional whole slide images at interactive rates on mobile client platforms. To the best of our knowledge this system has the highest performance available and is the only one designed for remote collaborative viewing on touch-enabled mobile computing devices. Second we propose a novel image compression method that leverages hardware decompression on mobile devices for faster decoding while achieving bit rates as low as 0.833 bits per pixel (bpp) with minimal loss of image quality. We evaluate our digital pathology system and image compression method with a user study to demonstrate their effectiveness.

## 2. Related Work

The success of any digital pathology system crucially depends on two factors: a database able to serve requests within



**Figure 2:** Common practice for collaborative diagnosis using a dual view microscope. In this workflow, one person drives the microscope while the other simultaneously watches through separate microscope oculars. Other versions of multiheaded microscopes have three or more oculars.

short time and a client to provide domain experts with a fast visual stream of information. Especially the client determines power and flexibility of the entire system. The use of web-based clients [NYU, NDP] has the advantage of working in any web browser (and therefore on any recent mobile device), but the viewer's function is limited to simple image display. For instance, advanced functions are not generally supported, such as tracking another user's view in real-time or fast switching of focal planes and zoom levels. A competing approach is to utilize a locally stored database [NDP, Sco] to offer high performance, but changing focal and zoom levels rapidly still results in large bandwidth requirements and memory footprints. It is therefore not fully solved satisfactorily. Jeong et al. [JST\*10] address this problem by compressing multiple image slices jointly. Using GPU-based decoding, their system allows fast switching of focal planes and adjustment of zoom levels. However, the latter method does not support efficient remote collaboration using mobile devices nor does it provide multitouch user interface for easier navigation.

Remote visualization systems [Bet00] can be roughly classified into *render-local* (the server transfers raw data to a client for rendering and display) and *render-remote* systems (the server forms the final image and sends it to the client for display). Since render-local systems are prone to becoming infeasible in case of data so large that it exceeds both bandwidth and local hardware limitations, render-remote systems (e.g., [EE99, MC00, SME02]) have been more popular. However, render-remote systems typically require significant investments in terms of server hardware while the hardware of the client remains largely idle. Therefore, a third class of systems uses a *shared rendering* approach (the server renders an intermediary format and sends this to the client for final compositing). Shared rendering systems typically utilize graphics hardware acceleration on the client to ensure interactive performance [BSL\*00, EEHT00, LP03].

While commercial and open source systems [IBM05, App, Gol, HRC\*06, Par] implementing these paradigms are readily available, the size and complexity of these systems is a barrier to entry for new users and pathologists. Furthermore, these systems are not optimized for interactive visualization, they do not fully support mobile clients, and they are completely agnostic of the particular task at hand (i.e., whole slide digital pathology image stacks).

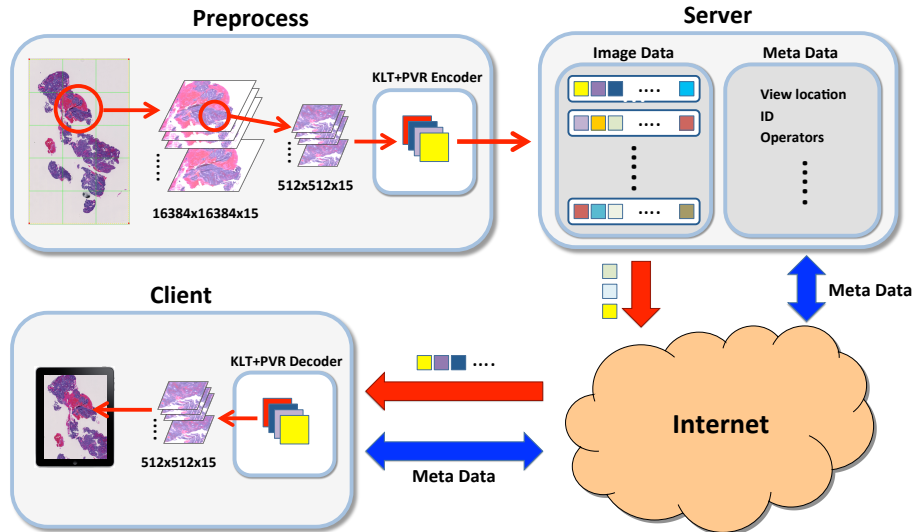
Our system classifies as a render-local system, but we overcome bandwidth limitations using a domain-dependent data compression scheme which can be decoded by mobile client GPUs. The use of domain-specific knowledge about the optical microscopy data used in digital histopathology has received comparably little attention. The prevalent approach is to encode images separately using JPEG or vector quantization [NH92, GG91] on Laplace pyramids [BA83, GY95]. Avinash [Avi95] uses JPEG compression on 8 bpp images with an adaptive quality heuristic to achieve compression ratios between 2:1 to 11:1 at an PSNR of 21.14dB to 48.13dB. Schneider et al. [SW03] use a vector quantizer to compress a 3-level Laplace pyramid encoding 3D volumes. As one application, they compress RGBA confocal microscopy stacks at a ratio of 31.2:1, but the distortion is not reported. Similarly, Cockshott et al. [CTG\*03] use four to five levels of a Laplace pyramid and vector quantization to encode the residuals, resulting in PSNRs of 30dB to 44dB at a compression ratio between 15:1 and 20:1. Jeong et al. [JST\*10] use a hierarchical vector quantization scheme with a linear predictor between slices. They report an SNR of 24.98dB at 0.88bpp and of 26.50dB at a compression ratio of 20:1; the PSNR is not reported. In their approach, the compression ratio is varied per image stack based on a quality threshold, and the format can be decoded on the GPU.

### 3. System Design

#### 3.1. Design Goals

The main design goal of our system is to build a scalable remote visualization system that can host large-scale datasets on a central server and handle multiple clients' request for random access to the data in parallel. We offload the computational burden from the server and let each client perform necessary computation locally to create the final image on the screen. In addition, the system should be able to run interactively so that it mimics the experience of driving a real microscope. Our client viewer can quickly decompress the multiple image planes of a three-dimensional tile allowing rapid change in the image plane without noticeable lag. The speed with which users can advance through image planes is comparable to that of changing focus on a microscope. This feature is one of the major differences between our system and existing digital pathology systems that treat each focal plane as a separate image.

In order to achieve these goals, we designed the system



**Figure 3:** Overview of our collaborative client-server digital pathology system. Input focal stacks are diced into fixed-size tiles, and each tile is compressed independently using our domain-specific compression method. The server stores compressed image data and meta data. The client requests tiles for visible regions from the server along with meta data. The final image on the client is generated by decoding compressed tiles on-the-fly. Image data communication (red arrow) is unidirectional, i.e., from the server to a client, whereas meta data communication (blue arrow) is bidirectional.

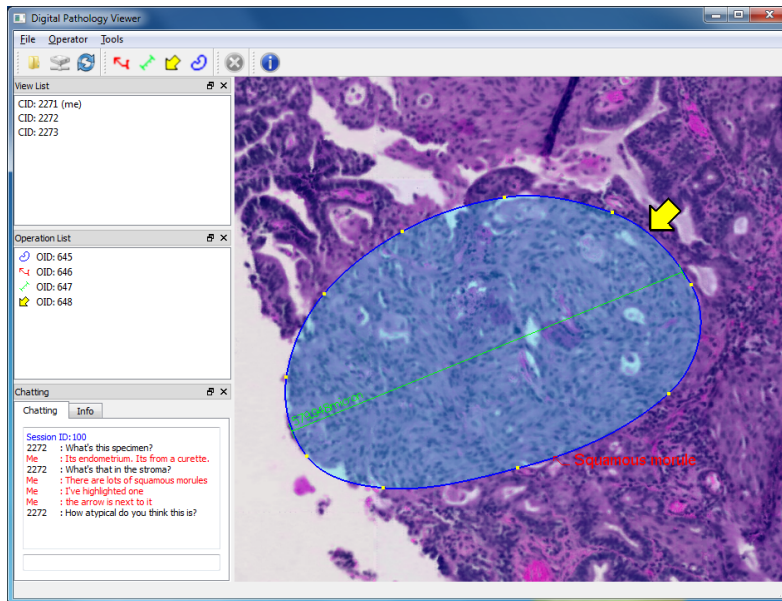
so that the single compressed data format can be seamlessly shared across heterogeneous devices over the network, such as mobile devices and high-end desktop PCs. Our approach is using an existing hardware compression format that can be quickly decoded on any client platforms. However, a naive application of an existing hardware texture compression method does not sufficiently reduce the large data size of the pathology image stacks. Therefore, we developed a novel domain-specific compression scheme based on an existing hardware texture compression format so that the compression ratio is higher than that of a native hardware texture compression format while the decoding can be done efficiently using the hardware. By doing this, we can minimize the network latency while achieving the interactive render-local visualization performance and providing compatibility of the data format across heterogeneous client systems.

