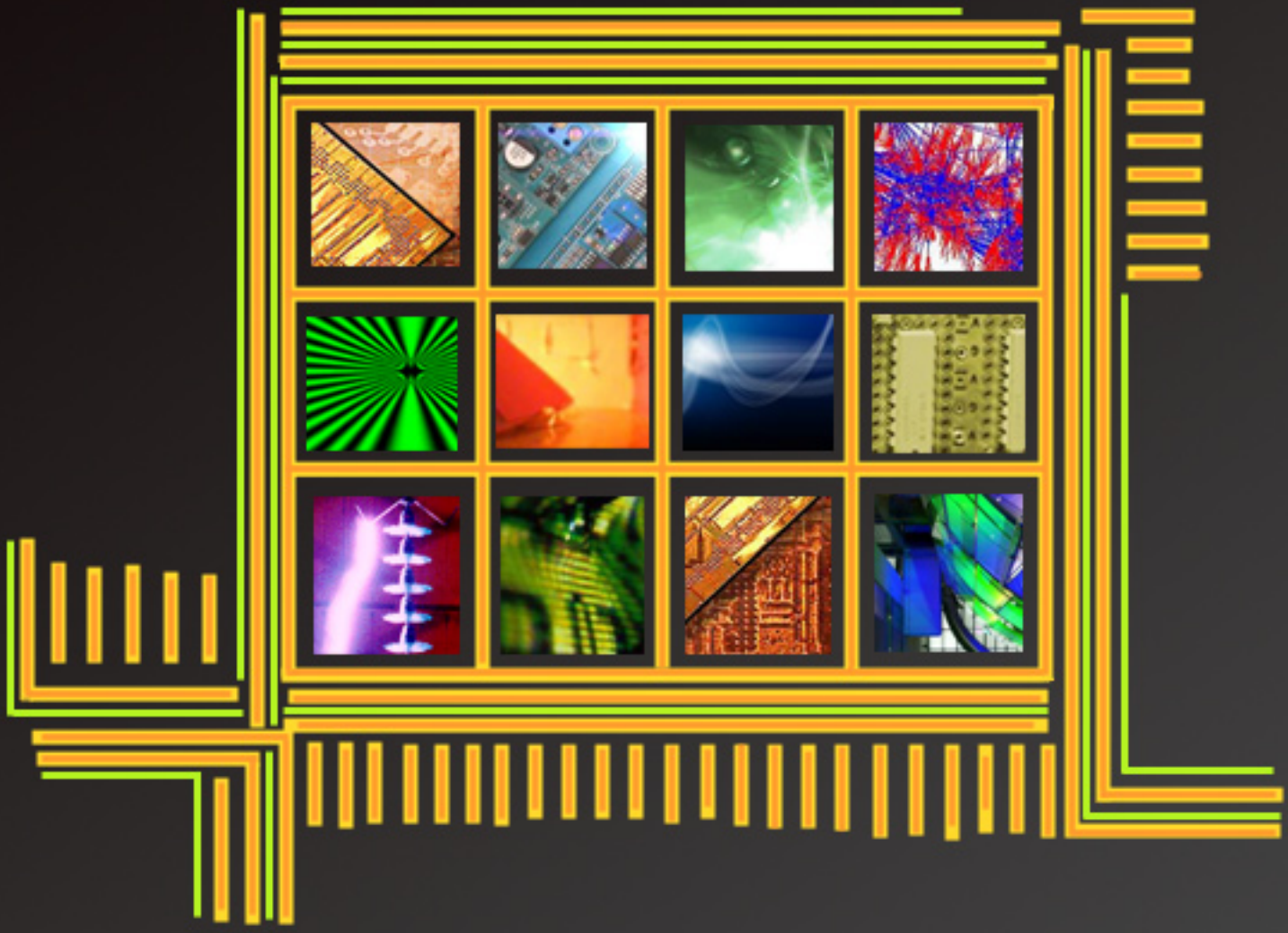
ECJ stanford

Stanford Electrical Engineering and
Computer Science Research Journal

Spring 2006



Faculty Reviewed Groundbreaking Research

A message from the Dean of Engineering

One of the great benefits for undergraduates studying in a research university like Stanford, is the opportunity to participate in leading edge research. While not all students will take advantage of this opportunity, for those that do, the results can be magical. All too often an undergraduate engineering or science education can seem like an endless series of lectures, homework assignments and exams. There is much more to engineering and science than this. Working on the frontiers of a discipline by spending a summer in a faculty research group or by doing a special project with a faculty member can provide tremendous insight into what a career might be like. Sometimes these research experiences result in published work, in technical journals or in magazines like this EE/CS Research Journal. I hope you enjoy reading about the research projects of the students who have written for this magazine. Behind each of the articles is a story of an exciting opportunity to explore the frontiers of a field.

Jim Plummer, Dean
School of Engineering

Contents

The ECJ showcases the top and latest research being done at the union of electrical engineering and computer science disciplines by Stanford graduate and undergraduates.

ECJ is a faculty reviewed technical journal available both online and in print, serving to create awareness across disciplines within the campus technical community, prepare current researchers for the publication and review process of larger journals, and inspire further research by Stanford community members.

the Staff

Sponsors

Editors' Pieces

ECJ Staff

Abstract. Research is often a very distant concept to engineering undergraduate students. Immersed in problem sets and labs, undergrads sometimes lose touch with what is *interesting* and why they chose to study engineering in the first place. The following articles presents what we believe to be relevant and, hopefully, refreshing. Enjoy!

