

Original Article

Histostitcher™: An informatics software platform for reconstructing whole-mount prostate histology using the extensible imaging platform framework

Robert J. Toth^{1,2}, Natalie Shih³, John E. Tomaszewski⁴, Michael D. Feldman³, Oliver Kutter⁵, Daphne N. Yu⁵, John C. Paulus Jr⁵, Ginaluca Paladini⁵, Anant Madabhushi³

¹Department of Biomedical Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ, ²Department of Biomedical Engineering, Case Western Reserve University, Cleveland, OH, ³Department of Pathology and Laboratory Medicine, University of Pennsylvania, Philadelphia, PA, ⁴Department of Pathology and Anatomical Sciences, University at Buffalo, Suny, Buffalo, NY, ⁵Department of Imaging and Computer Vision, Siemens Corporate Research, Princeton, NJ, USA

E-mail: *Anant Madabhushi - anant.madabhushi@case.edu

*Corresponding author

Received: 27 August 13

Accepted: 18 December 13

Published: 28 March 14

This article may be cited as:

Toth RJ, Shih N, Tomaszewski JE, Feldman MD, Kutter O, Yu DN, et al. Histostitcher™: An informatics software platform for reconstructing whole-mount prostate histology using the extensible imaging platform framework. *J Pathol Inform* 2014;5:8.

Available FREE in open access from: <http://www.jpathinformatics.org/text.asp?2014/5/1/8/129441>

Copyright: © 2014 Toth RJ. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Context: Co-registration of ex-vivo histologic images with pre-operative imaging (e.g., magnetic resonance imaging [MRI]) can be used to align and map disease extent, and to identify quantitative imaging signatures. However, ex-vivo histology images are frequently sectioned into quarters prior to imaging. **Aims:** This work presents Histostitcher™, a software system designed to create a pseudo whole mount histology section (WMHS) from a stitching of four individual histology quadrant images. **Materials and Methods:** Histostitcher™ uses user-identified fiducials on the boundary of two quadrants to stitch such quadrants. An original prototype of Histostitcher™ was designed using the Matlab programming languages. However, clinical use was limited due to slow performance, computer memory constraints and an inefficient workflow. The latest version was created using the extensible imaging platform (XIP™) architecture in the C++ programming language. A fast, graphics processor unit renderer was designed to intelligently cache the visible parts of the histology quadrants and the workflow was significantly improved to allow modifying existing fiducials, fast transformations of the quadrants and saving/loading sessions. **Results:** The new stitching platform yielded significantly more efficient workflow and reconstruction than the previous prototype. It was tested on a traditional desktop computer, a Windows 8 Surface Pro table device and a 27 inch multi-touch display, with little performance difference between the different devices. **Conclusions:** Histostitcher™ is a fast, efficient framework for reconstructing pseudo WMHS from individually imaged quadrants. The highly modular XIP™ framework was used to develop an intuitive interface and future work will entail mapping the disease extent from the pseudo WMHS onto pre-operative MRI.

Key words: Computer aided diagnosis, prostate histology, extensible imaging platform

Access this article online

Website:
www.jpathinformatics.org

DOI: 10.4103/2153-3539.129441

Quick Response Code:



INTRODUCTION

Background

Radical prostatectomy is a common therapy for prostate cancer (CaP), with 158,000 performed in 2009,^[1] which involves excising the gland from the body. Co-registration with pre-operative imaging (e.g., magnetic resonance imaging [MRI]) can be used to align and map CaP extent and to identify quantitative imaging signatures.^[2] This can be used to train a computer aided diagnosis (CAD) system to detect CaP extent *in vivo* whether by needle biopsies^[3] or by *in vivo* MRI.^[4] Diagnosing CaP extent using *in vivo* MRI scans is an important step toward accurate diagnosis without invasive biopsy. In addition, CAD systems could potentially be trained to assign risk scores to patients and hence that patients with CaP who have a lower risk score (i.e., less aggressive CaP) might be identified as candidates for active surveillance as opposed to immediate therapy. To train such a CAD system requires the CaP extent to be delineated on the MR imagery of the prostate.

Radiologist derived annotations of the CaP extent on the MRI is usually subject to inter-observer variability.^[5] In addition, these CAD systems need a sufficient number of annotated cases for the classifier training phase and the evaluation of the learning algorithms. The ground truth for determining the disease extent in the case of CaP is provided by histological examinations. An alternative to radiologist annotation of CaP extent solely based off visual inspection of CaP on MRI is to first annotate CaP areas in whole mount histological sections (WMHS) which can be done with much higher fidelity, precision, confidence and with significantly lower inter-reader variability compared to performing annotations on MRI) which can be mapped to onto *in vivo* MRI scans, providing the ground truth for CaP extent for training and subsequently quantitatively evaluating CAD techniques for CaP on MRI.

