A Network-Reactive Model for Distributed Telemicroscopy

Wei Lin[1], Joseph Wilder[1], Seth Grossman[2] and David J. Foran[1,2]

[1]Center for Advanced Information Processing (CAIP)
Rutgers University
Busch Campus, Piscataway, NJ

[2]Center of Biomedical Imaging & Informatics (CBII)
UMDNJ- Robert Wood Johnson Medical School
Busch Campus, Piscataway, NJ

ABSTRACT

One of the great challenges in developing Internet-based collaborative systems is that the available bandwidth fluctuates both as a function of time and the number of other users accessing and sharing the resources. While several technologies have been developed to provide a fixed allocation of bandwidth to each user, the focus of our research is to develop a means for dynamically adjusting operating parameters of a distributed telemedicine system in reaction to variations in bandwidth. In this paper we describe the design, development and evaluation of a network-reactive robotic telemicroscopy system, which uses high-speed image processing and Quality of Service (QoS) feedback in order to maximize performance while simultaneously minimizing operational delays. A motion-compensated JPEG strategy is used to provide real-time compression of the color video output of a high-resolution camera interfaced with a robotic microscope. A network subsystem transmits images and responds to user requests via a multithreaded architecture that allows differing degrees of QoS. Variable sub-sampling of images supports the delivery of a QoS level dynamically tailored to the available bandwidth of each user.

Keyword: Telemicroscopy, High-Speed Image Processing, Distance Learning, Video Streaming.

List of Figures

I. INTRODUCTION

Telemicroscopy techniques are being evaluated for a broad spectrum of applications [1-5, 21-23]. While the advantages of real-time video approaches over static strategies, [24-29] have been widely discussed, the high cost and low scalability of such systems has served to limit their use significantly. In recognition of these difficulties, a new generation of designers has attempted to combine the merits of thee two strategies in order to achieve high quality telemicroscopy, which can be implemented using relatively inexpensive telecommunications technology. The new generation of systems typically features remote microscope control as well as the capacity to rapidly resample and transmit discrete images and/or low-resolution video to accommodate a broader spectrum of bandwidths [30-32]. In addition, the new generation systems typically have a greater emphasis on providing an intuitive, collaborative working environment [33,34] in which multiple clients can simultaneously interact irrespective of their choice of computer platform and/or operating system. We present here a network-reactive model for telemicroscopy that uses high-speed image processing and QoS feedback in order to deliver near real-time performance under variable bandwidth conditions in a web-based, multi-threaded collaborative imaging environment.

II. METHODS

A. System Overview

Many telemicroscopy systems utilize multiple processes [1-5] to address video, motion, and robotic functionality. It has been shown, however, that inter-process communication can give rise to an overhead, which is 20 times larger than that of inter-threaded communications [11]. The Network Reactive Telemicroscopy (NRT) server that we have developed exploits the use of a single multi-threaded process on standard workstations capable of handling tens of thousands of packets per second [12]. The NRT server makes use of status flags, which indicate the state of the robotics and the amount of bandwidth available to a given client in order to modify the resolution and number of frames/sec, which are transmitted at any time within an interactive session.

Figure 1 shows the connections among the robotic microscope, server and clients. The server uses an advertised listening socket to detect new clients and utilizes a TCP/IP socket connection to communicate with each. An NT workstation server controls the motorized stage and turret (Prior) of the microscope (AX70 Olympus) through an RS-232 serial line. The video signal from a color video camera (OLY-750) feeds into a Datacube® MaxPCI board installed in the PCI slot of the server.

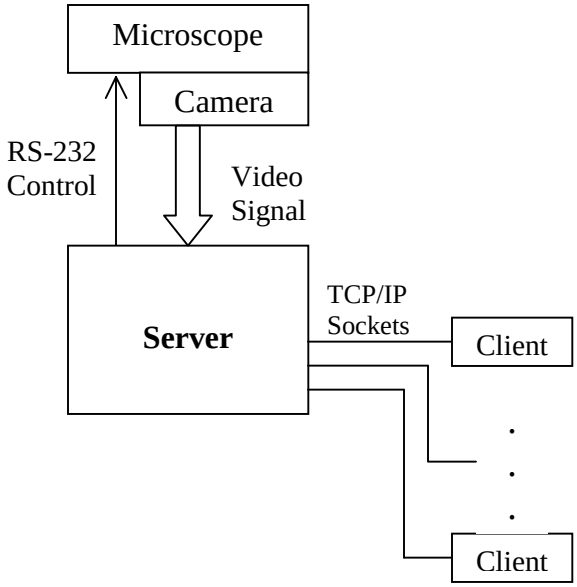


Figure 1. Diagram showing relationship among camera, microscope, server and clients.