"Being Undigital":Rethinking Photography with Computational Photography

Amal Dar Aziz

ART and science: two completely dissimilar fields? Though many readers may initially respond "yes," in actuality, researchers at Stanford and beyond are making great strides in a field that applies the best of both worlds. In 1995, Electrical Engineering Professor Steve Mann from the University of Toronto, and Professor Rosalind Picard, from the Department of Media Arts and Sciences at MIT, published an article entitled "Being Undigital: The Wyckoff Principle." The principle is named after the inventor of extended dynamic range film, Charles Wyckoff. The article was one of many works that opened readers' eyes to a new form of image processing. It is founded on the Wyckoff Principle, an idea based on the fact that images of the same scene with varying exposures capture different information. Why "undigital?" Read on to learn more.

Fig. 1. Scene captured with regular exposure (top right) and intrinsically relighted (bottom left)

Today, avid photographers and programmers alike are using computers to develop exciting functions for a relatively new application of computer graphics in photography: computational photography. Computational photography combines an understanding of light fields (functions that describe the amount of light traveling through every point in three-dimensional space in every direction), photography and high-performance imaging, and strives to overcome the traditional limitations of a camera. It represents a convergence of photography, computer vision, image processing and computer graphics. With contributions from individuals across diverse fields, computational photography offers insights and functions for the computer scientist and artist alike.

Also termed as *intelligent image processing,* computational photography is different than today's more common and accessible digital photography. Digital photography is rooted in film photography, and offers the same basic tools photographers have been using for decades. The physical limitations of the camera, restricted by its dynamic range and depth of field can be surmounted with the introduction of novel techniques introduced by computational photography, from modifying imaging parameters during capture, to modern image reconstruction methods using captured images. Whereas digital photography is concerned with finite quantities, i.e. pixel quantities that are usually integers, computational photography requires the use of much more precise quantities. Floating point numbers are used when the image is processed, stored, and displayed. The use of real numbers as opposed to integers allows for greater tonal management, motion magnification and enhanced resolution.

The applications of computational photography are vast. For example, even the seasoned veteran photographer often underexposes or overexposes images, resulting in a loss of contrast and detail. In the images below (Fig. 2), we can see the effects of overexposure and underexposure. The left, overexposed image

Fig. 2. Effects of overexposure (left) and underexposure (right)

shows shadow detail, yet the people in the background are completely washed out. In the case of the right, underexposed image, highlight details are enhanced and we can see our subjects clearly, yet detail of the brick interior is lost. Combining the two images, as is the case for Fig. 3, allows us to see our subject, capture shadow detail, and see enhanced highlights of the scenery.

With new technology, computer image processing takes multiple images of the same subject - overexposed, underexposed and with various other lighting schemes, to form an aggregate image or image sequence. In the case of tonal management, an image produced without flash can be used to "relight" a flash image, resulting in an image that depicts objects in the scene using original lighting and shows details of the scene with a realistic level of sharpness and correct colors according to the scene's lighting scheme. Are there other ways to better reveal shape to human views? This is a question researchers developing

Fig. 3. Combining overexposed and underexposed images

new methods in the field of computational photography constantly ask themselves and hope to answer.

The process involved in computational photography greatly differs from traditional photography. Light rays pass through a lens. According to the traditional digital photography model, information is then processed in a detector component which converts the information from light rays into pixels, and then converts it into an image. In the case of computational photography, many more steps and processes are involved. Light rays reflect through a generalized optics instrument and are processed through a sensor using multiple computations. The rays are finally reconstructed and the image is projected. Due to the complex computations required, there is a growing need for new sensors that do more than simply capture a scene with a regular grid of pixel intensities. Computational photography requires the use of multi-dimensional arrays to capture various components of a scene. Computing is much faster and cheaper than precision equipment currently available, and makes use of low-cost computing and storage computer-driven devices. Computational photography represents a great leap towards a more machine-readable form of visual media.

A new idea driving computational photography is ray reconstruction. Using gradient fields captured during computations, reconstruction is made possible, allowing for the implementation of several applications such as image fusion, editing and image matting. New research in the field of computational photography provides opportunities to compute new views of a scene from pre-existing views without the

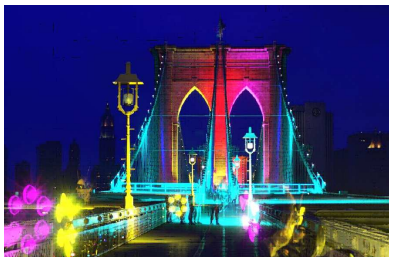


Fig. 4. Lightvector painting of Brooklyn Bridge

original photo subject. The image above is of the Brooklyn Bridge, taken by Steve Mann and used as the cover page of his textbook, "Intelligent Image Processing." It is an interesting, visually explosive example of computational photography as an art form. For visual media applications, these new technologies provide the tools necessary for enhanced, versatile, and aesthetic visual experiences. At Stanford, researchers such as Professors Marc Levoy, Pat Hanrahan, Mark Horowitz, and visiting Associate Professor Hendrik Lensch are engaged in computational photography research and related fields. Interested in computer graphics after taking an introductory course, and an avid photographer, I first learned of the field after speaking with Professor Lensch about current research in computer graphics. Exploring the Graphics Lab in the Gates building is itself a treat, as its walls of images, illustrating the progress of research, greet and educate students passing through. Other institutions, such as MIT and CMU, are also currently conducting cutting-edge research in the field. Perhaps in the near future, the average amateur photographer like you and I will have access to this new technology, and photography will truly become "undigital."