However, obtaining digitized WMHS to use for co-registration is cumbersome since (a) digital scanners are unable to accommodate the entire whole mount slice, (b) significant technical expertise is required for WMHS preparation and (c) additional storage requirements exist due to the larger sizes of the whole mounts.^[6,7] As such, only 16% of respondents from a survey of 255 members of the International Society of Urological Pathology reported using whole mounts.^[7] Alternative methods such as cryosectioning can contain additional problems such as introducing artifacts relating to the freezing of the tissue.^[8] Most centers working with freshly excised specimens prefer to section the gland into four quadrants. Although this procedure is somewhat easier compared with whole mount sectioning (which requires a certain degree of skill to do properly), it also introduces a registration challenge with the imaging

data, since co-registering a WMHS to the corresponding *in vivo* MRI slice is easier compared with attempting to individually align four individual quadrants onto a single MRI section. This therefore means that to co-register the *ex-vivo* WMHS with *in vivo* imagery (so that *in vivo* CaP features on MRI can be identified and used to train CAD classifiers), the four imaged quadrants must be reconstructed into a single contiguous image.

Prior Work

To rectify the problem of having four separate images per histology slice, the Histostitcher™ software^[6] was developed [Figure 1]. Histostitcher™, a minimally interactive digital quadrant stitching program to recreate pseudo WMHS first presented in,^[6] allows one to derive the benefits of using WMHS without the downside of cutting and storing the WMS. This tool allowed clinicians to reconstruct a stitched WMHS by selecting common fiducials on the edges of adjacent quadrants. The algorithm then computes the optimal affine transformation of these quadrants by minimizing the mean squared error of the fiducials and reconstructs the entire WMHS gland by merging the adjacent slices. Histostitcher™ was designed to expedite the process of creating a pseudo WMHS, in which an accurate stitching can be generated with only a few carefully selected fiducials, which can then be registered to *in vivo* MRI. The results from the original Histostitcher™ prototype resulted in accurate pseudo WMHS and was faster than using conventional image editing software, but still took almost 1 h to properly stitch a given slice. Several limitations contributed to the inefficiencies:

- The conventional central processing unit based rendering of the large images is quite slow and non-interactive
- The graphical user interface (GUI) environment resulted in a non-ergonomic interface for clinicians to select fiducials
- There is no easy way to transform the quadrants prior to selecting fiducials to visualize and preview the alignment and reconstruction.

Whereas this first prototype [Figure 1] had the core functionality required for a digital prostate histology

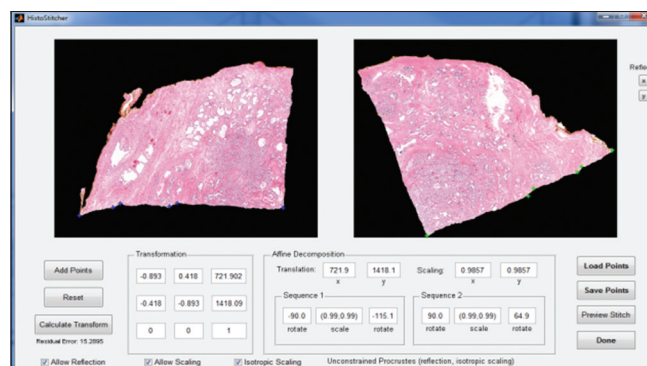


Figure 1: Previous version of Histostitcher™ graphical user interface usable prototype developed using Matlab

reconstruction system, the slow performance and inefficient workflow prohibits its practical use in larger cohort studies such as in clinical trials.

In this study, we present a novel software platform for stitching histology quadrants into a pseudo WMSH using an efficient, professional interface for use by the clinical community, based on the algorithm presented in.^[6] The creation of such a platform can be run efficiently and can be used in large clinical trial cohort studies. This platform has smooth rendering and is built upon the highly modular and extendible, extensible imaging platform (XIP™) allowing rapid incorporation of new features. While the original prototype was Matlab based and clunky, only working on computers with certain specifications, the new system is efficient and easily able to handle large images even on devices such as tablets. For example, since the original system loaded the entire quadrant into memory at once, it required a computer with significantly more memory than a standard desktop personal computer (PC). By comparison, the new system intelligently loads and caches histology data only in the visible viewport, thus enabling it to run on a multitude of devices. This opens up the possibility of viewing the histologic sections and performing annotations on tablets and machines with various specifications.

An overview of the new workflow is presented in Figure 2. After sectioning and digitizing the images, the images are loaded into the Histostitcher™ software. This is done using an intelligent caching system, such that only the pixels visible on the screen are loaded into memory. Then the clinician can add, delete, or modify fiducials common to two quadrants. After previewing the result, the WMHS is finalized by applying the stitching algorithm to the entire, full resolution images. In addition, for better interactions of stitching workflow, multi-touch interfaces have been explored.

The rest of the paper is organized as follows. The “Subjects and Methods” section describes the software architecture for Histostitcher™, the “Results and Discussion” section describes results of several use-cases by clinicians and the “Concluding Remarks” section describes future directions.

SUBJECTS AND METHODS

General Workflow

The revised workflow in the new Histostitcher™ software system entails the following steps,^[9] shown in Figure 3 below.

Module 1

Loading

The operator loads up to four histology image quadrants to be stitched and positions them in their correct relative anatomical locations. Only the pixels in the current viewport are read from disk, to facilitate efficient rendering and decrease the memory requirements of Histostitcher™.