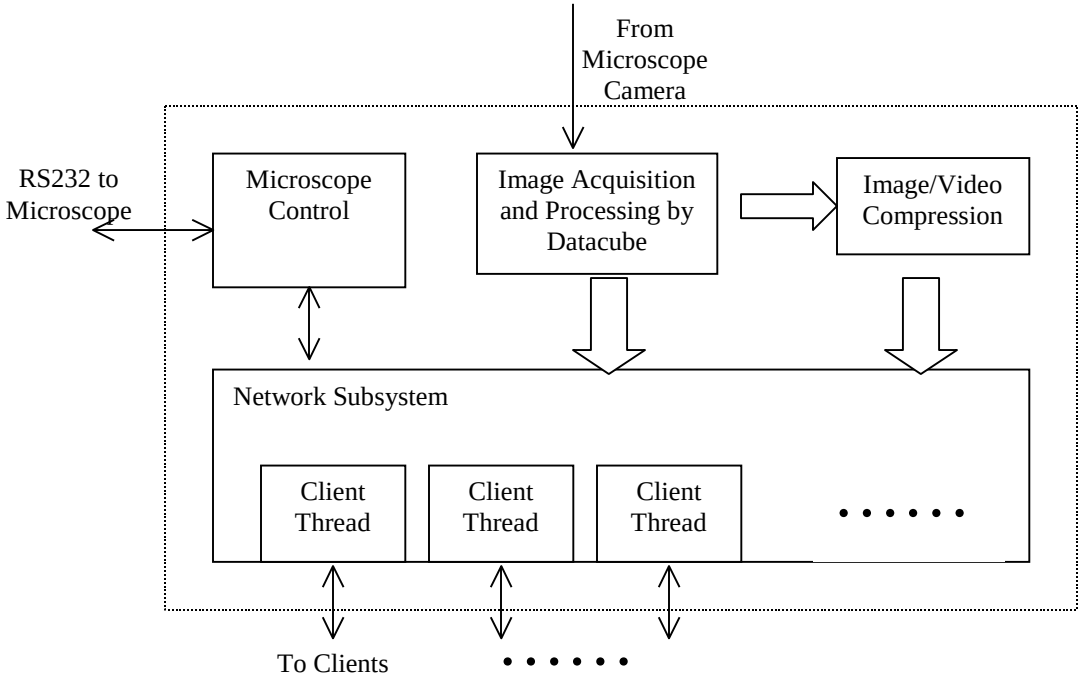


Figure 2. The internal component structure of the server. The components inside the dashed line constitute the server. Thick arrows represent the moving of image/video data inside the server.

Figure 2 shows the internal structure of the server. It is composed of a microscope control component, an image acquisition and processing component, an image/video compression component and a networking component. Each component is implemented in a multi-thread structure, as described in section D.

B. Imaging at the Server

The on-board storage, arithmetic and signaling modules of a Datacube® MaxPCI board were programmed to create interconnecting image processing pipes which allows run-time reconfiguration [6]. Every 33 milliseconds (corresponding to the 30 frames/second frame rate of the camera) both a color image acquisition operation and a gray scale image convolution is performed in order to provide video-rate auto-focusing and image transmission to each client.

The auto-focus algorithm was developed using energy calculations and stage-movement velocity profiles [7, 8]. The algorithm defines the focus function as $f_{OBJ}(z) = E(\delta(I(z)))$, where OBJ refers to the objective lens which is used for a given calculation. Where *I (z)* is the image acquired at vertical stage location *z*; δ is an edge detection operator and *E* is an energy operator. The f_{OBJ} function measures the entropy of the image at a given *x,y,z* stage position. The algorithm systematically examines multiple plane positions until an optimum focus (peak) is located. The δ operator convolves the gray scale image with a pre-programmed 7-by-7 mask thus producing a derivative of the original image. The *E* operator sums the squares of the convolution results at each location. The MaxPCI board performs δ and *E* calculations at each stage position *z*, $f_{OBJ}(z)$, within 33 milliseconds.

The focus function produces a local maximum at the preferred plane of focus, thus moving away from the optimal plane of focus results in a reduction in the f_{OBJ} value. The algorithms set the stage into continuous motion, while computing new f_{OBJ} values every 33 ms and comparing them with prior values. The auto-focus algorithm identifies the optimal plane of focus in a single pass at which point the software directs the robotic stage back to that specific *x,y,z* location and acquires an image. The algorithms simultaneously maximize stage velocity and accuracy. Additional, details regarding these algorithms are provided in the results section.

C. JAVA-based Design

The Network Reactive Telemicroscopy (NRT) system was developed in JAVA with a client/server design. The client is developed as a JAVA Applet so that it can be used with standard web browsers. The NRT system features shared graphical pointers, text messaging, white-boarding, and software tokens that can be passed among session participants to regulate which client possesses control of the robotic microscope. To facilitate navigation about the specimen, the graphical interface of the NRT continuously

displays an updated graphical rectangle to indicate the location of the current viewing field within a mapped (scaled-down) representation of the microscopic specimen. Pull-down menus provide the means to over-ride auto-focusing and for selecting objectives, filters, light intensity, and scaling.