Another design goal is to provide a mechanism for multiple users to *collaborate* easily. Traditional multiheaded microscopes, telepathology video systems and even many distributed whole slide image visualization systems only allow one person to drive the slide or image navigation at a time. The current driver's view is passively received by the others (Figure 2). To provide a more flexible *synchronous and distributed* collaborative visualization system [?], we designed our system to allow each user to follow any other's current view and vice versa. In addition a user can add annotation to any image they are driving (i.e., but not one they are following). The annotation becomes immediately visible to all followers. For each client the meta data for annotations and the current visualization state are exchanged with

other clients concurrently. With different users being able to drive the same image but in separate views, the potential for conflict with respect to who is driving any one image is eliminated. The technical challenges in our system are to maintain an interactive frame rate for driving and following image navigation while allowing sharing of multiple views and annotation concurrently across various client systems. Our hardware-accelerated render-local collaborative visualization system elegantly manages these issues.

### 3.2. System Overview

Figure 3 shows an overview of our system. We start with a set of images that covers the area of the tissue sample to examine. A conventional microscope equipped with a motorized stage can produce multiple overlapping images from a glass slide, which can be directly used as the input to our system. Instead, we use an Olympus Nanozoomer, which is an automated slide scanning device that produces a single stitched image per a glass slide. We dice each image from the Nanozoomer into fixed-sized subimages of  $16384 \times 16384$  pixels for easier storage and processing. We scan 15 images per glass slide at different focal depths, with a distance between adjacent focal planes of 0.75 microns. We call this group of 15 images a *focal stack*. Each focal stack is then processed and stored independently. An image pyramid is built per focal stack, and each level in the pyramid is diced into  $512 \times 512 \times 15$  fixed-sized tiles, similar to Jeong et al. [JST\*10]. Finally, each tile is compressed individually (Section 4), and the tiles from the same focal stack are grouped and stored together in a single file for faster access.



**Figure 4:** Desktop client showing (left, top to bottom) client list, annotation operators, and chat session. An image region has been marked using annotation operators: a yellow arrow, red text, green ruler, and blue segmentation curve.

Because each tile is compressed independently, random access to arbitrary locations in the image pyramid can be done efficiently. When the visible region of the image is determined by the client, tiles overlapping that region are loaded either from the server or from the local disk. Then the client decompresses the tiles on-the-fly using the graphics hardware accelerated decoder and displays them on the screen.

### 3.3. Server

The server's main roles are twofold – to serve image data and to exchange meta data with the clients. The server stores all the image data and returns the compressed stream of data for the requested tile. Meta data are all the non-image data that is stored on the server and shared among the clients. We distinguish between two categories: *Server-generated meta data* is essential information to coordinate communication between clients, such as session and user ID, data and header names, etc. *Client-generated meta data* is additional information created by each client during the session that will be shared with the other clients, such as client view location, annotations, text data from the chat sessions between remote users, etc. Our experiment shows that our server can easily handle concurrent tile requests and effectively increases the data transfer rate by hiding the network latency (see Table 1 in Section 5). The server uses a MySQL database to store meta data in three tables, and entries in the tables are returned in XML format messages to clients.

### 3.4. Clients

The clients are built upon a demand-driven large-scale image viewer framework [JST\*10] for efficient data management and hiding disk/network latencies. Our client platforms support OpenGL and GLSL shaders. Each client receives the images compressed with the same format, and it decompresses the focal stacks using hardware acceleration. We currently implemented clients for two widely available platforms: a mobile client for the Apple iPad and iPhone/iPod Touch, and a desktop client for Windows PCs with GPUs and OpenGL acceleration.

**Client Functions:** The clients provide basic functions, such as opening data files, selecting/creating sessions, and changing views. The user can either join an existing session or create a new session. The client list shows currently active clients for the current session, and the user can switch to any other client's view by selecting the client ID. Selecting the user ID in the list returns the client to the previous view. This allows multi-way view manipulation by multiple users, which is typically not possible in traditional collaboration systems. To follow the other client's view smoothly, the client polls the server frequently, about once every 100 milliseconds, and refreshes the screen accordingly. For ease of remote collaboration, a chat function is implemented so that the users can exchange messages while using the system (Figure 4 left).

The client system provides several annotation operators that are essential for diagnosis. The *text* operator allows the user to write text on the image. The *arrow* operator is used to mark a specific location with an arrow-shaped marker. The

*ruler* measures the length between two user-defined points using the actual pixel-size information from the microscope. The segmentation operator is used to draw closed curves to mark the region for segmentation. Figure 4 shows an example of each operator in a desktop client window.

**Implementation:** The iPad client is a native iOS application written in Objective-C. It uses OpenGL ES 2.0 for rendering. The client has three threads: the main UI thread, the tile loading thread, and the prefetching thread. The tile loading thread downloads the tile if necessary and then loads it to memory. The prefetch thread prefetches two rows/columns around the current view area and one level up and down in the focal stack pyramid. Once finished prefetching this region, it sleeps until the main thread signals that the viewpoint has changed and prefetching continue. The tile loader thread is signaled that there are new tiles to load and it begins loading them, prioritizing tiles currently shown on the screen over the tiles not visible but prefetched. The main thread renders the scene and controls the UI.

The multi-touch interaction in the iPad and iPhone/iPod Touch is well suited to support image navigation in digital pathology. We employed a two-finger pinch-and-zoom gesture for controlling image magnification (zoom), and a one-finger swipe gesture for moving the slide (pan). Other tasks are used less frequently and, so, are implemented using a menu that is revealed or hidden by tapping the surface with a single finger. This menu includes items for changing the focal plane, starting or joining collaborative sessions (changing the displayed image), markup and annotation.

The desktop client is implemented as an Win32 application written in C++, OpenGL, and Qt. The basic design for the viewer is similar to that of the mobile version, but there are several differences. First, desktop GPUs do not natively support PowerVR texture compression (PVRTC), so we have emulated this functionality using our own shader-based decoder (see Section 4.2). Second, the desktop client's GPU and main memory size are usually bigger than those of mobile devices, so we can prefetch much larger neighbor regions into the cache to hide network latency more effectively. Third, we implemented the multi-touch user interface for our desktop client using a wireless touch device. We tested several off-the-shelf touchpads for Windows, but they did not fully meet our expectations. Instead, we developed an iPod touch/iPhone application that remotely controls the desktop client using multitouch gestures. The application uses the TCP/IP protocol and ad-hoc wireless networking to communicate with the client. We implemented the same multitouch gestures used for our iOS clients so that the user can use similar interactions for image navigation.

#### 4. Compression

Our compression method is a novel combination of hardware and software techniques. Focal stacks are compressed

based on the PowerVR texture compression (PVRTC) method [Fen03], a block-based compression method natively supported by the PowerVR GPU found in numerous mobile devices. However, a naïve application of PVRTC only gives 2bpp compression ratio at best, which is not sufficient for our use. Therefore, we employ Karhunen-Loève Transform (KLT) [Kar47, Loé78] and adaptive encoding to leverage domain-specific image properties, which results in extremely low bit rate (0.8bpp) while providing a fast hardware decoding option on mobile GPUs.

**PVRTC overview:** The underlying idea of PVRTC is to store two downsampled images, each containing one color sample per  $8 \times 4$  image block (see also Figure 5). The decoder first performs a bilinear upscaling of these two images followed by a per-pixel linear interpolation between these intermediate image according to modulation weights. Per block, 2bpp mode PVRTC stores two 15bit colors, 2bit flag information, and 32bit modulation information (a 4bpp mode is supported as well but not discussed here). One of these flag selects one of two modulation encoding schemes. The first scheme stores one of the values  $\{0, 1\}$  per pixel, while the second stores one of the values  $\{0, 0.375, 0.625, 1\}$  for every other pixel in a checkerboard pattern. Missing modulation weights are averaged from their neighbors. This second modulation mode offers a better control over the modulation than the first mode, albeit at a lower spatial resolution. The other flag bit is not used in our work.