Module 2

Fiducial selection

The loaded quadrants can be interactively panned, rotated and zoomed in order to identify common features across the quadrants that can then be used to align the different quadrants. Fiducials common to the edges of two quadrants are visually located. The clinician picks two quadrants and places corresponding fiducials on the quadrants.

Module 3

Stitching

Once the clinician has placed a sufficient number of fiducials, stitching is performed by registering the two quadrants.^[6] A low resolution preview of the stitched

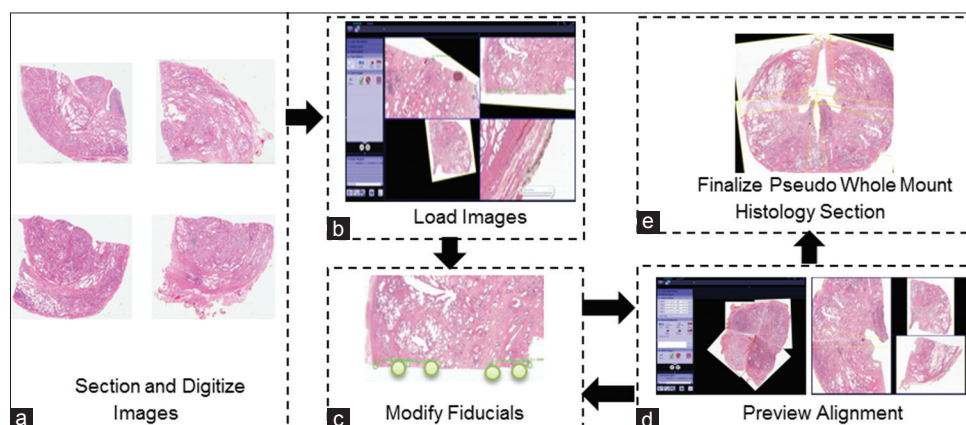


Figure 2: Workflow of the extensible imaging platform based Histostitcher™ system. (a) First, a set of four quadrants are sectioned and digitized; (b) They are then loaded into the Histostitcher™ software; (c) Fiducials common to two or more edges are selected; (d) The stitched result can then be previewed. The fiducials can then be moved until the desired alignment is achieved, at which point the final stitched whole mount histology section image is output (e)

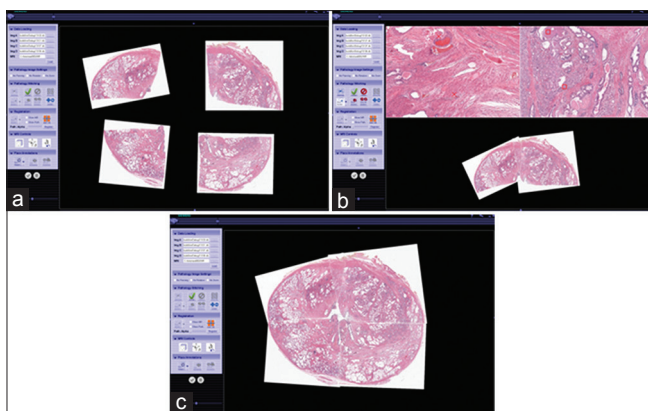


Figure 3: The display during the stitching process (Top). The initial four quadrant images are arbitrarily rotated and translated relative to each other (Middle). Two adjacent quadrants have been stitched, where the top two viewports shows zoomed in views of the two quadrants where the corresponding fiducial markers are placed. Blending has been performed at the intersection of the stitched quadrants in the bottom result view (Bottom). (c) Shown here is the final result, after all quadrants have been stitched together into a fully reconstructed prostate whole mount histology section. Some loss of tissue and non-linear motion can result from the quadrants' slicing, which is why a perfect reconstruction is not always achieved

result is rendered. If the result of the stitching is acceptable, the clinician can stitch the remaining quadrants. Otherwise the clinician can reject the result and refine, add, or remove fiducials. The final pseudo WMHS is generated from the stitching. At any point in the workflow, the user can reject the results and return to previous results, or save the session for continuation at a later date.

XIP Overview

The use of the XIP^[10] modular architecture in the revised Histostitcher™ application greatly facilitated the incorporation of the new workflow, stitching and registration algorithm.^[6] The modularity of XIP allows the integration of existing image viewing modules into a new display composition and interaction. Advantages of using XIP™ include:

- The core histology stitching workflow is a 2D workflow of multiple large 2D color image quadrants
- The XIP™ scene graph structure partition the display area into separate viewport regions
- Each viewport can be configured with a different camera configuration for 2D or 3D viewing and interactions as pre-configured or dynamically selected based on input data subgraphs
- Viewports for 2D image viewing allow the operator to pan, zoom, or in the stitching workflow, perform in-plane rotation. Viewports for 2D image stacks may, in addition, allow the operator to scroll through the stack of 2D images. Viewports for 3D images further allow the operator to rotate the image plane to arbitrarily oriented angles and scroll through the images at oblique angles to select the optimal viewing plane

- Graphics can be added to any viewport for the creation of fiducial markers and annotations
- XIP™ also supports the use of the OpenGL graphics pipeline and Compute Unified Device Architecture (CUDA),^[11,12] allowing for easy integration with graphics processor unit (GPU)-accelerated rendering.