D. Network Subsystem and Thread Structure in the Server

The server provides a separate TCP/IP socket connection to each client that accesses the Network-Reactive Telemicroscopy (NRT) system. The unicast design of the system guarantees reliable delivery of messages among the clients, the server, and the robotic stage and turret. This design also provides a means for establishing independent levels of Quality of Services (QoS), which are customized to each client based upon the bandwidth characteristics of that client's network connection. Each time that a client establishes a TCP/IP connection, the server measures the throughput of the link and assigns it an appropriate QoS level. The server uses a multithread design with separate threads reserved for the Datacube, the main process and each client. Figure 3 shows the relationship among the threads.

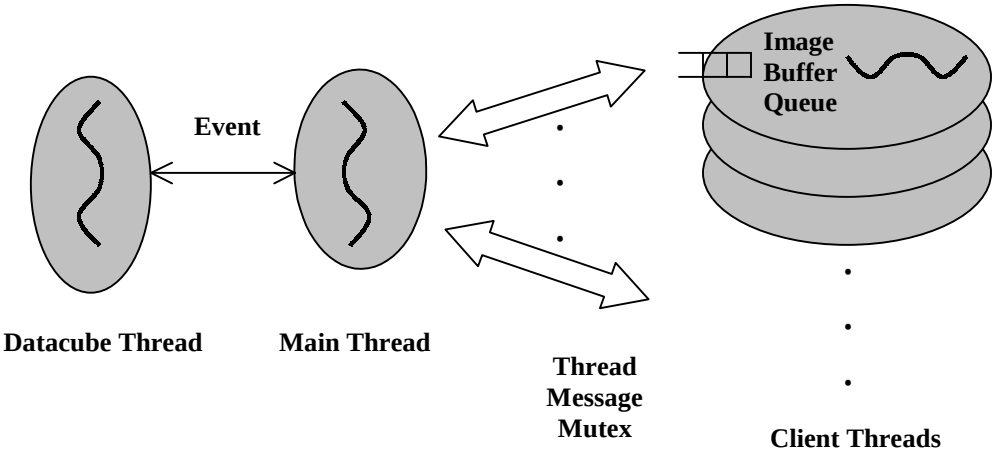


Figure 3. The multi-threaded structure of the server.

The primary advantages of this structure are: (1) by overlapping CPU computations with socket and Datacube board I/O functions operational delays are reduced; (2) by sequestering the behavior of one client from another, system stability is significantly enhanced. So, for example, if one client socket fails, connections between the server and all other clients remain intact. This design also enables multiple users to simultaneously participate in the same video session while the number of frames/second delivered to each client is dictated by the server to accommodate the bandwidth of its specific network link. This is accomplished through the use of a frame-discarding scheme which is discussed later in this manuscript.

All data processing functions are conducted on the Datacube thread whereas functions related to robotic control and image compression are carried out on the main thread. An event object module was

developed to coordinate communication between the Datacube thread and the main thread by examining the signal state of events every 33ms. Whenever the main thread requests an image or convolution result, that event is relayed to the Datacube thread which automatically executes a pipeline operation. Because of the division of Datacube operations into separate threads, other CPU-intensive operations, such as image and video compression, can be conducted while pipeline operations are carried out.

The main thread and client threads maintain two local data structures. The first as an image buffer queue while the second one stores general information including: thread ID's, the current status of the "Microscope Controller" (discussed in the " JAVA Client" section) and the current setting for the microscope objective.

A program object called the Mutex is used to enable multiple threads to share the same resource in synchronized fashion. During video transmissions, the server determines the status of the client socket. If the queue of any given client socket is full, a video frame is skipped. This provides a simple mechanism for automatically adjusting the rate of data distribution according to capacity of each connection.

E. Image Compression and Distribution

Image/Video compression/distribution techniques evolved quite rapidly over the past few years. Emerging standards likes JPEG-2000 [13], MEPG-4 [14,15] and H.26L [16] present unprecedented rate-distortion performance and object-based functionalities which facilitate the network distribution of image/video content. Our work presents an effort to integrate new developments in this area into a telemedicine application.

The Network Reactive Telemicroscopy (NRT) system was designed to meet the demands of two major functions: slide panning and object tracking. Due to the relatively small velocity of live cells, tracking of objects using block-based local motion compensation techniques embedded within MPEG-1 and MPEG-2 followed by the coding of the estimation error (difference between prediction and new frame) for each block usually produces adequate results (e.g. using a typical block-based compression scheme such as JPEG).

In cases where slide panning is controlled by signals from remote locations, however, the motion vector can become quite large. Here the concept of global estimation must be employed [10] by coding the entire image of the first frame and then deriving a global motion vector between the current and the previous frame. The remaining unmatched portion in the new frame is usually several "rows" and several "columns" (see Figure 4). These portions must be coded again and the procedure systematically repeated for each video frame.