#### 4.1. Encoding

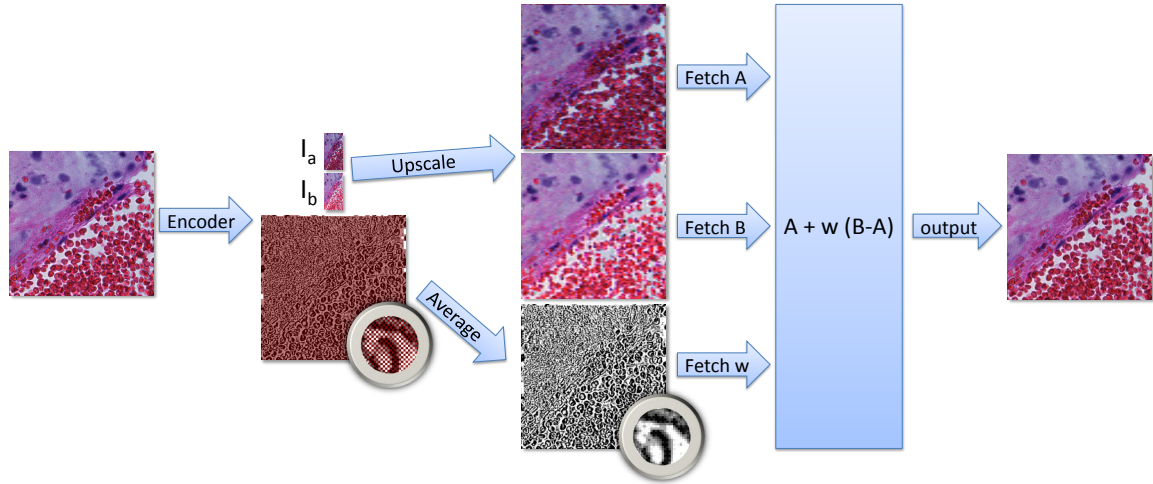
The data we received from our domain experts exhibits a close-to-planar color space, since it is stained using hematoxylin and eosin. While this need not be the case for other data sets, we first describe the treatment of close-to-planar color spaces for the sake of simplicity and generalize the method to other spaces later in this section. From a bird's view, our encoding scheme consists of color space decorrelation, chroma downsampling, and joint-PVRTC-encoding of adjacent focal planes.

**Color Space Decorrelation.** To exploit close-to-planar color spaces, we use a KLT to automatically detect and decorrelate this color space. The KLT is a linear and orthogonal transform that rotates RGB vectors into a new space, referred to as KLT-space, with components  $\alpha\beta\gamma$ . We start by computing the KLT matrix for each stack. Let  $rgb_i$  be the  $i^{th}$  color vector of the stack. We first compute the average color and covariance matrix of the stack:

$$a = \frac{1}{N} \sum_{i=1}^N rgb_i \quad (1)$$

$$C = \sum_{i=1}^N (rgb_i - a)(rgb_i - a)^T. \quad (2)$$

The covariance  $C$  is a symmetric real  $3 \times 3$  matrix. We therefore proceed by computing the eigendecomposition using



**Figure 5:** Conceptual overview of PVRTC. Two downscaled images  $I_a, I_b$  plus modulation weights are stored. Modulation weights may be specified for each other pixel; missing weights are averaged from neighbors. The decoder first upscales  $I_a, I_b$ , then averages missing weights. A linear interpolation between the upscaled  $I_a, I_b$  yields the output image.

the QL algorithm with implicit shifts [PTVF02]. Finally, we order the eigenvalues  $\lambda_i$  by absolute magnitude to obtain:

$$C = R \text{diag}(\lambda_1, \lambda_2, \lambda_3) R^T, \quad |\lambda_1| \leq |\lambda_2| \leq |\lambda_3|. \quad (3)$$

The eigenbasis  $R$  rotates RGB vectors to KLT-space. Since we are dealing with discrete 8-bit color values  $[0, \dots, 255]$ , we scale the rows of  $R$  and find a bias-vector  $B$  such that each component of the resulting  $\alpha\beta\gamma$  vectors is in the range  $[0, \dots, 255]$ . This is important to avoid excessive truncation errors in subsequent steps. The full transform can then be written as follows.

$$\alpha\beta\gamma_i = R \text{rgb}_i + B \quad (4)$$

The KLT has optimal decorrelation properties [GZV00] but requires  $R$  and  $B$  to be stored explicitly for each focal stack. Decorrelation of the color space is achieved by *maximizing* the variance of the  $\alpha$  component and *minimizing* the variance of the  $\gamma$  component. Since the color space in our application (and as attested by our experiments) is close to planar, the variance of  $\gamma$  will be close to zero. We therefore proceed *adaptive encoding* by computing a single, average  $\gamma$  value per stack and store only this value.

**Chroma Downsampling.** Similarly to the closest-to-constant  $\gamma$ -value, the variance of the  $\beta$  component is sufficiently small (and in our experiments also smooth enough) to store  $\beta$  at a reduced resolution. This is akin to chroma subsampling in image coding. Specifically, we use a variant of a centered 4:2:0 subsampling [Ker09], i.e. a 2 : 1 subsampling along both the  $X$  and  $Y$  axes of the stack. This low resolution image  $\beta$  is then bilinearly upsampled during decoding.

The quality of the decoded image will crucially depend on the quality of the downsampling of the  $\beta$  channel. Therefore, we devise a novel  $L_2$ -optimal downsampling method

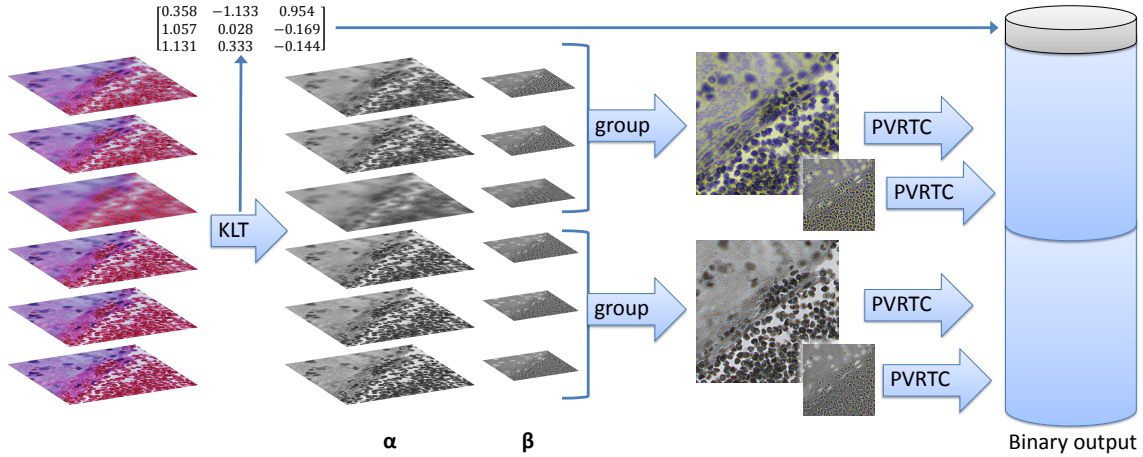
rather than using the box filter frequently used for mipmapping (also refer to the Appendix). We formulate the downsampling operator as a least-squares pseudo-inverse of the bilinear upscale operator. Solving this least-squares problem yields the following (separable and symmetric) convolution filter for the 1D case:

$$I' = \downarrow_2 * 2 \left[ \dots, 0, \frac{1}{3^3}, 0, -\frac{1}{3^2}, 0, \frac{1}{3}, \frac{1}{3}, 0, -\frac{1}{3^2}, 0, \frac{1}{3^3}, 0, \dots \right] * I. \quad (5)$$

Here,  $I$  denotes the discrete input signal (a row or column of pixels) and  $\downarrow_2$  denotes 2 : 1 subsampling (comb filter). Observing that the left half of the filter operates only on odd pixel positions while the right half operates on even pixel positions, this infinite impulse response (IIR) kernel can be implemented efficiently by splitting it into the sum of two IIR-convolutions, one from the left and one from the right. By exploiting recurrence in the kernel—each filter coefficient is obtained by scaling the previous non-zero coefficient with  $-\frac{1}{3}$ , this downsampling scheme can be implemented with linear complexity and only a single register.

At the image boundary, we implement *clamp-to-edge* boundary conditions by summing over the filter coefficients outside the image and using them as coefficients for the boundary pixel. This results in a simple change; namely the boundary pixel is weighted by  $\frac{1}{2}$  instead of  $\frac{2}{3}$ .

We acquire the 2D filter as the tensor product of two 1D filters. Our tests indicate that this novel  $L_2$ -optimal downsampling method improves the signal-to-noise ratio between the low-resolution, bilinearly upsampled image and the original by at least 1.4 dB and up to 2.3 dB. The resulting sampling positions in the downsampled image lie between the original pixel positions and agree with



**Figure 6:** Overview of our encoding pipeline. We first perform the conversion to KLT-space, then group 3 full-resolution  $\alpha$ - and 3 half-resolution  $\beta$ -channels into 3D pixel vectors. These intermediate images are encoded using PVRTC and comprise, together with the inverse KLT matrix, our binary output.

hardware-supported filtering.