Sources for images

1. Image published in Elmar Eisemann and Frédo Durand's article, Flash Photography Enhancement via Intrinsic Relighting, SIGGRAPH Volume 23, 2004.
2. Series of black and white photos: Prof. Steve Mann's page: http://wearcam.org/immedia/manncomp.htm.
3. An example of Prof. Steve Mann's "Wearable Computational Photography" lightvector paintings.

Cool Technologies and News Flashes

Tom Wang and Amy Wu

Research in Motion - Blackberry

RIM - The Abridged Company History

RESEARCH in Motion, founded in 1984 by University of Waterloo graduate Mike Lazaridis, aspired to be a company that would design innovative technologies and product lines to be used by larger companies.

RIM got its start with a contract from General Motors to create a data display system for an assembly line. Years afterward, it received a contract from US Robotics to work on wireless data modems, the technology that has now become the distinguishing feature of the company. RIM focused its innovation on the wireless domain, coming up with new ways to interact with wireless data, leading to the development of an interactive pager in the mid-1990s.

Enter the Blackberry

RIM's capstone wireless device, the Blackberry, launched at the turn of the millennium and quickly gathered a devoted "Crackberry" following. As reported by USA Today, CEO of Salesforce.com, Marc Benioff, refers to the Blackberry device as the "heroin of mobile computing."

Key Features

- Blackberry users can not only send and receive emails through wireless Internet networks, available in most

metropolitan areas, but can also synchronize the device with multiple email accounts on a remote computer.
- Blackberries are equipped with a keyboard small enough to type on with just one thumb as well as a large brightly lit display, perfect for browsing the web.
- The latest models of Blackberries have GPS-enabled mobile phone and walkie-talkie features as well as location-based tracking.
- Even when offline, Blackberry users can enjoy the standard features of organization and draft composition offered by most conventional PDAs.

In The Future

With the recent launch of Microsoft's intensely marketed UltraMobile PC and the growing popularity of Tablet PCs, sophisticated email-capable cell phones, and paper-thin laptops, the Blackberry has become only one of many options in the arena of the lightweight all-in-one remote computing device. However, it is important to note that the Blackberry remains an innovative driving force in remote computing and its influence will continue to shape the direction of the industry for years to come.

Transparent Organic LEDs

Fraunhofer Institute for Applied Polymer Research

In December of 2005, the Fraunhofer Institute for Applied Polymer Research announced a new display technology known as Transparent Organic Light Emitting Devices, or TOLED for short. This technology builds upon prior Organic Light Emitting Devices, which are often used as screens for cell phones and MP3 players.

Traditional OLEDs can be brightly lit but are usually mostly opaque; TOLEDs incorporate light-emitting polymers to selectively allow amounts of light to pass through the display. Armin Wedel of the Fraunhofer Institute explains: "We achieved this result by using a new type of metal electrode to supply the polymer film with electric current. The clue of the transparency lies in its physical properties."

Essentially, transparent LEDs are now a reality, enabling users to view not only the LED foreground but also the background behind the display.

Universal Display Corporation

Announced almost at the same time, Universal Display Corporation details its own TOLEDs as a series of layers, or an "organic stack," sandwiched between a transparent cathode and a transparent anode. The TOLED is built on the bottom substrate, which can range from from semitransparent glass to opaque plastic depending on the desired transparency of the display. Universal Display claims that TOLEDs can achieve transparency of up to 70 to 85% and can even be built on flexible surfaces such as metallic foil.

Future Technology Applications

Though TOLED technology is still in its nascent stage, innumerable practical uses exist for such displays. One idea proposed is the new ability to convert the flat surfaces of commonplace items and furniture into displays; for starters, imagine a display on the door of a refrigerator to view the items within.

In the field of computer-aided driving, TOLEDs can be used as heads-up displays on automobile windshields to display various dashboard statistics, weather conditions, and directions – all the while maintaining semi-transparency. Taking advantage of the flexibility of TOLEDs, advances can be made in developing electronic papers and newspapers.

Since TOLED technology is only a recent development, we can expect a myriad of practical applications in the future as well as further research in the field.

Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) is a new way to track and identify items using tags. RFID tags are microchips that can be attached to virtually anything, sending and receiving messages to and from RFID transceivers.

There are three types of RFID tags: passive, semi-passive, and active. Passive tags do not need a power supply, and are instead powered by incoming signals. By using a small battery, semi-passive tags are able to receive incoming signals sent without power, making them faster and stronger than passive tags. Active tags are more like beacons – they send out their own signals and have more memory and range than the other types of tags.

Here's how a typical RFID system might work: First, the RFID tags are attached to specific products, and are set to store information that identifies the product such as a serial number. Next, the tag, along with an attached antenna, can send information to a reader. The reader then collects and processes the incoming information to pass on to computers.