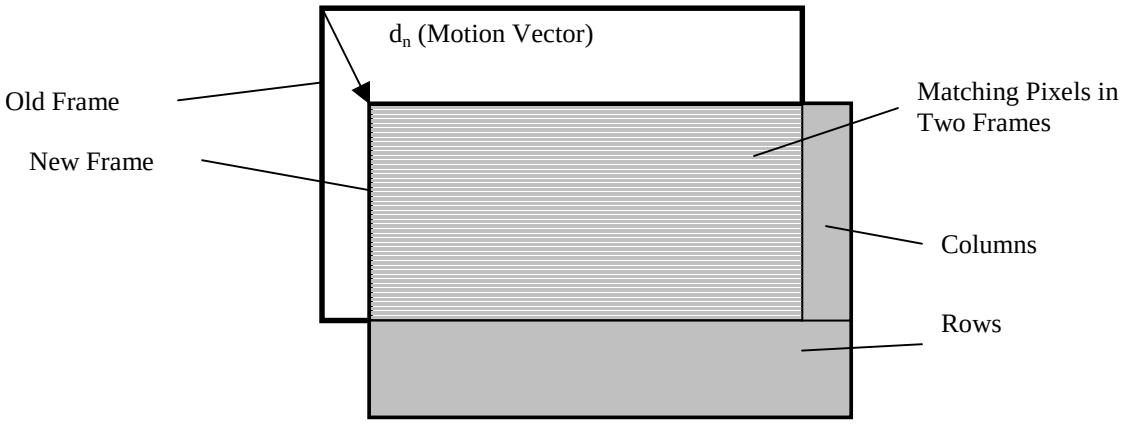


Figure 4. Illustration of motion compensation in the coding of slide scanning sequence.

Since the stage motor motion can be approximated, to the first order, with a constant velocity, the estimated motion vector for frame *n* is

$$\hat{d}_n = d_{n-1} + (d_{n-1} - d_{n-2}) \tag{1}$$

A full search approach is used to derive the real motion vector d_n based on prediction $\hat{d}_n$. To be specific, M (currently M=100) pixels are randomly selected from the old frame, excluding boundary regions (since these pixels may not have a match in a subsequent frame). Then the motion vector is searched in an N-by-N (currently N=16) block centered on the predicted match pixels. The correlation measure to be minimized is

$$E(\Delta d) = \sum_{i=1}^{M} \left| I_n[p_i + (\hat{d}_n + \Delta d)] - I_{n-1}[p_i] \right|^2$$

where I_n represents a new frame, I_{n-1} represents the previous frame, p_i are randomly selected M pixels, $\hat{d}_n$ is the predicted motion vector for the new frame according to the first order model in (1) and Δv lies inside the N-by-N search block. The final motion vector then becomes $d_n = \hat{d}_n + \Delta d_{\min}$, where $\Delta d_{\min}$ minimizes $E(\Delta d)$.

Software implemented JPEG is used to compress video frames prior to transmissions. Based upon our preliminary performance studies, the NRT application requires a sustainable compression rate of about 614,400 (640*480*2) pixels/second, where the current image size is 640 pixels (H) by 480 pixels (W). A standard 450 MHz Pentium II was shown to produce an encoding rate of 3,000,000 pixel/second and a higher decoding rate when Independent JPEG Group's version 6b JPEG codec is used.

III. RESULTS

A. Auto-focus

Due to the bandwidth limitation of most networks, interactive focusing of microscopic specimens is impractical. Therefore, an auto-focusing algorithm was developed for the Network-Reactive Telemicroscopy (NRT) system. Figure 5 shows the focus function that was determined for a 4x objective of the NRT using a routine blood smear specimen. The custom-defined focus function $f_{OBJ}(z)$ reveals a distinct peak. Although it is not strictly symmetrical, it can be roughly approximated by a normal distribution.

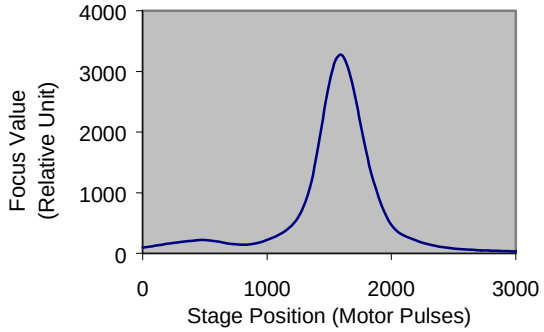


Figure 5. Focus Function $f_{OBJ}(z)$ at the objective of 4x.

Defining the width of the peak, *w*, as the distance between the two sides of the peak at half the peak value, this value is related to the depth of field for each objective. Table 1 lists *w* measurements for the blood smear specimen.

Objective	4 x	10 x	40 x	60 x
w (microns)	200	60	15	10

Table 1. Approximate *w* value measured using a blood smear slide.