**Joint-PVRTC-Encoding.** To exploit the high correlation between slices in a focal stack (also see Figure 8), we encode three adjacent  $N \times N$  pixel slices in two 2 bpp PVR textures: one  $N \times N$  texture for three  $\alpha$  components, and a  $(N/2) \times (N/2)$  texture for three  $\beta$  components. Furthermore, we combine the average  $\gamma$  value and the bias vector  $B$  in order to only store a single  $3 \times 3$  matrix for color-space conversion (also see Figure 6).

Overall, we store  $3 \times N \times N$  24-bit RGB pixels in  $N \times N \times 2$  bits ( $\alpha$  image) plus  $(N \times N \times 2)/4$  bits ( $\beta$  image). Neglecting the overhead for the  $3 \times 3$  matrix, we thus obtain a bit rate of  $(2.5 \times N^2) / (3 \times N^2) = 0.833 \dots$  bits per pixel.

**Non-Planar Colorspaces.** It is intuitively clear that for non-planar colorspaces an additional third component  $\gamma$  has to be stored. From the eigendecomposition of the covariance matrix  $C$ , we use a planarity measure

$$p(C) = 1 - \frac{3|\lambda_3|}{|\lambda_1| + |\lambda_2| + |\lambda_3|} \quad (6)$$

to determine planarity.  $p(C)$  is normalized to be 1 for perfectly planar (or linear) colorspaces while it will be 0 for colorspaces that evenly comprise all three dimensions. If  $p(C)$  is below a threshold (0.85 in our current implementation), we add a spatially resolved third layer of textures downsampled by a factor of 2 along each axis. In this case, our compression rate increases to 1bpp and we need to store a  $3 \times 4$  colorspace conversion matrix. See also Figure 8 for planarity measurements of representative data sets.

## 4.2. Decoding

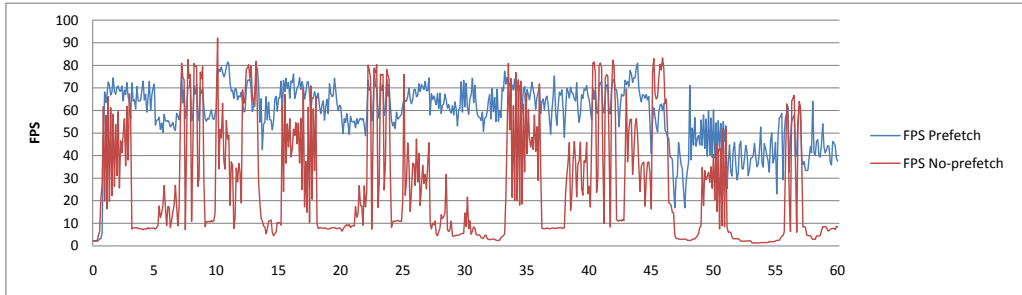
If PVRTC decoding is supported in hardware, decoding consists only of bilinearly sampling the  $\alpha$  and  $\beta$  textures, selecting the component corresponding to the current slice (due to encoding 3 slices in three channels) followed by color-space conversion in the shader. Since the color space conversion is linear, this yields the correctly interpolated value.

Because PVRTC is not supported natively by desktop GPUs, we decode it in a GLSL shader in three passes. The binary representation of the compressed focal stack is stored in an OpenGL Texture Buffer Object to avoid limitations of available texture formats. We first reconstruct the two low frequency images  $I_a$  and  $I_b$  in a single render pass using multiple render targets (MRTs). These two images are  $(N/8) \times (N/4)$  RGB images, where  $N \times N$  is the resolution of our input stack in the  $X, Y$ -directions. Then, we decode the modulation weights as outlined above into an  $N \times N$  render target  $M$ . This additional step allows us to fill in missing interpolation values by averaging their neighbors (checkerboard pattern mode) without having to reconstruct the neighbors multiple times. In the third pass, we bilinearly upsample images  $I_a$  and  $I_b$  by binding them as textures, we fetch a modulation weight from  $M$ , and, in case of the checkerboard pattern layout, average missing modulations from their four immediate neighbors.

## 4.3. Treatment of Image Boundaries

One minor drawback of PVRTC is that only power-of-two texture resolutions are supported. Furthermore, all images are assumed to be tiling. The latter limitations can clearly lead to problems in our scenario. Therefore, we pad each stack with information of neighboring stacks in both  $X$  and  $Y$  direction. This effectively reduces our image resolution





**Figure 7:** Rendering performance with and without prefetching. Frames per second over time in seconds.

per stack by 8 pixels in  $X$  and  $Y$  direction, but the rest of our method remains unchanged.

## 5. Results

We measured the performance of our system on a Windows desktop client equipped with 2.66 GHz Intel i7 CPU, 12 GB RAM, NVIDIA GTX 480 GPU with 1.5 GB VRAM, and an Apple iPad 2 mobile client with 32 GB flash memory. The server was a virtual machine running CentOS 5 Linux on a 2.93 Ghz x86\_64 CPU with 2 GB RAM. Desktop and mobile clients communicate with the server either through wired or wireless network.

### 5.1. Client-Server Performance

Decoding a  $512 \times 512$  image tile on a desktop PC using our PVRTC decoder takes 0.6ms including CPU to GPU memory transfer time. We achieve more than 55 fps rendering to a full HD ( $1920 \times 1080$ ) screen if all tiles are in memory. This result was measured with 30 tiles covering the entire screen and while constantly changing the focal plane. On the iPad, decoding the same size of tiles takes 0.87ms using hardware PVRTC and our shader-based colorspace conversion. We achieve 47 fps to update the  $1024 \times 768$  pixels iPad screen using 24 tiles in the worst case, i.e., tile size is about half of its original size due to zoom out.

Since our system is build upon a standard web server, many parallel requests from clients can be efficiently handled by the server. The major bottleneck is slow network transmission, so our clients request multiple tiles concurrently using parallel threads to hide network latency. Table 1 shows server-to-client data transmission rates for various numbers of simultaneous tile requests. As shown in this table, simultaneous tile requests increase the data throughput up to four times. We empirically found that around 20 concurrent tile requests can be used to achieve maximum average data rates for desktop clients.

To assess the performance of our viewer for a realistic application scenario, we measured the total rendering time including server-to-client data transfer time. In this experiment, the client continuously changes the viewpoint, fo-













**Table 1:** Server-to-client data transfer rates (in MB/s) for multi-threaded tile fetching. The maximum average data rate is achieved for 20 threads.

|     | Number of Threads |       |       |       |       |
|-----|-------------------|-------|-------|-------|-------|
|     | 1                 | 3     | 5     | 10    | 20    |
| Ave | 20.70             | 49.17 | 68.07 | 76.58 | 84.08 |
| Min | 18.68             | 41.82 | 59.08 | 68.46 | 76.25 |
| Max | 22.19             | 55.61 | 73.88 | 84.46 | 87.57 |

cal plane, and zoom level by following a pre-defined path. We changed the viewpoint slowly and in a continuous manner without abruptly jumping from one location to another, mimicking the movements of a typical user. As shown in Figure 7 (blue), our system can handle these smooth movements at interactive frame rates. The number of tiles requested from the server each frame is small because the neighboring tiles are constantly loaded in the background by the prefetching thread. Without prefetching, we observe a significant drop in frame rates as shown in Figure 7 (red).

### 5.2. Compression

Our compression fidelity is summarized in Figure 8. All data has been encoded at 0.833 bpp. We provide, per stack, the root-mean-square error (rms), the peak signal-to-noise ratio (PSNR), and the normalized cross-correlation (Corr.) between adjacent slices. Furthermore, we provide metrics about the planarity of the colorspace (also see Eqn. 6), the rms after projection (rms project.) to a planar colorspace, and the rms after both projection and application of our  $L_2$ -optimal downsampling of the second component (rms downsm.). All pixel values were normalized to the range  $[0, 1]^3$ . The normalized cross-correlation was computed for all pairs of adjacent slices and measures, normalized to the range  $[-1, 1]$ , how close adjacent slices are to each other. Identical images will result in a normalized cross-correlation of 1. The results show that adjacent slices have high correlation and that our compression scheme is able to achieve high fidelity at low bitrates. Furthermore we observe that in all examples the colorspace was close-to-planar. As can