Stitching

The application supports loading of standard tagged image file format (TIFF)-histology slices (each quadrant containing several gigabytes of data), which are pre-converted into a multi-resolution compressed representation file cache for loading and rendering efficiency. The software is capable of stitching most virtual slides and the ability to load additional data types requires only minor changes to the platform and is planned for future releases. Once data has been loaded, the first step is to perform stitching of the quadrants to obtain the WMHS. The original workflow of the Histostitcher™ Matlab prototype^[6] consisted of selecting two quadrants to stitch, adding the fiducials, computing the optimal registration and saving the result. In addition, a preview function allowed the physician to compute a first approximation of the registration result and add additional fiducials when necessary. The whole process had to be repeated for all the quadrants until the complete slice had been reconstructed. Figure 3 below illustrates the stitching workflow with significantly improved ergonomics for viewing and defining corresponding fiducials.

The new design and workflow aims to address the previous shortcomings as follows:

- Smooth GPU rendering of quadrants and stitched result
- Ability to pan, zoom and rotate to align the quadrants prior to placing fiducials
- Ability to easily move existing fiducials and quickly regenerate a new stitched preview.

These requirements are fulfilled with the multi-resolution compressed representation, an intelligent caching system and an out-of-core rendering design.

The Histostitcher™ platform allows the clinicians to view all the quadrants simultaneously and stitch them in any order. Fast GPU rendering in XIP™ allows clinicians to smoothly align the quadrants prior to selecting fiducials. Panning, zooming and rotating take advantage of an intelligent caching system, to align the quadrants in real time. This allows the users to move all the quadrants into an approximately correct position prior to selecting fiducials. Once selected, fiducials can be added, moved and removed again at any point in the work flow and the ability to save and load previous stitching sessions allows the easy continuation of work by editing existing fiducials.

The ability to move and delete fiducials dynamically and see real time update of the stitching results is made

possible by the fast rendering capabilities. The algorithm for using the fiducials to calculate the transformation has been fully described previously.^[6] A multi-resolution pyramid cache is generated for each histology quadrant to allow efficient, real-time rendering of the quadrants [Figure 4].

Software Architecture Overview

Internally, the system is separated into two components running in separate processes: The front end and service process and the rendering process. Inter-process communication keeps these two components completely independent of each other and allows for communication of user inputs and data exchange. An overview of the Histostitcher™ software architecture can be seen in Figure 3 above.

Since XIP™ is built on top of the open inventor framework,^[13] the nodes that perform the visualization encapsulate the rendering functionality in Open Inventor nodes. This scene graph contains the visualization nodes as well as nodes that manage the different viewports, nodes that allow the manipulation of the visualization via user input and nodes that manage overlays that enable annotations on top of the histology slices.

Histostitcher™ Visualization

The visualization component consists of multiple, vertically organized components. A state manager maintains the state of the rendering application and manages the scene graph. The state manager receives the relevant user events from the GUI layer, processes them and forwards them to the XIP scene graph component. Separating the work flow logic from the scene graph reduces the amount of additional Open Inventor nodes dealing with workflow logic, keeping the scene graph simpler and the easier to maintain. This separation of concerns between workflow logic and visualization allows the scene graph to be solely focused on the core visualization tasks, increasing performance and efficiency.

Histostitcher™ Scene Graph

The scene graph itself handles the workflow state dependent viewport layouts for visualizing all, or a limited number, of the images. Furthermore it contains Open Inventor nodes for visualizing the large

images and overlay graphics, as well as the interaction and event handling functionality. The algorithm for implementing the histopathology stitching^[6] is integrated into the new platform. The algorithm is invoked once the user defines the fiducial pairs and the resulting transformation from the algorithm is forwarded to the visualization nodes.

Figure 5 shows a high level view of the current XIP scene graph within the platform. The Open Inventor nodes shown (yellow shapes) correspond to loading the images, editing fiducials, image visualization and overlay graphics rendering, or stitching. This modularity guarantees that in future versions, additional algorithms can be easily added to the application, by either direct integration in source code or by adding new engines and nodes that interact with existing nodes. In addition, the modularity of XIP allows for easy incorporation of algorithms developed using the insight toolkit (ITK)^[14] and the visualization toolkit (VTK).^[15] A summary of the different XIP modules as shown in Table 1.