While software directs the robotic stage to move the stage along the vertical direction (changing *z*), the Datacube MaxPCI pipeline produces a focus value every 33 milliseconds. To fulfill the real-time constraint of the system, we assume that optimal focus must be completed within 1 second. At 30 frames/second, this corresponds to 30 stage positions on the $f_{OBJ}(z)$ curve. Assuming the peak lies at z_0, if

the focus plane derived by our algorithm has a vertical position $\hat{z}$ in the interval [z_0-*w/6*, z_0+*w/6*], the Normal approximation becomes:

$$f_{OBJ}(\hat{z}) / f_{OBJ}(z_0) \geq 0.986$$

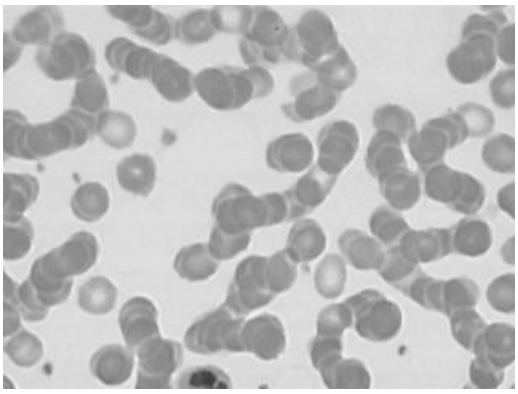
The width of the interval is *w/3*. Assuming that stage movement velocity is constant, the algorithm will locate the optimal focus plane in 1 second so long as the initial position is within the range of *10w* away from z_0. Based upon Table 1, a search range of *10w* would correspond to about 2 millimeters for the 4x objective, which should be sufficient to account for most variations in sample thickness.

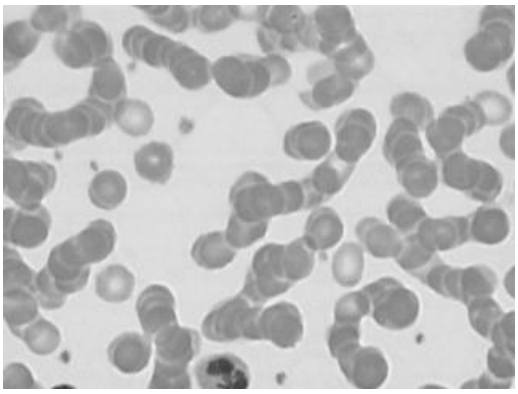
The NRT server auto-focuses at 4x (lowest magnification for the system) any time a new slide is place in the optical path. During the process the NRT continuously updates a file logging the distance between the focus planes of each objective in order to compensate for any possible mechanical drifting that might be introduced into the system.

B. Motion Estimation and Video Compression

Global motion compensation was achieved by computing difference images between adjacent frames followed by concentrating the information contained on unmatched portions of the data (Figure 6). Figure 7 shows a correlation distribution for a range of different motion vectors.

As a result of using the combination of motion estimation immediately followed by JPEG compression, bit rates of 50-200 Kbps were sustained during a series of NRT sessions. Although performance varies as a function of scene complexity, using 5 randomly selected test sequences of video, the compression ratio of raw video sequence versus compressed sequences ranged from 336:1 to 1213:1. The computation time for motion estimation is measured to be negligible compared to the JPEG compression time.





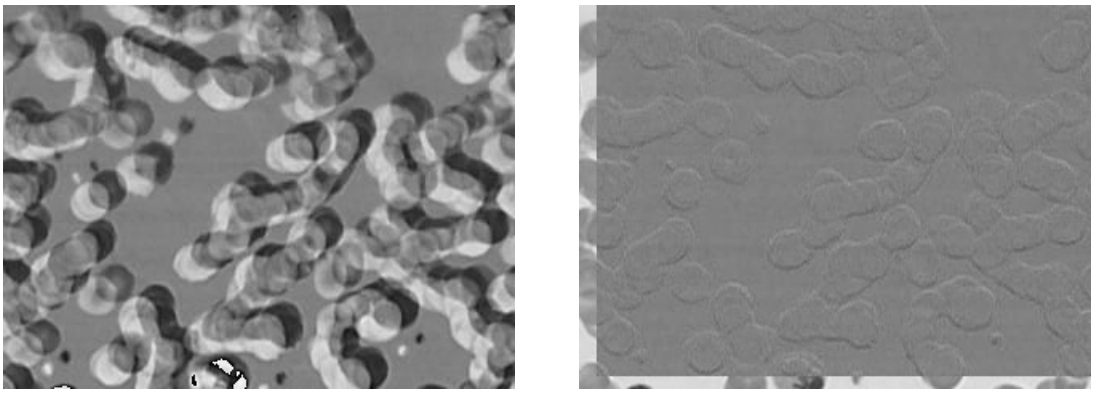


Figure 6. An example of motion estimation. Top two images are from two adjacent frames. Bottom left panel shows the difference image before motion compensation. Bottom right panel shows the difference image after motion compensation. All difference image portions are offset to center at 128.