|              |   |   |   |   |   |   |   |  |   |   |   |   |
|--------------|---|---|---|---|---|---|---|--|---|---|---|---|
|              |  |  |  |  |  |  |  |  |  |  |  |  |
| rms total    | 0.0502<br>± 0.0096  | 0.0258<br>± 0.0078  | 0.0442<br>± 0.0124  | 0.0322<br>± 0.0071  | 0.0434<br>± 0.0095  | 0.0274<br>± 0.0061  | 0.0494<br>± 0.0089  | 0.0684<br>± 0.0096   | 0.0567<br>± 0.0119  | 0.0508<br>± 0.0082  | 0.0464<br>± 0.0091  | 0.0324<br>± 0.0072  |
| PSNR         | 26.14dB<br>± 1.74dB   | 32.13dB<br>± 2.67dB   | 27.41dB<br>± 2.52dB   | 30.04dB<br>± 1.94dB   | 27.44dB<br>± 1.89dB   | 31.45dB<br>± 1.93dB   | 26.26dB<br>± 1.57dB   | 23.38dB<br>± 1.27dB  | 25.11dB<br>± 1.83dB   | 25.99dB<br>± 1.42dB   | 26.83dB<br>± 1.70dB   | 30.00dB<br>± 1.95dB   |
| Corr.        | 0.8647<br>± 0.0289  | 0.9577<br>± 0.0107  | 0.9516<br>± 0.0085  | 0.9818<br>± 0.0013  | 0.9700<br>± 0.0029  | 0.9895<br>± 0.0009  | 0.9881<br>± 0.0007  | 0.9726<br>± 0.0023   | 0.9854<br>± 0.0009  | 0.9880<br>± 0.0007  | 0.9861<br>± 0.0010  | 0.9764<br>± 0.0016  |
| Planarity    | 0.8887  | 0.9704  | 0.9485  | 0.9617  | 0.9438  | 0.9846  | 0.9752  | 0.9391   | 0.9703  | 0.9513  | 0.9665  | 0.9579  |
| rms project. | 0.0316  | 0.0141  | 0.0216  | 0.0170  | 0.0246  | 0.0136  | 0.0165  | 0.0263   | 0.0197  | 0.0225  | 0.0168  | 0.0113  |
| rms downsm.  | 0.0324  | 0.0142  | 0.0223  | 0.0173  | 0.0252  | 0.0138  | 0.0173  | 0.0336   | 0.0214  | 0.0245  | 0.0181  | 0.0122  |

**Figure 8:** Results of our compression scheme. We present results for 12 representative stacks exhibiting significantly different statistics.

be seen, our L<sub>2</sub>-optimal downsampling is able to perform chroma subsampling virtually lossless in most cases.

Compared to Jeong et al. [JST\*10], we observe similar or slightly better fidelity. While the method of Jeong et al. can theoretically be implemented on mobile GPUs, its many intermediate render passes coupled with frequent table lookups poses bandwidth and state change requirements which are not yet available on mobile GPUs. We therefore did not consider this method for our mobile client. On desktop GPUs, where both methods are available, the method of Jeong et al. is faster (0.55–0.73ms on a GeForce 285GTX vs. 0.6ms on the faster GeForce 480GTX).

## 6. User Study

We have conducted three different user evaluations to assess the usability and performance of our system for digital pathology.

**Image Quality Evaluation.** In the first evaluation, we asked 12 study participants to conduct an image comparison test to assess the image quality of our compression method. The study participants consist of non-medical experts, such as computer science major students and faculty. For the experiment we used 12 example images from endometrial biopsies, and compressed each image with our method (K-PVR) and with JPEG at a similar bitrate (0.833 bpp). All experiments were conducted using Mac Preview image viewing software on a iMac desktop PC. The participants were given instructions for their task and shown a series of side-by-side images (see Figure 9). The session concluded with verbal questions and feedback. We made three groups of 12 image pairs, i.e., original/K-PVR, original/JPEG, and K-PVR/JPEG. Each of the image pairs were shown to the participants in random order. They were asked to "pick the higher quality image among the two". They were allowed

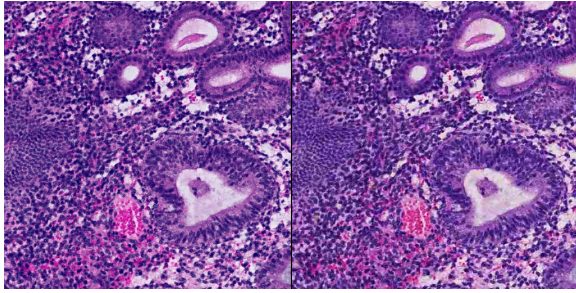
**Table 2:** Compressed image quality evaluation result (numbers shown in the table are the total number of images preferred by the test participants.)

|                 | Preferred Image Type |       |      |       |
|-----------------|----------------------|-------|------|-------|
|                 | Original             | K-PVR | JPEG | Equal |
| Orig. vs. K-PVR | 103                  | 22    | N/A  | 19    |
| Orig. vs. JPEG  | 75                   | N/A   | 12   | 57    |
| K-PVR vs. JPEG  | N/A                  | 64    | 53   | 27    |

to answer "equal" in cases where they could not find discernible differences between the two images. Table 2 shows the result of this study.

As one may expect, the participants preferred the original images to either of the compressed images (71% for original over K-PVR, and 52% for original over JPEG). However, among the people who did not prefer the original images, more people chose K-PVR over JPEG (15% vs. 8%). If we only consider the people who chose one image (i.e., people did not answer "equal"), slightly more people prefer K-PVR to JPEG (17% vs. 13%). Finally, when K-PVR is directly compared to JPEG, 44% people preferred K-PVR and 36% people preferred JPEG. This result shows that K-PVR may introduce more visible artifact than JPEG compared to the original images, but without the presence of the original images the type of artifact in K-PVR might be perceptually more natural than JPEG. We also observed that JPEG suffers from blocking artifacts that are more noticeable when compared with fuzzy/grainy/color-shift artifacts presented in K-PVR. These results are encouraging because our compression method generates images that are visually comparable/superior to JPEG compressed images while providing an efficient hardware decoding option.

**System Usability Evaluation.** We asked three pathologists



**Figure 9:** A sample image pair used in the image quality evaluation. Left: our compression method (K-PVR), Right: JPEG compression.

to conduct a second evaluation, test driving our iPad and desktop client systems and make a clinical diagnosis. The participants were two women and one man, with 8, 15 and 16 years of professional pathology experience respectively. Three slide images of endometrial biopsies were examined using the desktop client. One pathologist navigated the images while the others observed the image on the screen. The task was made as difficult as possible by selecting slides with technical problems, such as thick or folded sections. All three pathologists had previously viewed similar images as part of a validation study for whole slide imaging in endometrial pathology, using the commercially available Hamamatsu NDP view software. Performance on this system was used as the baseline for subsequent comparisons. All three pathologists were able to reach a diagnosis in every case.

The pathologists had several comments with respect to the performance of our system. First the ability to change focus quickly while navigating the images is an advantage for making a diagnosis quickly. They also appreciated the fast frame rate of the viewer. To match the image quality and features of an optical microscope they suggested two additional features: gamma adjustment and the ability to step rapidly between predefined image magnifications (e.g., 2 $\times$ , 4 $\times$ , 10 $\times$ , 20 $\times$  and 40 $\times$ ). They also requested the ability to rotate images. Finally, to facilitate documentation for reports they suggested the option to record all actions that were taken on an image (zoom in/out, pan, etc.) together with the other annotations. These recordings would be useful for teaching and collaboration, and they may be especially useful in situations where a hospital network is not fast enough to support live multi-user reviews. Display of a predefined grid would be particularly helpful for slides with multiple tissue fragments such as lymph node dissections so that pathologists could verify that all parts of the slide were reviewed. All the requested features are straightforward modifications to the current client software and we plan to add them in the near future. The pathologists were uniformly positive about the performance and usability of the iPad client. They found that the multi-touch interface

for image navigation was intuitive and easy to use and more closely matches the interaction with a microscope.

**Collaborative Diagnosis Evaluation.** Four pathologists performed the third evaluation, specifically aimed at collaborative review. There were three men and one woman with 5, 7, 16 and 15 years of pathology experience, respectively. All are subspecialty trained and all have normal color vision. A series of surgically excised anal lesions, clinically and architecturally consistent with benign warts (condyloma) but with premalignant intraepithelial changes (high grade squamous intraepithelial lesion - dysplasia, HGSIL) on histologic review [MMF\*07, SFM\*] were evaluated. This type of lesion is a recognized problem area in pathology as we currently lack clearly defined diagnostic criteria for HGSIL in warty lesions, and these cases are likely to undergo collaborative consensus review by two or more pathologists. One pathologist (JH) identified patients from department files over a 10-year period, each of whom had clinical condylomas containing histologic HGSIL. 23 biopsies from eleven patients (three biopsies from one patient, two biopsies from the remaining ten patients,) were reviewed. For each patient, one of the two biopsies was randomly assigned to the "glass slide review" group and the other to the "whole slide image (WSI) review" group. Slides for the "WSI" group were scanned as thin image stacks (15 planes spaced every 0.75 microns) at 40x objective magnification on a Hamamatsu NanoZoomer 2.0-HT. The scan sizes ranged from 1 to 350GB uncompressed.