Some innovative uses for RFID systems include:

1) Badges and access control
2) Inventory control
3) Equipment/personnel tracking in Hospitals
4) Product tracking through manufacturing
5) Water analysis
6) Asset tracking
7) Livestock tracking
8) Sports events timing

Sources

http://www.wikipedia.org
http://www.korteks.com/Applications/Applications_TOC.htm
http://www.rfidjournal.com

Nanocrystals - Quantum Lightbulbs

Quantum dots, which have been used to produce different colors by changing the size of its nanocrystals, could now replace light bulbs as a light source.

This technology, discovered by researchers at Vanderbuilt University, starts with an LED that produces an intense blue light.

The LED is then coated with a thin layer of special microscopic beads called quantum dots. With this layer, the LED is able to give off a warm white light, similar to that of an incandescent lamp.

Unlike light bulbs however, quantum dots do not give off infrared radiation and are thus more efficient. Although more expensive, they can give off twice as much light per watt and can last up to 50 times as long as a regular incandescent light bulb!

The use of quantum dots is not a new concept, but this is the first time they have been reported to give off white light without any chemical processing. With current processes, less than an hour is required to manufacture quantum dots, and they will be considerably easier to use in making light sources powered by electricity. The key to producing white light is to shrink the nanocrystals down to a specific experimentally determined size, allowing the light emission to move to the surface of the crystal, thus creating a full spectrum of color.

This unusual property of quantum dots was actually discovered unintentionally by a graduate student, after being asked by a coworker to make batches of smaller-sized quantum dots.

Sources

http://www.vanderbilt.edu/exploration/stories/quantumdotled.html
http://eetimes.com
http://sitemason.vanderbilt.edu/files/e9Ldvi/HybridquantumdotLEDweb.jpg

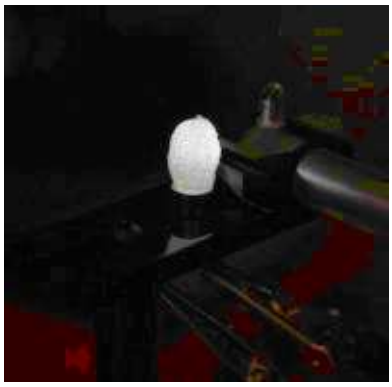


Fig. 5. An example of quantum dot LED

Electronic Paper

E-Paper is a technology that allows text to be displayed on a flexible piece of "paper," which is actually made of organic electronics that uses plastic containing balls that respond to electric charge. It produces no backlighting, which when found on monitors, hurts human eyes.

E-Paper was originally developed at Xerox PARC, and first sold in Japan by Sony starting in 2004. However, Sony has now decided to open up sales to the US.

Since then, Fujitsu also announced the launch of a flexible electronic paper, claiming to consume less power and require no electricity except when changing the screen content. As such, this type of paper is thought to be more suitable for advertisements.

Sources

http://cache.gizmodo.com/gadgets/images/epaper_250-1.jpg
http://www.wikipedia.org
http://www.letsgodigital.org/images/artikelen/68/fujitsu_electronic_paper.jpg

Fig. 6. Fujitsu's electronic paper

iRobot

You may have heard of the iRoomba, which is a small round vacuuming robot. It is actually produced by iRobot, which has a line of products for both home and government use.

For home use, iRobot produces the iScooba, which is the mop equivalent of iRoomba. It also uses the key Aware Robot Intelligent System, which uses sensors to monitor the robot's environment. The robot's behavior is then adjusted up to 67 times a second.

For government and industrial use, iRobot produces the Pack-Bot and the R-Gator. The Packbot is a series of small unmanned ground vehicles that can enter dangerous or inaccessible areas and take footage or dispose of explosives. The R-Gator is an autonomous unmanned ground vehicle that can act as a scout, a guard or a supply carrier.

iRobot was originally founded in 1990 by roboticists from MIT.

Sources

http://www.irobot.com/images/irobotdynhp/packbot-eod-mid.jpg

Fig. 7. iRobot product

Interview with Authors

Jacob Chakareski, Aaron Flores, Ryan Wisnesky

Abstract. Scholars from different walks of life contribute to our campus' EECS community. While their academic interests may be different, students and faculty alike share the element of passion for their work. ECJ had a chance to interview three of our contributing authors about their research and the path they took towards producing Stanford faculty-approved ECJ publications.

Jacob Chakareski

Jacob Chakareski was born in the small town of Struga on the coast of the beautiful and antique Lake Ohrid in the southwestern part of the Republic of Macedonia. At present, he is a postdoctoral researcher in the Signal Processing Institute at the Swiss Federal Institute of Technology (EPFL) in Lausanne. Chakareski received a B.S. degree from Ss. Cyril and Methodius University, Skopje, Macedonia, in 1996, an M.S. degree from Worcester Polytechnic Institute, Worcester, MA, in 1999, and a Ph.D. degree from Rice University, Houston, TX, in 2005, all three in electrical engineering. He performed his doctoral thesis research at Stanford University, Palo Alto, CA from 2002-2005. He received the best student paper award at the IS&T/SPIE VCIP 2004 conference. He has held industrial research positions with Microsoft and Hewlett-Packard. Dr. Chakareski has co-authored over 50 international publications and has two pending patent applications. His fields of interest include networked media systems, distributed computation and control, and wireless communications. In his spare time, Chakareski researches Macedonian history and practices weight lifting and fitness.