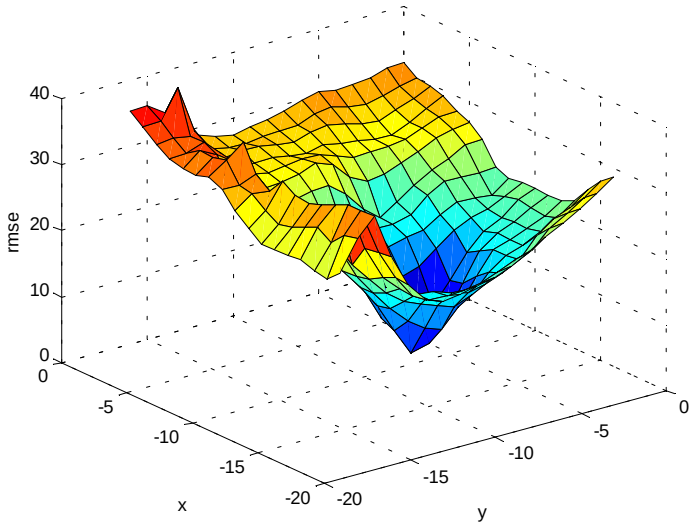


Figure 7. Root mean square error for different motion estimations. Minimum is reached at (-11, -8).

C. Networking Performance

In order to gauge the performance of the NRT system under real case scenarios it was tested in a multi-user environment over fractional T1 LAN service. The NRT server consists of a Pentium 4 Processor, 1.9 GHz, 1 GB RDRAM. This configuration was shown to support at least 25 simultaneous users without significant impact on the operations. System overhead is relatively small, consequently

message processing time for the server and the network transmission delay total less than 100 milliseconds. Please see Figure 8.

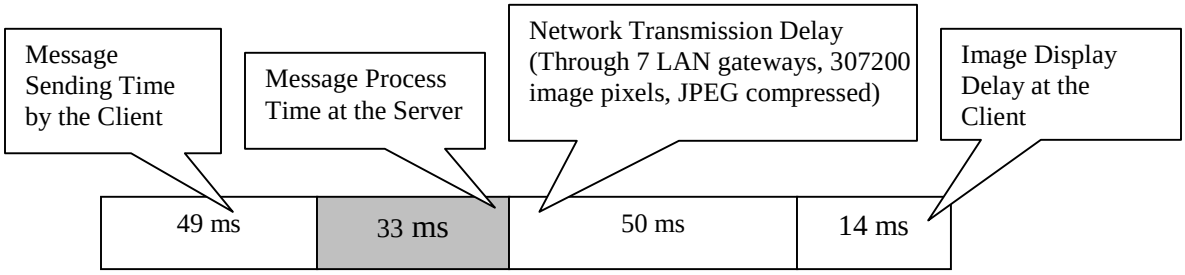


Figure 8. Breakdown of the Delays for an Image Request

The largest operation delay of the remote microscope is experienced when panning a specimen, but even then the delay is measured to be less then 100 milliseconds, over LAN and WAN connections. In order to cope with network jitter, the NRT is equipped with a 300 millisecond playback delay through buffering at the client side in order to provide smooth, video-guided control of the microscope.

Figure 9 shows the server throughput measured over an Ethernet path while acquiring and sending 19.2 KB video frames under two conditions: 1) using a single thread; 2) using a multi-threaded design. In the single threaded case, the total throughput remains almost constant irrespective of the number of clients accessing the server. In fact, it decreases slightly when there are more than 3 clients due to the additional overhead. As a result, each client's bit rate varies inversely with the number of the clients. On the other hand, the throughput for the multi-thread server is able to accommodate additional clients by increasing the throughput as a function of the number of individuals accessing the system. This is accomplished through the use of parallel packet-transmission.

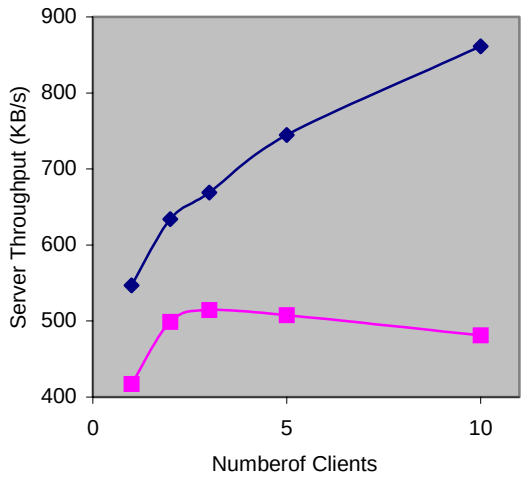


Figure 9. Comparison of the throughput of a multi-thread server and a single-thread server when sending 19.2 KB video frames over an Ethernet link. The upper curve corresponds to a multi-thread server. The lower curve corresponds to a single-thread server.