Cases for the "glass slide" group were reviewed by each of pathologists (BFJ, RN and EY) and scored for "HGSIL present" with concurrent assessment of three histologic features, namely orderly normal maturation of the squamous epithelium towards the surface, the presence of abnormally maturing (dyskeratotic) single cells within the upper half of the epithelium and the presence of abnormally located and actively dividing cells (mitotic figures) in the upper half of the epithelium. HGSIL was reported as positive regardless of whether it was focal or diffuse. Results were collated and cases where the pathologists disagreed on the presence of HGSIL were reviewed concurrently by all three pathologists at a single multi-headed microscope to render a consensus diagnosis, which was used as the "gold standard". Cases for the "WSI group" were reviewed by each of the same three pathologists alone, and scored for the same parameters as the glass slide group. Pathologists used the desktop client on computers running Windows 7 (Xeon CPU, NVIDIA Fermi GPUs, 30 inch monitors, Gigabit Ethernet). Results were again collated and cases where the pathologists disagreed on the presence of HGSIL were reviewed collaboratively using the "chat" window and annotation tools to reach a consensus diagnosis. The cases reviewed in both groups are very similar with respect to the number with HGSIL. Accurate diagnoses made using both viewing methods with similar levels of initial agreement between the pathologists after their individual review (glass slide 8/11 and WSI 10/12). A con-

**Table 3: Results of Glass slide versus WSI review.**

|                    | # Biopsies | HGSIL present | Initial agreement after individual review | Consensus agreement | Review Time                        |
|--------------------|------------|---------------|---|---------------------|------------------------------------|
| Glass slide Review | 11         | 6 of 11       | 8 of 11 (73 %)                            | 100%                | 1.21 to 6.00 min, average 2.47 min |
| WSI Review         | 12         | 6 of 12       | 10 of 12 (91%)                            | 100%                | 1.02 to 5.47 min, average 2.56 min |

sensus diagnosis was reached in all cases. One pathologist timed each case and the range and average viewing time are comparable between modalities. All three pathologists reliably identified the three histologic features, namely epidermal maturation, high-level dyskeratosis and high-level mitoses. The results are detailed in Table 3.

The performance of our digital pathology system assessed in terms of speed and accuracy of final diagnosis, was comparable to traditional glass slide review. Pathologists could identify all features necessary to make the diagnosis with a similar degree of ease and within the same time frame. This equivalence between WSI and glass slide systems is only just being realized [YYK\*] and is a major strength of our system. Additional strengths of our system are the ability for remote collaborative review, the ability to focus through thin image stacks and the ability to rapidly read through aligned thin z-stacks of serial section.

Collaborative review is an essential part of the pathology experience, from initial training through a professional lifetime of practice, whether as a general pathologist or as a subspecialist. Geographic constraints often limit interaction, and remote collaborative review with real time "chat" removes them and simulates the experience of using the multi-headed microscope. When "double-scoping" usually only one pathologist "drives" the slide, and the experience for the viewer(s) is intrinsically more passive. With existing digital video telepathology systems (used primarily for frozen section diagnosis), the passive viewer is also often the consultant. Enabling the consultant to actively drive the slide for diagnosis is more efficient and increases confidence in a diagnosis that may have significant clinical implications. With our system, all users have the ability to drive the slide in their own style, which differs between individuals, to review in a manner in which they are diagnostically confident and to point out specific features or areas of interest to ensure accurate consensus review. Active slide review by all parties is an advantage over traditional consensus microscope review. In addition, the ability to collaboratively review annotated structures or highlighted areas identified during solo review of multiple individuals is also an advantage. Digital annotations can also have a degree of microscopic precision that is lacking using the traditional permanent marker approach. Use of voice or voice recognition software during chat sessions may also enhance communication and improve efficiency.

Another advantage of our system is the ability to focus through a thin image stack. For example, in this study we assessed the presence of mitotic figures within the epithelium. To accurately assess mitotic figures, it can be necessary to focus up and down through the tissue, and this has been a limitation of single plane digital images. The use of thin image stacks permits the pathologist to focus through the WSI in the same way as a traditional glass slide and enhances the ability to diagnose features that require limited three-dimensional data for accurate identification. Accurate identification of mitoses is of fundamental importance in other areas of pathology, where the presence and number of mitoses are used for tumor staging and to guide management decisions.

Finally, many small biopsies are cut as multiple serial sections onto a single slide (see Figure 10). The sections are cut at about 4 microns and together represent a small tissue volume (20 microns thick in this example). These additional sections may be enough to provide useful information for the pathologist. For example, in this study we assessed the presence of dyskeratotic cells and mitotic figures in the upper half of the epithelium. To make this assessment of location within the epithelium, the tissue should ideally be sectioned perpendicular to its surface. Tissue is frequently tangentially oriented and cut, hindering this assessment. A feature of our system is the ability to "stack" the images of the serial sections on the slide (each image having multiple focal planes) and then rapidly step through them. Tissue orientation can be more rapidly and efficiently assessed than by manual inspection of the slides.

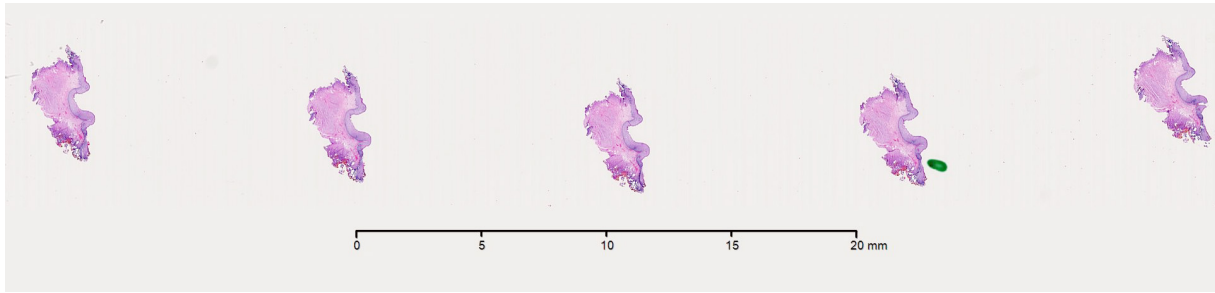
## 7. Discussion

### 7.1. Comparison to other digital pathology systems

Existing digital pathology systems can be classified according to three different types:

**Dedicated viewer with data sharing:** In this group, the viewer software is designed around displaying images on a dedicated graphics workstation. The dedicated viewers provide advanced view manipulation and annotation tools, and support limited collaboration through network data sharing. Remote collaborative review requires opening the same image and annotation files. Representative applications are Hamamatsu's NDP.view [NDP] and Biolumica's Viewer

**Dedicated viewer with conference functionality:** In this



**Figure 10:** Multiple serial 4  $\mu$ m sections of tissue are routinely placed on a single glass slide for review. This also shows the "traditional method" of annotation (a green permanent marker point next to the fourth tissue section, which implied the section/region of interest).

approach, a distributed visualization and data management system supports remote data sharing and collaborative review. Some commercially available systems have advanced collaboration features, e.g., text/voice chat and view/annotation sharing. Among them, the Olympus OlyVia system [?] is similar to our PC client viewer. It provides advanced annotation and built-in conference features via a data server. However, the design of the conference features is different from that of our system. OlyVia supports passive collaboration – a single user, i.e., *Speaker*, has the right for view manipulation and annotation, and this information is broadcast to the other users. While this design is consistent with the way pathologists do collaborative review on a multi-headed microscope, but it does not fully utilize the flexibility of distributed computer systems. In contrast, our system allows multiple users to change annotations concurrently, and any user can follow any other other's view in realtime without the constraint of there being just a single view shared by all users. Lack of mobile client support is another drawback of their system. Aperio provides iPad client software (ePath Viewer [?]), but it is a simple image viewer without collaboration functions.

**Web-based systems:** An advantage of this approach is that users do not have to install special viewer software. Images can be viewed via the web from almost any computer with an internet connection. One example is the NYU Virtual Microscope [NYU]. This system is built upon the Google Maps framework and, so, is optimized for navigating large 2D images. However, the system only supports simple marker-based annotation and sharing, and there is no advanced collaboration functionality and no ability to quickly advance through focal planes. Recently Kitware developed a WebGL-based virtual microscope system, *SlideAtlas* [?]. This system implements per-image pyramid processing (similar to our system) and interactive registration of images to handle extremely large image data efficiently. The current system provides basic functionality for collaborative review. At any one time a single user can share a view and annotation that is followed by others. The system supports multi-touch input on mobile web-based viewers and the ability to leverage GPU acceleration by using GL shaders. However the cur-

rent system uses GPU acceleration only for texture-based 2D rendering and its performance suffers during 3D navigation (i.e., changing focal planes) because of the fact that each z-slide is treated separately.