Q: How did you pick your research? What is interesting about it?
A: I selected the research that the paper is about after doing an internship with Microsoft Research in Redmond, WA in 2001. Eventually this research grew into a full blown PhD thesis, which I did at Stanford from 2002-2005. I find the subject of networked media systems quite interesting to explore as it combines different areas such as signal processing and coding, computer networking, as well as optimization and control theory. Moreover, digital multimedia has become a common part of our daily lives, a trend that will only increase in the future. Therefore, efficient and adaptive networked media systems have also a very practical importance.

Q: How would you entice undergraduates towards your research field?
A: I would invite undergraduates to attend interesting courses/seminars/talks on multimedia systems that could attract their attention and interest. In addition, I would encourage them to participate on research projects in this area as it would expose them to an experience that is not encountered when attending a lecture. This is particularly rewarding in the case when a project results into a new algorithm or technique that improves a multimedia system in terms of its performance. Seeing yourself being part of a team that delivers a new and improved technology can be quite motivating for further explorations. Finally, I would tend to expose the students to the latest technological advances in the area, as their appeal can be another motivating factor to pursue research in multimedia systems.

Q: Could you sum up using four sentences, in layman's language, what your paper is about?
A: The paper considers a scenario where there are multiple path/routes over which a sender and receiver can communicate in the Internet. The sender is a media server that has a collection of packets comprising a multimedia presentation that need to be sent to the receiver (a media client), where each of the packets has an individual deadline by which it needs to arrive at the client. The paper proposes a framework for organizing the transmissions of the media packets over the multiple routes such that the quality of the media presentation at the client based on the received packets is optimized (maximized). The framework respects the constraints imposed by the network routes on the media packets sent over them in terms of data rate, random packet loss and random packet delay.

Q: Where do you see yourself in 5 years, 10 years?
A: I still see myself as a researcher in 5 and 10 years from now, whether in an academic or a corporate environment. Exploring and discovering new things, something that has not been done before, has always been fascinating for me. The collaboration aspect of doing research is another strong motivating factor for me.

Aaron Flores

Hailing from Mexico, Aaron Flores received his bachelor's degree from the Monterey Institute of Technology, M.S. from Stanford University, and completed his Ph.D. in Winter '06 from Stanford all in electrical engineering. His areas of interest include Statistical Signal Processing, Statistical Learning and Pattern Recognition, and Biomedical Engineering; he is also becoming more familiar with Finances and Data Mining . Flores was the president of the Stanford Mexican Student Association in 2001. In his spare time, Flores enjoys electronic music, playing soccer and video games.

Q: How did you pick your research? What is interesting about it?
A: When I was the TA of Digital Filtering, Adaptive Signal

Processing, and Adaptive Neural Networks, my responsibility included the supervision of several research projects. From such work I became knowledgeable in the LMS algorithm to which I dedicated the last three years of my graduate research. What is cool about my research is that it shows a very simple yet insightful analysis on the performance of the LMS algorithm, task that has been pursued for more than 40 years already.

Q: How would you entice undergraduates towards your research field?
A: The field of adaptive signal processing is increasingly important as digital signal processing is applied in more and more areas (an important example being digital communications). The mathematics and statistical background needed for this field also provide the tools needed to spread into related fields such as statistical learning, which is gaining increasing significance in today's information based society.

Q: Could you sum up using four sentences, in layman's language, what your paper is about?
A: The LMS algorithm is widely used in a variety of applications ranging from acoustic noise canceling to adaptive control. Its performance can vary greatly depending on the environment where it is used, so predicting the algorithm is important in practice. The paper shows a simple way of doing this when certain information about the specific application at hand is known in advance. The results of the paper enjoy simplicity and offer insight into the behavior of the LMS algorithm which is of theoretical interest.

Q: Where do you see yourself in 5 years, 10 years?
A: In 5 years I see myself having a significant role as a researcher in a high tech company. In 10 years I expect to go back to academia, along with consulting to industry on issues of my specialty.

Ryan Wisnesky

Ryan Wisnesky is a co-terminal student in the Computer Science department interested in programming languages and systems. Next year he will be entering Harvard's Ph.D. program in Computer Science.

Q: How did you pick your research? What is interesting about it?
A: This idea came from my mentor at IBM during a summer internship. Grid computing can be a powerful and cost-effective way to deploy software as a service, so it has a lot of relevance to emerging technologies like Web 2.0 and utility computing.

Q: How would you entice undergraduates towards your research field?
A: I think the hardest part is communicating to undergraduates that research can be fun and interesting. Especially early on, it is often far more lucrative to do an industrial internship than spend a summer doing research, and so students that might enjoy research simply aren't exposed to it. Research-oriented courses like CS191W help a lot, but approaching a faculty member for supervision can be daunting.

Q: Could you sum up using four sentences, in layman's language, what your paper is about?
A: A grid is a large collection of computers in which each computer may have different capabilities. When we give a grid work to do, we would like some assurance that the work will be done in a reasonable amount of time. This requires estimating how long the work will take when it is run on different sets of computers in the grid, so that the work may be assigned to computers in an effective way. This paper examines how incorrect estimates of task duration affect how sure we can be that the work will be finished on time.

Q: Where do you see yourself in 5 years, 10 years?
A: In five years, I'll hopefully be finished with graduate school; beyond that, I have no idea.

