

# HPC Enabled Real-Time Remote Processing of Laparoscopic Surgery

Karan Sapra\*<sup>1</sup>, Zahra Ronaghi\*<sup>2</sup>, Ryan Izard\*<sup>1</sup>, Edward Duffy<sup>3</sup>, Melissa C. Smith<sup>1</sup>, Kuang-Ching Wang<sup>1</sup>, David M. Kwartowitz<sup>2</sup>  
<sup>1</sup>Electrical and Computer Engineering <sup>2</sup>Bioengineering <sup>3</sup>Clemson Computing and Information Technology  
Clemson University, Clemson, SC

ksapra, zronagh, rizard, eduffy, smithmc, kwang, robodoc@clemson.edu

**Abstract**—Laparoscopic surgery is a minimally invasive surgical technique where surgeons insert a small video camera into the patient’s body to visualize internal organs and use small tools to perform surgical procedures. However, the benefit of small incisions has a drawback of limited subsurface tissue visualization. Image-guided surgery (IGS) uses images to map subsurface structures and can reduce the limitations of laparoscopic surgery. One particular laparoscopic camera system is the vision system of the daVinci robotic surgical system. The video streams generate approximately 360 MB of data per second, demonstrating a trend towards increased data sizes in medicine. Processing this huge stream of data on a single or dual node setup is a challenging task, thus we propose High Performance Computing (HPC) enabled framework for laparoscopic surgery. We utilize high-speed networks to access computing clusters to perform various operations on pre- and intra-operative images in a secure, reliable and scalable manner.

**Keywords**—Laparoscopy, Image-guided Surgery, Medical Imaging, High-performance Computing, OpenFlow Network

## I. INTRODUCTION

Minimally invasive surgery (MIS) reduces patient trauma and recovery time through reducing incisions. However, small field of view of the laparoscope and incision size will result in a small visual field of underlying tissues of interest. Image-guided surgery (IGS) uses images to map a surgical region of interest, providing surgeons with visualization of subsurface structures [1]. The accurate co-registration of an IGS system with laparoscopic video allows for a resection with higher specificity [2].

We focus on the daVinci robot, which uses two parallel 1080p High Definition video cameras [3]. These video streams generate approximately 360 megabytes of data per second. Processing the data has become demanding for a local system, and high-performance computing (HPC) hardware is needed [4]. To process this data on remote HPC clusters at 30 frames per second (fps), it is required that each 11.9 MB video frame be processed by a server and returned within 1/30th of a second for the left and right eye synchronization of the daVinci. As a result, utilizing high-speed networks to access computing clusters will lead to real-time medical image processing and improve surgical experiences.

## II. FRAMEWORK OVERVIEW

The overall implementation is divided into: Networking Framework and Computing Framework.

### A. Networking Framework

Steroid OpenFlow Service (SOS) is an OpenFlow-based network service that can seamlessly increase the performance of large data transfers over long-distance and high bandwidth networks [5]. TCP is typically unable to consume the available network bandwidth over large networks and only permits the connection source to send its window-size number of packets before receiving an acknowledgement. In a large network, this can result in the sender waiting idly for the acknowledgement before being permitted to send additional packets. SOS redirects the TCP connection to a local SOS agent using an OpenFlow switch in the network path.

As the source SOS agent (surgeon console) accumulates a buffer of data from the data transfer source, a destination SOS (HPC Cluster) agent is located in close proximity to the intended destination. The source and destination SOS agents communicate and agree to use a number of parallel TCP connections in order to rapidly transfer the data from the source SOS agent to the destination SOS agent. The destination SOS agent collects data from the parallel TCP connections and presents it to the intended destination as if it originated from the source across the large network.

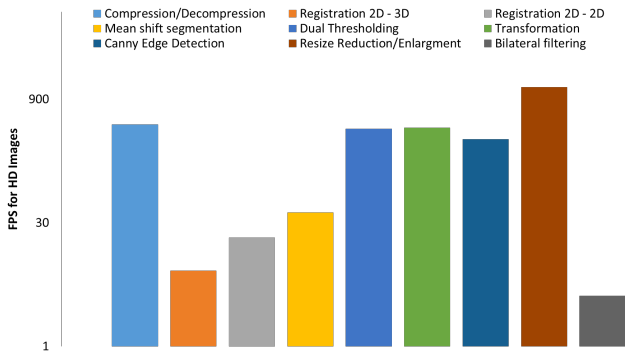
### B. Computing Framework

Our computing framework also enables reliability using replication of computation. Pre-surgery, our setup allows for transfer of per-operative images from a remote location to compute nodes and GPU Memory. We then allow warm-up of the networking protocol described in Section 2.A, followed by verification of compute nodes capabilities and devices.