IV. DISCUSSION

We have presented a robust, network-reactive robotic telemicroscopy system which uses high-speed image processing and Quality of Service (QoS) feedback in order to maximize performance while simultaneously minimizing operational delays. A motion-compensated JPEG strategy is used to provide real-time compression of the output of a color video camera. A network subsystem transmits images and responds to user requests via a multithreaded architecture that allows differing degrees of QoS. Variable sub-sampling of images supports the delivery of a QoS level which is dynamically tailored to the available bandwidth of each user. The underlying model for this system might potentially serve a wide range of telemedicine and distance learning applications. In the next generation NRT system, we plan to integrate an additional multicast channel to provide a scalable image/video service to all clients [9] as illustrated in Figure 11. Using this approach the TCP/IP connection for each client can be maintained to guarantee reliable delivery of messages. The term “messages”, as is used here, may refer to still images or video. In addition, a new UDP/IP multicast channel will be added to the NRT. It will enhance the scalability of the system in two ways. First, the server will only need to send one copy each message, regardless of the number of clients participating in the session. Second, there will be, at most, only one

copy of the message on any network path, thereby reducing the possibility of network congestion. Combined with a scalable image/video coding scheme [18-20], the multicast channel may provide differential QoS to clients on links with differing bandwidths by designating a base-level multicast group address and one or more enhancement-level multicast group addresses [9,17].

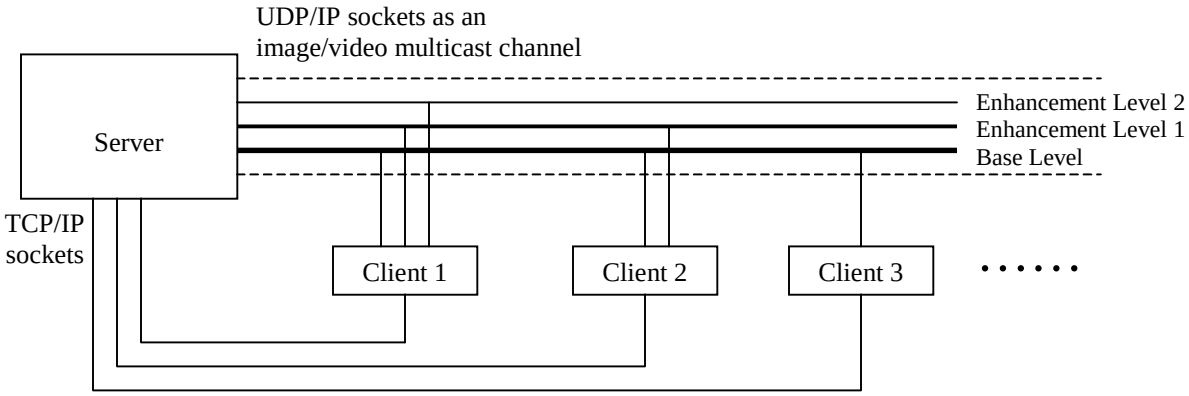


Figure 10. Proposed new system architecture. The added multicast channel alleviates the burden on the network, as well as providing differential QoS to clients. As an example, client 1 receives video with the highest quality and demands a high-speed link. Client 3, on a narrow bandwidth link, can still receive the base level video.

ACKNOWLEDGEMENTS

This work was supported in part through grants from the New Jersey Commission on Science & Technology and by NIH contract 1 RO1 LM07455-01A1 from the National Library of Medicine. The authors would like to thank Marilyn Tremaine for her contribution to the interface design and Hesong Leng for her contribution to the local microscope control design.

REFERENCES

1. B. Parvin, J. Taylor, G. Cong, M.A. Okeffe and M.H. Barcellos-Hoff. “DeepView: A Channel for Distributed Microscopy and Informatics”. *IEEE Conf. on High Performance Computing and Networking*, Nov. 1999.
2. B. Parvin, J. Taylor, D.E. Callahan, W. Johnston and U. Dahmen. “Visual Servoing for Online Facilities”. *IEEE Computer*, July 1997.
3. B. Parvin, D.E. Callahan, W. Johnston and M. Maestre. “Visual Servoing for Micro Manipulation”. *International Conference on Pattern Recognition*, Aug. 1996.
4. M. Hadida-Hassan, S. Young, and et al. Web-based telemicroscopy. *Journal of Structural Biology*, 125:235–245, 1999.
5. N. Kisseberth, G. G. Brauer, B. Grosser, C.S. Potter and B. Carragher. “JavaScope: A Web-Based TEM Control Interface”. *Journal of Structural Biology*, 125:229–234, 1999.
6. Datacube Company. “Advanced Topics in PC ImageFlow: Using the MaxPCI Training Manual”.
7. E. Krotkov. “Focusing”. International Journal of Computer Vision, 1:223-237, 1987.
8. G. Lighthart and F. Groen. “A comparison of Different Autofocus Algorithms”. Proceedings of the 6th International Conference on Pattern Recognition, October 1982.
9. D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, J.M.Peha. “Streaming Video over the Internet: Approaches and Directions”. *IEEE Transaction on Circuits and Systems For Video Technology*, 11:3, 282-300, 2001.
10. Dufaux and J.Konrad. “Efficient, Robust and Fast Global Motion Estimation for Video Coding.” F*IEEE Transactions on Image Processing*. Vol. 9, No. 3, pp. 497-501, March 2000.
11. T. Anderson, B. Bershad, E. Lazowska and H. Levy. “Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism.” *ACM Transactions on Computer Systems*, Feb 1992.
12. Peter Druschel and Gaurav Banga. “Lazy Receiver Processing (LRP): A Network Subsystem Architecture for Server Systems.” *Proceedings of the 2nd USENIX Symposium on Operating Systems Design and Implementation (OSDI),* Seattle, WA, Oct 1996.
13. C.Christopoulos, A.Skodras and T.Ebrahimi. “The JPEG 2000 Still Image Coding System: An Overview.” *IEEE Transactions on Consumer Electronics*. Vol. 46, No. 4, pp. 1103-1127, November 2000.
14. Official Website of the MPEG Committee. http://mpeg.telecomitalialab.com/