Image-Enhanced Endoscopy: Implementation, Calibration and Accuracy Testing

Michael R Bax, Rasool Khadem, and Ramin Shahidi

Abstract. A surgical navigation system called *image-enhanced endoscopy* is described. It enables surgeons to explore the underlying layers of the patient's anatomy through a surgical endoscope with the endoscope display overlaid in real time with volumetrically-reconstructed three-dimensional preoperative diagnostic images of a localized area of a patient's anatomy. The new system also allows the surgeon to virtually "fly" through and around the surgical site to visualize several alternatives and qualitatively determine the best surgical approach using the previously acquired magnetic resonance or computed tomography image data. This represents a marked improvement over conventional image guidance systems, which display only two-dimensional reconstructed images based on a single point in the surgical volume. The endoscopes are tracked using infrared diodes and stereo cameras during surgery, and calibration of the endoscope is necessary to ensure that the mapping of the real endoscope image to its virtual counterpart is accurately performed. Extensive accuracy testing of the calibration technique has been performed using a dedicated testing apparatus, and the results are presented. Two approaches to compensating for lens distortion without significant computational impact are presented.

Index Terms: Calibration, computer-assisted surgery, endoscopy, image-guided surgery, minimally invasive surgery, volume rendering.

I. Introduction

RECENT advances in computation technology have greatly expanded the applications of medical imaging, improving diagnosis and facilitating the increased use of computer-assisted surgery (CAS). Improvements in computation power have allowed computer-assisted methods to provide real-time navigation during surgical procedures, including analysis and inspection of three-dimensional (3D) diagnostic images using magnetic resonance (MR) and computed tomography (CT) data. Conventional frame-based stereotactic registration techniques are being replaced by frameless systems that incorporate fiducials (small, removable markers with high contrast in the image data) for registration of the imaging data with the patient.

Endoscopic technology has also undergone rapid development in recent years. During the last two decades, new minimally-invasive surgical approaches have developed at a rapid pace for applications in the spine and the brain, as well as in soft-tissue applications such as abdominal and pelvic surgery. Endoscopes are however still limited in being able to display only visible surfaces, lack adequate orientation indicators, and are unable to provide views of the interior of opaque tissue.

A new technique called *image-enhanced endoscopy* enables real-time 3D surgical navigation using a surgical endoscope by overlaying volumetrically-reconstructed patient images onto the endoscopic view of the surgical field. This technique permits surgeons to use the endoscope to explore inner anatomical structures of the patient, providing "on-the-fly" 3D and two-dimensional (2D) information for planning and navigational purposes. The crucial component in this system is the navigation software, which performs the complex computational, registration, and calibration procedures necessary in order to offer an intuitive interface to the overlaid information.

The author is with the Image Guidance Laboratories at Stanford. Email: mbax@stanford.edu

Since the goal of endoscopy is to reduce the amount of tissue affected by the surgical procedure, the size of the pathology and the overall region of interest are generally small. Consequently the errors introduced into the system at stages such as medical image acquisition, patient-image registration, endoscope calibration, and image overlay have relatively large and cumulative effects on the overall error in registration of the overlaid endoscopic images with the patient's anatomy. Precise calibration of the endoscope and exhaustive accuracy testing of the entire system is therefore necessary to determine how accurately the virtual endoscopic image is registered with the video image from the physical endoscope and to ensure surgical quality.

II. Background

In the past the successful correlation of pre- and intraoperative imaging studies depended on the surgeon's knowledge of human anatomy and visualization skill. Stereotaxis was one of the first methods to effectively correlate preoperative images with a patient's anatomy.

A. Stereotaxis

In frame-based stereotaxis a stereotactic frame is screwed into the patient's skull prior to imaging and is worn throughout the procedure for registering [1] the patient with the preoperative data [2]–[4]. Use of a frame allows for extreme precision, such as for the biopsy of small areas of brain tissue, but often causes patient discomfort and limits surgical flexibility. This technique also does not allow the surgeon to visualize the surgical approach to the lesion.

Techniques have been developed which have largely eliminated the need for the stereotactic frame by instead using alternatives such as a robotic arm for registration [5]–[7], ultrasonic tracking

[8], or an optical tracking system (OTS) which uses infrared light-emitting diodes or reflectors, cameras and triangulation [9], [10] to track moving objects in 3D space [11]–[14]. These systems use fiducials to register imaging data with the patient [15], but have limited positional accuracy and use only 2D images.

Intraoperative 3D image guidance has been implemented using various methods [16]–[18]. Volumetric imaging techniques have greatly enhanced the surgeon's ability to create a plan prior to surgery, to follow it during the operation, and to modify the approach as the surgery progresses. Until recently, however, intraoperative use of 3D imaging was limited by computation power; new advances in graphics hardware have led to feasible intraoperative guidance using 3D images, such as the fusion of preoperative 3D images with video images of the surgical site [19], [20].

B. Endoscopy

New developments in endoscopic hardware has provided small, lightweight endoscopes that are used during surgery to internally inspect body structures. Endoscopy has been used in laparoscopic abdominal surgery and is beginning to be used for smaller, more delicate surgeries such as neurosurgical third ventriculostomy.

New techniques are being developed to allow image-enhanced endoscopy based on fluoroscopic images of the lumbosacral spine that are taken during the surgery itself: the system performs image matching and registration based on two orthogonal fluoroscopic views to allow for minimally-invasive spinal surgery.