The compute nodes are broken into three layers: Main Compute Layer (ML) and the two Duplication Compute Layer (DL). Each layer contains node that performs duplicate computation. This computation is then verified with a quick checksum followed by verification using NVIDIA GPU Direct [6] with compute nodes in the two DL. If the ML matches any one of DL calculation, the ML overlay information is sent back to surgeon. In case of a failure of GPU, the computation between the two DL are verified and sent to the surgeon and an idle node in the same layer overtakes the computation task of the failed compute node.

### III. IMPLEMENTATION AND PRELIMINARY RESULTS

We implemented a system on Palmetto Supercomputer [7] using the Dual NVIDIA K40 GPUs per node, CUDA5.5 enabled with Intel Xeon and 32GB of Memory. Figure 1 shows run-time for HD images time of various algorithm of interest to the surgeon in CUDA.



**Figure 1: GPU run-time of image processing algorithms**

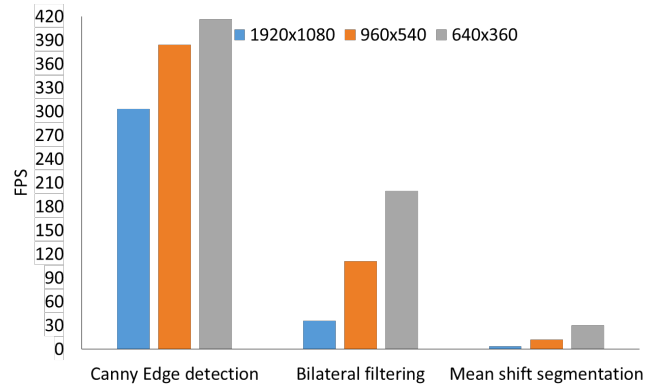
As shown in figure 1, some algorithms have slower frame rates. To achieve a higher frame rate, we will enable GPU image resizing using Bi-cubic interpolation to reduce image resolution prior to the operation flowed by resizing to original image size. In order to reduce and resize to original resolution using GPUs took only 0.8msec. Furthermore if required, we can also enable compression of images using [8] allowing 1.3msec to perform compression and decompression to achieve a higher frame rate.

Figure 2 shows the performance of these algorithms. As seen in the figure certain algorithms such as Bilateral Filtering, Mean-shift segmentation can achieve strong performance gain. However other operations the reduction in image size does not improve the performance.

### IV. CONCLUSION

We present a framework that enables reliable and secure processing of terabytes of information to aid in laparoscopic surgery using GPUS, SOS and Open flow. The output of our framework is an overlay that can be enabled or disabled by the

surgeon at the surgical console. Furthermore, our framework enables replication to prevent failure and allow for scalability for additional operations/algorithms.



**Figure 2: GPU Run-time for varying image sizes**

### REFERENCES

- [1] T. M. Peters, "Image-guidance for surgical procedures," *Physics in medicine and biology* 51(14), R505 (2006).
- [2] D. Van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *International Journal of Virtual Reality* 9(2), 1 (2010).
- [3] D.M.Kwartowitz, M.I.Miga, S.D.Herrell, and R.L.Galloway, "Towards image guided robotic surgery: multi- arm tracking through hybrid localization," *International journal of computer assisted radiology and surgery* 4(3), 281–286 (2009).
- [4] R. Shams, P. Sadeghi, R. A. Kennedy, and R. I. Hartley, "A survey of medical image registration on multicore and the gpu," *Signal Processing Magazine, IEEE* 27(2), 50–60 (2010).
- [5] Rosen, A., & Wang, K. C. (2012). {Steroid OpenFlow Service: Seamless Network Service Delivery in Software Defined Networks}.
- [6] Nvidia GPU-Direct, <https://developer.nvidia.com/gpudirect>
- [7] Cyberinfrastructure Technology Integration (CITI) group at Clemson University, "Overview of the palmetto cluster," <http://citi.clemson.edu/palmetto/pages/userguide.html> (2015).
- [8] Holub, P., Šrom, M., Pulec, M., Matela, J., & Jirman, M. (2013). GPU-accelerated DXT and JPEG compression schemes for low-latency network transmissions of HD, 2K, and 4K video. *Future Generation Computer Systems*, 29(8), 1991-2006.