Table 1: List of XIP™ components used in the histostitcher™ system

Component	Description
Open inventor node	A node is a data container which can contain such things as lighting, or can perform a function such as rotating a quadrant, or calculate the stitched result. A node can be used to notify other nodes when its state changes, such as when a GUI event is triggered
Scene graph	The scene graph contains open inventor nodes for visualization and blending, and a state manager receives events from the GUI, updating the appropriate nodes, signaling when a new rendering is required. In addition, the algorithm for using fiducials to stitch quadrants is contained as a node in the scene graph
XIP builder	XIP Builder is a separate application which is used to construct and debug the scene graph
OpenGL fragment shader	A fragment shader is used to instruct the graphics processor unit how to interpolate and blend tiles to render a view of a histology quadrant, or stitched result

XIP: Extensible imaging platform, GUI: Graphical user interface

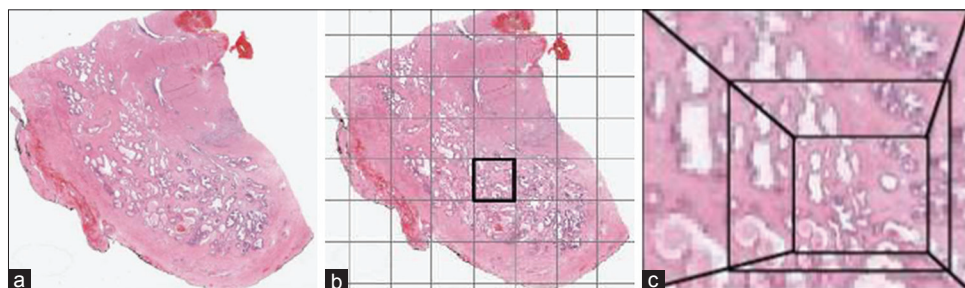


Figure 4: (a) A histology quadrant is loaded; (b) The quadrant is split into multiple unique, non-overlapping blocks; (c) Each block is converted into a multi-resolution pyramid, and stored on disk using joint photographic experts group compression

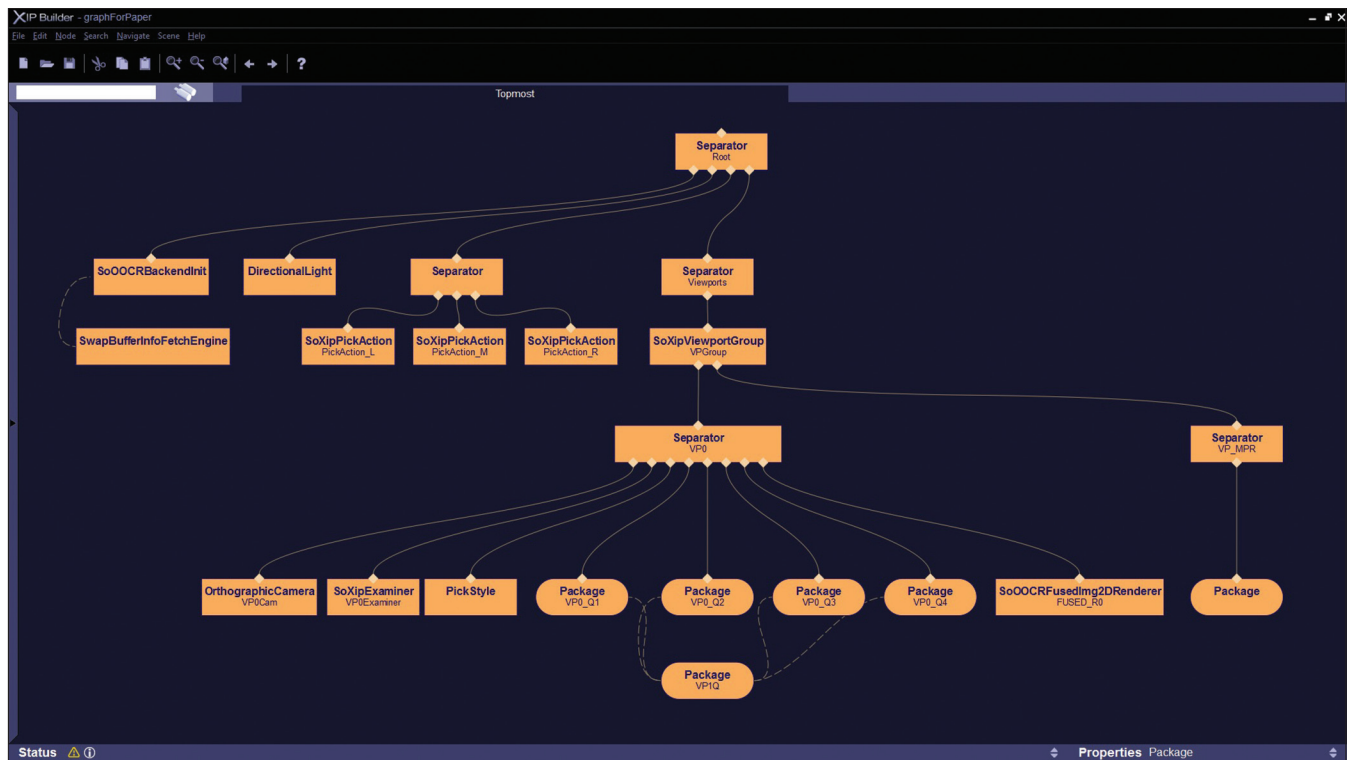


Figure 5:The extensible imaging platform scene graph structure for histopathology stitching views containing individual Open Inventor nodes