In summary a major difference between our system and others is the sophistication of our collaborative tools. Our system is unusually flexible in its ability to support realtime view and annotation sharing between multiple users across diverse client systems. Each user can drive an image or follow the view of another as they wish. A previous study [?] has shown that the diagnostic path in four dimensions (i.e., x, y, time and zoom) is an important factor affecting accuracy of histopathology diagnosis. Most existing systems provide only limited view sharing functionalities such as one to many static view broadcasting or switching the regions of interest. By contrast our system supports multi-way realtime view sharing so that the users can interactively follow the diagnostic path of others as if they are looking at the screen together. In addition there can be multiple subgroups of collaborative review under the same session when needed; that is, the system has the flexibility of allowing more than one collaborative view per image. This added flexibility is possible mainly due to the novel design of our advanced render-local remote visualization system that supports GPU hardware acceleration on heterogeneous client platforms by leveraging the domain-specific image compression method.

## 7.2. Usage Scenarios

We envision the following usage scenarios.

**Take-away visualization.** Domain experts either connect to the server using WiFi or download content to mobile devices. They then sit together and each of them explores the data set in parallel while discussing their findings. The compact form factor of mobile devices allows them to be passed around in order to gather secondary opinions. This in essence parallelizes the time-consuming exploration of the data while still maintaining the classical, collaborative way of diagnosis. The major advantage of migrating to a digital data representation is the fact that all domain experts can actively navigate the data.

**Distributed diagnosis.** In this scenario, domain scientists cannot meet physically. Our system allows data to be distributed with ease and at WiFi network bandwidths. A voice channel can be established using, e.g., a phone line or Skype connection, and landmarks—once found—can be shared via our client-server architecture. This implements and augments the classical external consultation which otherwise requires samples to be physically shipped. It is worth noting that physical shipment usually implies a 4 week turn-around time due to administrative overhead. Also, a digital sample (unlike a physical one) can be accessed by multiple external experts at the same time.

### 7.3. Limitations

One limitation of our system is that we always assume input data to be measured at 8 bits per color channel. This is an intrinsic problem of the mobile GPU since PVRTC only supports 8 bit color channels, and the hardware is likely to perform the bilinear upscales during decoding at a limited and fixed precision. While it is clearly possible to extend our desktop client in the future, mobile clients will not immediately benefit of this extension.

Another limitation of the system is that PVRTC encoding is currently costly (on the order of a minute per  $512 \times 512 \times 15$  stack) and we will investigate a CUDA-based faster encoder in the future.

The current system is built on top of a standard Apache web server, so the performance of handling multiple clients solely depends on the ability of the web server and network speed. We will investigate parallel approaches, such as using a distributed web server, to process a large number of concurrent clients' requests more efficiently.

### 8. Conclusions

We have presented the first interactive collaborative whole slide digital pathology system that can efficiently handle navigation of multi-plane whole slide images on both multi-touch mobile and desktop computing platforms. We implemented mobile and desktop clients that provide essential tools for remote collaborative diagnosis, such as realtime view sharing, digital annotation and chat functions, shared by multiple users in remote locations. To minimize bandwidth and to leverage hardware decompression on mobile devices, we developed a novel domain-specific hardware-accelerated image compression method that achieves extremely low bit rates (0.833bpp) with minimal loss of image quality. The user feedback from professional pathologists is very encouraging and indicates that our system is comparable/superior to existing commercial packages including those from Aperio, Hamamatsu and Olympus.

In future work we will implement several additional annotation features that were suggested by the pathologists,

such as gamma adjustments, fixed resolution steps, image rotations, and action recordings. We plan to develop several multi-touch user interface specifically designed for pathology diagnosis tasks. We will also continue to improve our image compression method. The KLT color transform is sensitive to noise, and we plan to use a preconditioned form that is robust to outliers. The encoding is currently slow due to iterative fitting. We will investigate novel PVRTC encoding algorithms to overcome this issue. We will also explore a CUDA-based encoder implementation. Finally, we plan to deploy our system to our pathology collaborators for use in collaborative clinical diagnosis.

### Acknowledgements

We would like to thank Tobias Hector from Imagination for providing 64bit PVRTexLib and technical support, and the students and faculty at UNIST ECE for participating the user study. This work has been partially supported by the year of 2011 research fund of UNIST, Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A1039929), NSF grants PHY 0938178, OIA 1125087, NIH grant 2R44MH088088-03, the Conte Center at Harvard (NIMH), Transformative R-01 (NIH), the Gatsby Charitable Trust to JWL, Department of Molecular and Cellular Biology at Harvard University, NVIDIA, Google, and the Intel Science and Technology Center for Visual Computing.

### References

- [App] APPLICATION P. V.: *VisIt: Visualize It*. Lawrence Livermore National Laboratory. 3
- [Avi95] AVINASH G.: Image compression and data integrity in confocal microscopy. *SCANNING—The Journal of Scanning Microscopies* 17, 3 (1995), 156–160. 3
- [BA83] BURT P. J., ADELSON E. H.: The Laplacian pyramid as a compact image code. *IEEE Transactions on Communication* 31, 4 (1983), 532–540. 3
- [Bet00] BETHEL W.: Visualization dot com. *IEEE Computer Graphics and Applications* 20, 3 (2000), 17 – 20. 3
- [BSL\*00] BETHEL W., SHALF J., LAU S., GUNTER D., LEE J., TIERNEY B., BECKNER V., BRANDT J., EVENSKY D., CHEN H., PAVEL G., OLSEN J., BODTKER B.: Visipult - using high-speed wans and network data caches to enable remote and distributed visualization. In *Super Computing* (2000), pp. 108–119. 3
- [CTG\*03] COCKSHOT W., TAO Y., GAO G., BALCH P., BRIONES A., DALY C.: Confocal microscopic image sequence compression using vector quantization and 3d pyramids. *SCANNING—The Journal of Scanning Microscopies* 15 (2003), 247–256. 3
- [EE99] ENGEL K., ERTL T.: Texture-based volume visualization for multiple users on the world wide web. *Eurographics Workshop on Virtual Environments* (Jan 1999). 3
- [EEHT00] ENGEL K., ERTL T., HASTREITER P., TOMANDL B.:

- Combining local and remote visualization techniques for interactive volume rendering in medical applications. *Proc. IEEE Visualization* (Jan 2000). 3
- [Fen03] FENNEY S.: Texture compression using low-frequency signal modulation. In *ACM SIGGRAPH/EUROGRAPHICS Graphics Hardware* (2003), pp. 84–91. 6
- [FWSB07] FORLINES C., WIGDOR D., SHEN C., BALAKRISHNAN R.: Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), CHI '07, ACM, pp. 647–656.
- [GG91] GERSHO A., GRAY R. M.: *Vector Quantization and Signal Compression*, 1st ed. Kluwer Academic Press, 1991. 3
- [Gol] GOLD E.: <http://www.ensight.com/ensight-gold.html>. 3
- [GY95] GHAVAMNIA M. H., YANG X. D.: Direct rendering of laplacian pyramid compressed volume data. In *Proceedings of IEEE Visualization* (1995), pp. 192–199. 3
- [GY08] GILBERTSON J., YAGI Y.: Histology, imaging and new diagnostic work-flows in pathology. In *Diagnostic Pathology* (2008), vol. 3, p. S14.
- [GZV00] GOYAL V. K., ZHUANG J., VETTERLI M.: Transform coding with backward adaptive updates. *IEEE Trans. Inform. Theory* 46 (2000), 1623–1633. 7
- [HRC\*06] HEIRICH A., RAFFIN B., CEDILNIK A., GEVECI B., MOREL K.: Remote large data visualization in the paraview framework. *Eurographics Parallel Graphics and Visualization* (2006), 163–170. 3
- [IBM05] IBM: *IBM Deep Computing*. Tech. rep., IBM Systems and Technology Group, 2005. 3
- [JST\*10] JEONG W.-K., SCHNEIDER J., TURNEY S. G., FAULKNER-JONES B. E., MEYER D., WESTERMANN R., LICHTMAN J., PFISTER H.: Interactive histology of large-scale biomedical image stacks. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization)* 16, 6 (2010), 1386–1395. 3, 4, 5, 10
- [Kar47] KARHUNEN K.: Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.* 37 (1947), 1–79. 6
- [Ker09] KERR D.: Chrominance subsampling in digital images. *The Pumpkin* 1, 2 (2009). 7
- [Loé78] LOÉVE M.: *Probability Theory*, 4th ed., vol. II of *Graduate Texts in Mathematics* 46. Springer-Verlag, 1978. 6
- [LP03] LAMAR E., PASCUCCI V.: A multi-layered image cache for scientific visualization. *Proc. of the IEEE Symposium on Parallel and Large-Data Visualization and Graphics* (Jan 2003). 3
- [MC00] MA K., CAMP D.: High performance visualization of time-varying volume data over a wide-area network status. *Proc. ACM/IEEE Conference on Supercomputing* (Jan 2000). 3
- [MMF\*07] MCCLOSKEY J., METCALF C., FRENCH M., FLEXMAN J., BURKE V., BEILIN L.: The frequency of high-grade intraepithelial neoplasia in anal/perianal warts is higher than previously recognized. *Int J STD AIDS* 18, 8 (2007), 538–42. 11
- [NDP] NDP: <http://sales.hamamatsu.com/en/products/system-division/virtual-microscopy/products/software.php>. 3, 12
- [NH92] NING P., HESSELINK L.: Vector quantization for volume rendering. In *Proceedings of the Workshop on Volume Visualization* (1992), pp. 69–74. 3
- [NYU] NYU virtual microscope. <http://cloud.med.nyu.edu/virtualmicroscope>. 3, 13
- [Par] PARAVIEW: <http://www.paraview.org/>. 3
- [PTVF02] PRESS W., TEUKOLSKY S., VETTERING W., FLANNERY B.: *Numerical Recipes in C++: The Art of Scientific Computing*, 2 ed. Cambridge University Press, 2002. 7
- [Sco] SCOPE A. I.: <http://www.aperio.com/pathology-services/imagescope-slide-viewing-software.asp>. 3
- [SFM\*] SCHLECHT H. P., FUGELSO D. K., MURPHY R. K., WAGNER K. T., DOWEIKO J. P., PROPER J., DEZUBE B. J., PANTHER L. A.: Frequency of occult high-grade squamous intraepithelial neoplasia and invasive cancer within anal condylomata in men who have sex with men. 11
- [SME02] STEGMAIER S., MAGALLÓN M., ERTL T.: A generic solution for hardware-accelerated remote visualization. *Proc. of the Symposium on Data Visualisation* (Jan 2002). 3
- [SW03] SCHNEIDER J., WESTERMANN R.: Compression domain volume rendering. In *Proc. 14th IEEE Visualization* (2003), pp. 293–300. 3
- [TPSP02] TAN D. S., PAUSCH R., STEFANUCCI J. K., PROFIT D. R.: Kinesthetic cues aid spatial memory. In *CHI '02 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2002), CHI EA '02, ACM, pp. 806–807.
- [YYK\*] YAGI Y., YOSHIOKA S., KYUSOJIN H., ONOZATO M., MIZUTANI Y., OSATO K., YADA H., MARK E. J., FROSC M. P., LOUIS D. N.: An ultra-high speed whole slide image viewing system. *Analytical Cellular Pathology* 35. 12

## Appendix

This appendix describes a method to derive the  $L_2$  optimal downsampling used in Section 4.

Let  $I$  be a scalar input image  $I : [1 \dots N] \times [1 \dots M] \rightarrow \mathbb{R}$ . We wish to obtain an image  $I' : [1 \dots N/2] \times [1 \dots M/2] \rightarrow \mathbb{R}$ , such that a GPU-accelerated bilinear upscale  $\uparrow^2 \star I'$  (see also Fig. 11) results in a minimal  $L_2$  error with respect to  $I$ . In other words, we want to obtain a filter  $\downarrow_2$  with

$$\tilde{\downarrow}_2 := \arg \min_{\kappa} \left\| \left( I - \tilde{\uparrow}^2 \star (\kappa \star I) \right) \right\|_2^2 \quad (7)$$

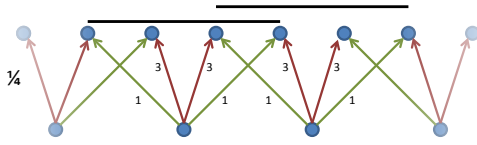
Since  $\tilde{\uparrow}^2$  is a 2D separable kernel, so is  $\tilde{\downarrow}_2$ . The reason is that a separable reconstruction kernel, once separated, does not take information along the orthogonal axis into account. Hence, it is sufficient to treat the problem in 1D and then generalize it to images in a tensor product fashion. We therefore consider an image row  $r : [1 \dots N] \rightarrow \mathbb{R}$  and its downsampled version  $r' : [1 \dots N/2] \rightarrow \mathbb{R}$ .

In matrix formulation,  $r = \tilde{\uparrow}^2 \star r'$  corresponds to a matrix-vector multiplication  $r = Ar'$ , where  $A$  is a circular matrix in  $\mathbb{R}^{N \times (N/2)}$ . Formulating the problem in matrix notation thus couples  $r'$  to  $r$  by an overdetermined system of linear equations. We solve this system by its Moore-Penrose pseudo inverse  $(A^T A)^{-1} A^T$  which results in the desired  $L_2$  optimal

filter  $\downarrow_2$ . However, due to boundary effects, this is only true for images with infinite extent.

We thus construct a sufficiently large matrix  $A$  with

$$A = \frac{1}{4} \begin{pmatrix} 3 & 0 & 0 & \cdots & 1 \\ 3 & 1 & 0 & \cdots & 0 \\ 1 & 3 & 0 & \cdots & 0 \\ 0 & 3 & 1 & \cdots & 0 \\ 0 & 1 & 3 & \cdots & 0 \\ 0 & 0 & 3 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 3 \\ 1 & 0 & 0 & \cdots & 3 \end{pmatrix}. \quad (8)$$



**Figure 11:** A schematic overview of hardware-accelerated linear upscale. The bottom pixels are upscaled to twice the resolution at the top. By doing so, each pixel gives a weighted contribution to four pixels in the finer image (domains marked by black bar at the top). Weights are  $\frac{1}{4}$  and  $\frac{3}{4}$  respectively.

We then proceed by computing the pseudo-inverse  $(A^T A)^{-1} A^T$  and select its middle row to minimize boundary effects. This results in the following convolution kernel.

$$\tilde{\downarrow}_2 = 2 \downarrow_2 * \left[ \cdots, 0, \frac{1}{3^3}, 0, -\frac{1}{3^2}, 0, \frac{1}{3}, \frac{1}{3}, 0, -\frac{1}{3^2}, 0, \frac{1}{3^3}, 0, \cdots \right],$$

where  $\downarrow_2$  is a subsampling by a factor of 2 (comb filter). The filtered pixel's position is then between the two  $\frac{1}{3}$  factors in the kernel.

This kernel can be efficiently implemented as an IIR (infinite impulse response) filter by splitting it into an odd half

$$2 \left[ \cdots, 0, \frac{1}{3^3}, 0, -\frac{1}{3^2}, 0, \frac{1}{3} \right],$$

and an even half

$$2 \left[ \frac{1}{3}, 0, -\frac{1}{3^2}, 0, \frac{1}{3^3}, 0, \cdots \right].$$

By observing that non-zero factors decay with a constant factor of  $-\frac{1}{3}$ , we can thus implement the filter by scanning the image from the left and the right using a single accumulation register. After scanning the image, odd and even contributions are added to form the final result.

Since the kernel implies an infinite image domain, we introduce clamp-to-edge boundary conditions as follows. At the image boundary, one half of the filter (odd at the left

boundary of the image, even at the right boundary) will overlap unavailable values. To multiply all of these values with the first pixel in the scan, we observe that (after grouping terms  $i$  and  $i + 1$  and adding a dummy term for  $i = 0$ )

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N (-1)^{i-1} \frac{2}{3^i} = \lim_{N \rightarrow \infty} \left( 4 \sum_{i=0}^N \frac{1}{9^i} \right) - 4 = \frac{1}{2}. \quad (9)$$

Hence, instead of starting scans with a weight of  $\frac{2}{3}$ , as the kernel would suggest, we start with a value of  $\frac{1}{2}$  to correctly treat boundaries.

This method is extremely fast. Assuming an  $N \times N$  image, we need one scan along the X-axis from the left accessing odd pixels and one from the right accessing even pixels. Together, these two scans access all  $N^2$  pixels in the image to compute an  $(N/2) \times N$  image. A second pass along the Y-axis then reduces the image to  $(N/2) \times (N/2)$ . The cost of this method is thus still in  $O(N)$  and is in practice about half as fast as the traditional box filter mipmap generation.