15. A. Smolic, T. Sikora and J.-R. Ohm. “Long-Term Global Motion Estimation and Its Application for Sprite Coding, Content Description, and Segmentation”. IEEE Transactions on Circuits and Systems for Video Technology. Vol. 9, No. 8, pp. 1227-1242, December 1999.
16. G.J. Sullivan, T. Wiegand and T. Stockhammer. “Using the Draft H.26L Video Coding Standard for Mobile Applications”. *IEEE International Conference on Image Processing 2001*. Thessaloniki, Greece, October 2001
17. K.-W. Lee, S. Ha, J.-R. Li, V. Bharghavan. “An application-level multicast architecture for multimedia communications.” *Proceedings ACM Multimedia 2000. ACM. 2000, pp.398-400. New York, NY, USA.*
18. W.Lin, J. Wilder and D. Foran. “A scalable Color Image Compression Algorithm for Distributed Diagnostic Imaging Applications”. *Proceedings of the American Medical Informatics Association.* In Review.
19. W.Li. “Overview of Fine Granularity Scalability in MPEG-4 Video Standard.” *IEEE Transaction on Circuits and Systems for Video Technology.* Vol. 11, No. 3, pp. 301-317, March 2001.
20. D.Taubman. “High Performance Scalable Image Compression with EBCOT.” *IEEE Transactions on Image Processing*. Vol. 9, No. 7, pp. 1158-1170, July 2000.
21. Furness P. The use of digital images in pathology. J. Pathol. 1997; 183:253-263;
22. Raab SS, Zaleski MS, Thomas PA, Niemann TH, Isacson C, Jensen CS. Telecytology: Diagnostic accuracy in cervical-vaginal smears. Am. J. Clin. Pathol. 1996; 105(5):599-603;
23. Wells CA, Sowter C. *Telepathology: a diagnostic tool for the millennium?* J. Pathol. 2000; 191:1-7
24. Halliday BE, Bhattacharyya AK, Graham AR, Davis JR, Leavitt SA, Nagle RB, McLaughlin WJ, Rivas RA, Martinez R, Krupinski, EA, **Weinstein** RS. *Diagnostic accuracy of an international static-imaging telepathology consultation service.* Hum Pathol 1997; 28(1):17-21;
25. Eusebi V, Foschini L, Erde S, Rosai J. *Transcontinental consults in surgical pathology via the internet.* Hum Pathol 1997; 28(1):13-16;
26. Singson RPC, Natarajan S, Greenson JK, Marchevsky AM. *Virtual microscopy and the internet as telepathology consultation tools.* Am J Clin Pathol 1999; 111:792-795;
27. Weinberg DS, Allaert FA, Dusserre P, Drouot F, Retailliau B, Welch WR, Longtine J, Brodsky G, Folkerth R, Doolittle M. Telepathology diagnosis by means of digital still images: an international validation study. Hum Pathol 1996; 27(2):111-118;
28. Weinstein MH, Epstein JI. Telepathology diagnosis of prostate needle biopsies. Hum Pathol 1997; 28(1):22-29; Weinstein LJ, Epstein JI, Edlow D, Westra WH. Static image analysis of skin specimens: the application of telepathology to frozen section evaluation. Hum Pathol 1997; 28(1):30-35;

29. Weinstein RS. *Telepathology comes of age in norway.* Hum Pathol 1991; 22(6):511-513]. An initial trial of a prototype telepathology system featuring static imaging with discrete control of the remote microscope. Am J Clin Pathol 1998; 110:43-49
30. Peterson I, Wolf G, Roth K, Schluns K. *Telepathology by the internet.* J Pathol 2000; 191:8-14;
31. Winokur TS, McClellan S, Siegal GP, Reddy V, Listinsky CM, Conner D, Goldnam J, Grimes G, Vaughn G, McDonald JM. An initial trial of a prototype telepathology system featuring static imaging with discrete control of the remote microscope. Am J Clin Pathol 1998; 110:43-49;
32. Dunn BE, Almagro UA, Choi H, Sheth NK, Arnold JS, Recla DL, Krupinski EA, Graham AR, Weinstein RS. *Dynamic-robotic telepathology: department of veterans affairs feasibility study.* Hum Pathol 1997; 28(1):8-12; 21
33. Wolf G, Petersen D, Dietel M, and Petersen I. Telemicroscopy via the internet. Nature 1998; 391:613-614;
34. Burton K and Farkas DL. Telemicroscope: Net progress. Nature 1998; 391:613-614