Rendering Requirements

The limitations of the prior system^[6] can be attributed to the lack of rendering and interactivity considerations in the fundamental software architecture. Particularly, the system does not leverage the high performance GPU's that are available in media ready PC systems. Graphics processors can deliver extremely high framerate and interactivity, but require specific architecture to support their use. This is especially the case when the typical histology quadrants can have sizes of up to 30,000 by 30,000 pixels. For RGB quadrants, this translates into an uncompressed image size of approximately 2.5 GB per quadrant, more than the available memory on a typical graphics card. Ideally, the display should handle full resolution for large displays at a frame rate of 60 frames per second. In addition, the system is expected to provide smooth panning and rotation as well as seamless zooming between different resolution levels, essentially mimicking the view a microscope would provide. Moreover, up to four quadrants need to be rendered simultaneously, containing up to 10 GB of image data. Displaying this huge amount of image data on a standard PC requires an intelligent caching strategy that supplies the correct information when the user needs it, without time consuming streaming from hard disk storage

Tile Caching of Histology Quadrants

Interactive visualization of the large quadrant images is achieved using tile-based rendering with out-of-core caching and on-demand-loading of requested tiles.

A multi-resolution pyramid is pre-generated when loading the quadrants, which is used to render only the current resolution [Figure 4]. In addition, pre-caching of tiles adjacent to the current field of view is performed to facilitate smooth panning and zooming. For optimal speed, the standard multi-resolution TIFF-encoded.svs files are converted into a tile-based, joint photographic experts group (JPEG)-encoded, multi-resolution image format, developed specifically for this renderer. Each data block is compressed individually, which enables easy addressing of data blocks as well as concurrent processing of these blocks in separate threads. JPEG was chosen due to its fast decompression whereas concurrently providing good image quality. The conversion of the raw quadrants to multi-resolution JPEG tiles is performed off-line during the loading of the images and stored on disk. These files contain tiles of 256×256 pixels, which are recursively downsampled to create the coarser resolution levels

Rendering of Histology Quadrants from Cache

Tiles are requested from the visualization layer and cached in a hierarchy of increasing resolution. All the tiles of an image are available on the hard drive cache, loaded on demand into the main memory cache and then transferred into GPU memory. Only the tiles covered by the actual view are requested by the visualization layer. For a smoother user-interface, certain tiles are additionally preloaded: Tiles that are adjacent to the currently displayed tiles as well as tiles from the coarse resolution levels. Preloading of adjacent tiles in the

hierarchy allows smooth panning, rotating and zooming, whereas preloading coarser visualization levels allows to substitute missing tiles with coarser resolution tiles, until the needed tiles have been loaded into graphics memory.

The renderer is separated into a display thread to handles the visualization and worker threads to concurrently load and transfer tiles between different layers of the cache hierarchy. A synchronization point allows loading the requested tiles from the cache into the GPU texture memory, used to rendering the final images.

Since the multi-resolution pyramid is cached at discrete resolutions, OpenGL fragment shaders are used to interpolate the requested resolution, which may be in between cached resolutions. This enables the visualization layer to incorporate seamless zooming of the histopathology images. Due to cache misses, some of the tiles are not available at the requested resolution and temporarily substituted by lower-resolution tiles that are available in the GPU cache. The fragment shaders interpolate between different resolutions among different blocks for visualization without distracting seam artifacts.

Multi-touch Interface

To further explore improvements in human interactions of stitching workflow, multi-touch interfaces for navigating the large histology images have been explored. The initial implementation of navigation and annotation of large histology images has been explored on both tablet sized, multi-touch devices, as well as on a 27" desktop multi-touch display. Multi-touch gestures such as pinch-zoom, flick, pan and touch-and-hold gestures have been evaluated for viewing and selection of the large histology image. The set of multi-touch gestures incorporate into a lightweight histopathology image viewer to assess the usability. Conventions for image manipulation were deduced from more widely accepted consumer photo viewing applications. In combination with the multi-resolution GPU accelerated rendering, the result is a highly natural interface and smooth interaction, especially when changing between high and low resolution views of the images. The result yields an efficient workflow with broad accessibility. In addition, the intelligently cached zooming ability allows the user to precisely place fiducials within the image space, which overcoming precision issues currently present in a capacitive multitouch surface.

RESULTS AND DISCUSSION

Stitching Results

Pseudo whole mount reconstructions were generated from quadrants acquired via an Aperio Whole Slide Scanner Scanscope CS at $\times 20$ magnification. The best effort was made to maintain the same thickness and height between slices, although minor variations between

images were inevitable. The scaling component of the registration was able to take into account these artifacts. The new *ex-vivo*, *in vivo* histopathology stitching platform has yielded much more efficient pseudo WMHS reconstructions than the previous generation. Firstly, the ability to quickly pre-align the quadrants prior to selecting fiducials has facilitated the process of selecting corresponding edges on the quadrants. The ability to smoothly zoom in on specific regions of the quadrants has helped picking corresponding fiducials tremendously. Figure 6 represents the selection of