A fundamental limitation of the surgical endoscope is its ability to display only visible surfaces. *Virtual* endoscopy, the use of 3D medical imaging data to simulate the field of view of a surgical endoscope [21], can make opaque tissues appear transparent, giving surgeons the capability of looking beyond visible surfaces. By registering the video output of the endoscope with the virtual endoscopy system [22], the pathology and its surrounding tissue is given additional exposure during surgery. Fusion of these displays, by varying the transparency of the virtual endoscopy images and overlaying them on top of the endoscope video, presents surgeons with the most complete understanding of the surgical corridor.

C. The Image-Enhanced Endoscopy System

The system is implemented using the following steps:

- Patient to image registration — matching the patient's physical anatomy to the patient's imaging data.
- Virtual patient reconstruction — generating a 3D model from previously-acquired 3D imaging data of the region of interest in the patient.
- Dynamic data filtering — adjusting the opacity of different tissue types in the virtual patient so that tissue of interest can be revealed in an interactive manner by making the surrounding tissue transparent.
- Conventional navigation — tracking surgical tools within the operating room environment and rendering standard views of the virtual patient.
- Perspective volumetric navigation — tracking surgical tools and rendering the 3D virtual patient as seen from the tool's perspective.
- Image-enhanced endoscopy — synchronizing virtual patient images with intraoperative endoscopic video output.

A description of each technology follows, using a hypothetical scenario to delineate the system features. For clarity the limited scope of surgical guidance in the brain has been used, but it should be emphasized that this system is not limited to neurosurgical procedures alone.

1) Patient to Image Registration: Prior to obtaining preoperative images, five to seven adhesive fiducial markers are placed on the patient's skull. A CT or MR scan is typically performed, using an intravenous contrast agent if appropriate. The image dataset is then transferred to the endoscopy system (and in the case of multiple imaging modalities, fused).

In the operating room, the patient is positioned according to standard practice and registration is accomplished using an optically-tracked surgical probe to locate the physical position of the markers in the 3D image dataset.

2) Virtual Patient Reconstruction: The image dataset is reconstructed using volume rendering, which creates a 3D model of the patient's anatomy and preserves the integrity of the dataset during visualization. This allows surgeons to visualize imaging data from any perspective and to see a "tool's-eye view" of the patient's internal anatomy beyond the incision. This perspective view is used during preoperative planning and intraoperative navigation.

3) Dynamic Data Filtering: Intensity-based data filtering allows surgeons to selectively alter the displayed transparency or opacity of different tissues. Tissue can be emphasized, de-emphasized, or removed so the surgical tool can see beyond the standard rendered view. Using dynamic data filtering the 3D image data is filtered interactively, allowing detailed visualization of the lesion within the appropriate anatomic context (vascular structures, brain, or skull). Segmentation of the lesion (if applicable) will however be only partly automatic because irregular anatomy surrounding lesions is currently too unpredictable for automatic segmentation algorithms.

4) Conventional Navigation: After registration, any tracker-equipped surgical probe can be tracked in physical space, and its virtual counterpart can mirror its orientation and position. A line extending from the virtual probe indicates the trajectory of the planned approach. Reformatting the 3D data allows the visualization of arbitrary cross-sections as cutting planes.

5) Perspective Volumetric Navigation: 3D images of the surgical corridor from the tool's perspective are displayed in real time, and the region of interest is visualized as seen through the surgical tool's bird's-eye-view. When perspective volumetric navigation is coupled with dynamic data filtering, internal organs and their spatial relationship with other sensitive structures beyond the operative site are revealed.

This system not only simplifies preoperative planning for minimally invasive surgery and transforms it into an interactive task, but also provides instantaneous variations of transparency so that deep and superficial structures can be seen in context.

6) Image-Enhanced Endoscopy: Image-enhanced endoscopy synchronizes endoscopic video images with perspective volume-rendered images of the virtual patient, replacing the surgical probe with a tracked rigid endoscope.

There are two approaches to integrating the virtual and video images. They may be displayed side-by-side, or overlaid.

Overlaying the video image on the virtual image is problematic, as it is difficult to determine what parts of the video image to make transparent — this falls into the domain of image recognition, which is a largely unsolved problem.

A more practical approach is to project the video image onto the volume-rendered virtual patient as a texture map [23]–[26]. This has some problems for real-time application, however: there is little point in rendering the virtual image from the same position as the endoscope camera, since the result would look like the existing video image.

Rendering the image from other positions also has pitfalls: much of the video image detail is lost if the viewpoint is not at a similar distance from the region of interest as the endoscope, other structures are likely to obscure the view of the image, and it requires the surgeon to master yet another user interface in parallel with the endoscope.

If one image is to be overlaid on another, the virtual image presents itself as a natural candidate for the overlay: dynamic data filtering may be used to render only those structures of interest, even if they are below optically opaque surfaces from the position of the endoscope.

Due to design constraints, the telescopic lens of an endoscope introduces non-linear and primarily radial distortion into the optical path. (An example of this distortion may be seen below in Fig. 3.) Consequently the video output is substantially different to the undistorted display produced by perspective volume-rendering.

If the virtual display is to be overlaid on the endoscope video, either the distortion must be removed from the video image or the virtual image must be distorted to match the video. Following calibration, this system adjusts the perspective images produced by the virtual trajectory to match the calibrated characteristics of the endoscope's lens, thus allowing a synchronized side-by-side or overlaid display of the preoperative data and live endoscope video.