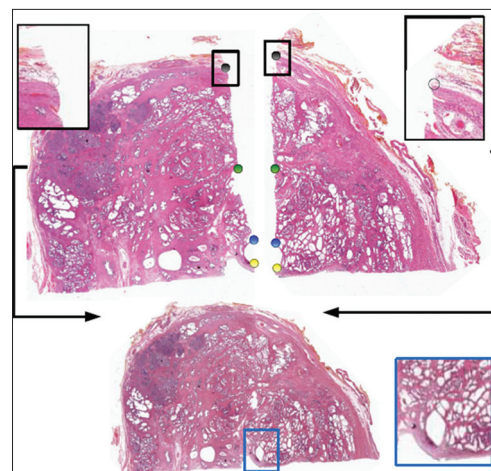


Figure 6: Two adjacent quadrants (top) are rotated and aligned prior to selecting fiducials. The clinician zooms in to select corresponding fiducials on the quadrants, which yields in a final stitched result (bottom). It can be seen that the nodule highlighted in blue was able to be seamlessly stitched

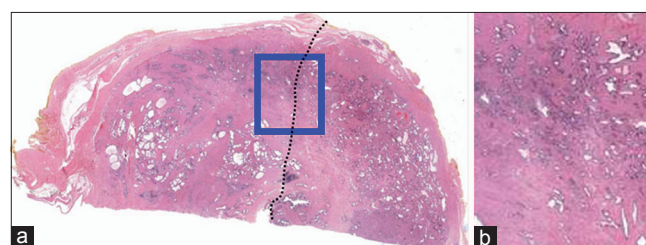


Figure 7: (a) Stitched result from two quadrants, where the dotted black line shows the separation of the quadrants. (b) Shows a zoomed in region of the boundary between the quadrants, where it can be seen that rendering with the extensible imaging platform based Histostitcher™ resulted in a seamless transition between quadrants

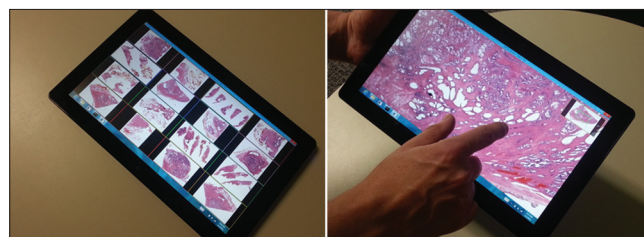


Figure 8: View of the large histopathology image viewing on a multi-touch tablet

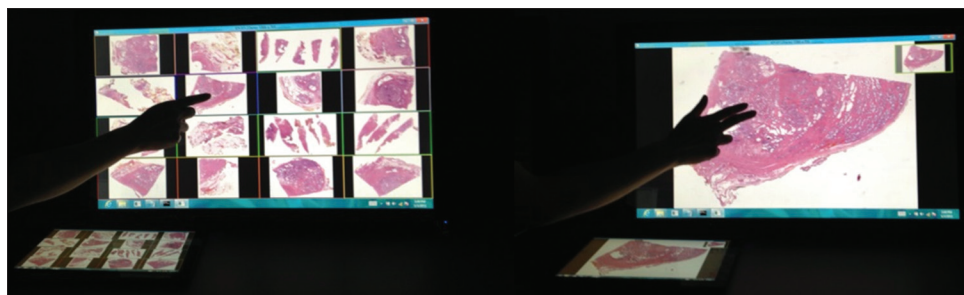


Figure 9: View of interaction with a 27" multi-touch display with the rendering powered by integrated graphics on a conventional tablet device

fiducials on two adjacent quadrants after pre-aligning them and shows the seamlessness of the final stitch. For quantitative evaluation of the stitching algorithm, we refer the interested reader to Chappelow *et al.*^[6] Three independent expert pathologists scored the stitching results using a 6-point scoring system for several reconstructions and it was found that the Histostitcher algorithm quantitatively outperformed manual reconstructions using Photoshop. In addition, the ability to accurately select corresponding fiducials after zooming in has resulted in reconstructions with seamless transitions between quadrants. A seamless transition of a stitched half-slice is shown in Figure 7 where the two original quadrants are indicated by a dotted black line in Figure 7a and the blending of the adjacent quadrants as shown in Figure 7b.

Multitouch Results

The implementation was tested on a conventional Windows 8 Surface Pro tablet device to assess its interactivity of rendering and viewing such large histopathology data [Figure 8]. It was found that interactive frame rates were achieved with the multi-resolution histopathology renderer leveraging integrated graphics hardware available on the Surface Pro. The performance delta between running on a tablet and running on mid-range desktop PCs was small enough to be unobservable; this small performance delta demonstrates the scalability of our multi-resolution image rendering techniques.

The implementation was also tested with the tablet device used as the rendering engine powering a 27" multi-touch display [Figure 9]. There was a little performance impact in driving the larger display. The human interaction on the larger display, due to the size, further alleviates some of the precision issues such as the size of finger touches relative to the image features.

CONCLUDING REMARKS

In summary, we have created an efficient software system Histostitcher™ suitable for routine creation of WMHS, which will be extremely important in the context of clinical trials and large cohort studies where accurate and efficient stitching is required. The new system is accelerated with fast GPU rendering, the

ability to remove or move fiducials and the ability to quickly move and align the quadrants prior to fiducial selection. The core rendering technology is also efficient on newer generations of tablet devices where multi-touch interactions with histopathology stitching requirements are assessed and found to be quite intuitive to use. In addition, the system is highly modular, in that additional algorithms can be loaded as libraries and quickly integrated into the software. The framework showcased here is also being applied in the context of image registration and segmentation of multimodal data.

In addition, the following features fusing MRI with histology data are currently under active development and planned for future versions of the platform. This fusion component is currently being quantitatively evaluated, the results of which will be presented in future work.

- The operator additionally loads one or more MRI scans
- The operator may select any of the MRI scans from a thumbnail bar
- The operator may rotate, pan, zoom the selected MRI scan until a best match image is identified for comparison with the pseudo WMHS
- Fiducials common to the MRI image plane and the pseudo WMHS are placed as pairs to identify common features
- The pseudo WMHS and MRI image plane is registered based on the fiducials based on a thin-plate splines registration algorithm
- The MRI image plane is deformed to match the pseudo WMHS and is displayed as semi-transparent overlay onto the WMHS for comparison
- Annotation as position text markers or polygons can be drawn onto the image display to demarcate regions of interests
- Incorporating the Sciport database system for storing and loading the data, manual tumor annotation from clinicians and automatic image analysis algorithms^[3] loaded through external libraries.

REFERENCES

1. National hospital discharge survey-procedures by selected patient characteristics-number by procedure category and age, 2009. Available from: http://www.cdc.gov/nchs/data/nhds/4procedures/2009pro4_number_procedureage.pdf. [Last accessed on 2013 Dec 17].

2. Chappelow J, Viswanath S, Monaco J, Rosen M, Tomaszewski J, Feldman M, et al. Improving supervised classification accuracy using non-rigid multimodal image registration: Detecting prostate cancer. In: SPIE Medical Imaging. Vol. 6915. Available from: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=828471>. 2008. p. 6915.
3. Doyle S, Feldman M, Tomaszewski J, Madabhushi A. A boosted Bayesian multiresolution classifier for prostate cancer detection from digitized needle biopsies. *IEEE Trans Biomed Eng* 2012;59:1205-18.
4. Xiao G, Bloch BN, Chappelow J, Genega EM, Rofsky NM, Lenkinski RE, et al. Determining histology-MRI slice correspondences for defining MRI-based disease signatures of prostate cancer. *Comput Med Imaging Graph* 2011;35:568-78.
5. Hambrock T, Vos PC, Hulsbergen-van de Kaa CA, Barentsz JO, Huisman HJ. Prostate cancer: Computer-aided diagnosis with multiparametric 3-T MR imaging – Effect on observer performance. *Radiology* 2013;266:521-30.
6. Chappelow J, Tomaszewski JE, Feldman M, Shih N, Madabhushi A. HistoStitcher(©): An interactive program for accurate and rapid reconstruction of digitized whole histological sections from tissue fragments. *Comput Med Imaging Graph* 2011;35:557-67.
7. Samaratunga H, Montironi R, True L, Epstein JI, Griffiths DF, Humphrey PA, et al. International Society of Urological Pathology (ISUP) Consensus Conference on Handling and Staging of Radical Prostatectomy Specimens. Working group 1: Specimen handling. *Mod Pathol* 2011;24:6-15.
8. Shen SS, Truong LD, Ro JY, Ayala AG. Use of frozen section in genitourinary pathology. *Pathology* 2012;44:427-33.
9. Toth R, Chappelow J, Kutter O, Vetter C, Russ C, Feldman M, et al. Incorporating the whole-mount prostate histology reconstruction program histoStitcher(©) into the extensible imaging platform (XIP) framework. *SPIE Medical Imaging*. Available from: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1345988>; 2012. p. 8315.
10. Paladini G, Azar F. An extensible imaging platform for optical imaging applications. In: SPIE 7171, 717108, 717108-1-717108-10. 2009.
11. Compute unified device architecture (CUDA). Available from: <http://www.developer.nvidia.com/category/zone/cuda-zone/>. [Last accessed 2013 Dec 17].
12. Gidxen V, Moeller T, Ljung P, Paladini G. Scene graph-based construction of CUDA kernel pipelines for XIP. *MICCAI 2008 Workshop on High-Performance Computing*; 2008.
13. Wernecke J. *The Inventor Mentor: Programming Object-Oriented 3d Graphics with Open Inventor*, Release 2. 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 1993.
14. Yoo T, Ackerman M, Lorensen W, Schroeder W, Chalana V, Aylward S, et al. Engineering and algorithm design for an image processing api: A technical report on itk-the insight toolkit. In: Westwood E, editor. *Proc. of Medicine Meets Virtual Reality*. Amsterdam: IOS Press; 2002.
15. The visualization toolkit (VTK). Available from: <http://www.vtk.org>. [Last accessed on 2013 Dec 17].