III. Theory

A. Calibration

Calibration of a tracked surgical instrument yields six extrinsic parameters: three each for rotation and translation relative to the attached tracking device. Since the tracking system supplies the dynamic orientation and position of the tracking device as the instrument moves, the orientation and position of the instrument itself can subsequently be determined by applying the local relative transformation.

Endoscope calibration requires the additional estimation of intrinsic parameters which are used to model the operation of the camera as a pinhole projection camera. In particular, radial lens distortion can be modeled as an even power series of radial position [27]; typically only the first power term is significant:

$$r_u = r_d(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4 + \ldots). \quad (1)$$

Four parameters were calibrated and used here:

f	Focal length in mm.
κ_1	Radial lens distortion coefficient.
(C_x, C_y)	Video image coordinates in pixels of the collocated center of radial distortion and piercing point of camera's z-axis in the focal plane.

A method for determining the above-mentioned parameters has been developed by Tsai [28]. Unlike intensity-based methods, the feature-based algorithm is fast as it requires no rendering, and initialisation is robust. It computes both the extrinsic and the intrinsic parameters of the endoscope; other approaches [29]–[31] handle only endoscope lens-distortion and require an additional step to determine the remaining parameters.

The input parameters to Tsai's algorithm are the physical coordinates (x, y, z) and the corresponding image coordinates (u, v) of a set of points visible through the camera. The best results are obtained when the collection of points are distributed across x, y, and z in physical space and fill as much of the camera's field of view as possible.

Given the camera focal length and inter-pixel distance, there is a direct trigonometrical relationship between the camera field of view (FOV) angle and the extent of the visible field in the camera image, and therefore the FOV angle may be determined directly from the video image after camera calibration.

B. Coordinate Transformation

In order to track the endoscope and determine what is visible in the video output, a transformation from the dynamic orientation and position of its attached tracker to the orientation and position of the camera model is required. This transformation may be calculated as

$$T_{EC} = T_{GC}(T_{UR}T_{GU})^{-1}T_{ER} \quad (2)$$

using the following 4x4 transformation matrices:

T_{ER}	Converts orientation and position in the endoscope tracker's space to orientation and position in system reference space. Supplied by the tracking system.
T_{UR}	Converts from the calibration unit tracker's space to system reference space. Supplied by the tracking system.
T_{GU}	Converts from the calibration target grid's space to the calibration unit tracker's space. Determined by calibration unit geometry.
T_{GC}	Converts from the calibration target grid's space to the endoscope camera model's space. Computed by endoscope calibration.

C. Lens-Distortion Compensation

1) No Compensation: If lens-distortion is not considered, the images are simply scaled to match each other at the image center (where distortion is negligible), and no attempt is made to correct radial distortion:

$$r_u = r_d \quad (3)$$

where r_d is the radius from the image-center of a point within the lens-distorted endoscope video image, and r_u is the radius that point would have if there were no lens distortion.

This will result in substantial differences towards the edge of the image in the presence of high distortion, but it is nevertheless an attractive option since it requires no additional processing time and does not affect the frame rate of the display.

2) Full Compensation: The ideal lens-distortion compensation solution is either to remove the distortion from the endoscope video output or to distort the virtual display using the radial lens-distortion model from calibration [28]:

$$r_u = r_d(1 + \kappa_1 r_d^2). \tag{4}$$

This is not without cost as it is a non-linear computational task which may reduce frame rate if performed in addition to other tasks for each frame. Either approach would result in image-matching free of all modeled lens-distortion error, but in practice it is better to distort the virtual display, as surgeons are accustomed to and comfortable with conventional endoscope lens-distortion.

3) Linear Compensation: If available processor time is limited, a compromise approach here dubbed *"invariant-radius linear compensation"* is to linearly scale the images such that all points around a given radius in each image match without error:

$$r_u = r_d(1 + \kappa_1 r_{d_0}^2) \tag{5}$$

where r_{d_0} is the chosen invariant radius.

A typical choice would be half the distance from the center to the circumference of the image. In this case the image match would be free of lens-distortion error both at the image center and at any point halfway between the image center and its outer circumference.

Fig. 1 illustrates the three approaches.

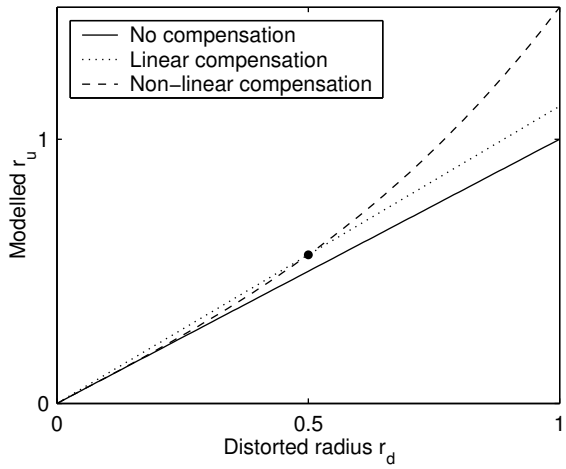


Fig. 1. Models for radial distortion compensation.

IV. Materials

The endoscope equipment comprised an Endovision TRICAM SL video camera system with a Parfocal Zoom lens, a XENON NOVA Light Source; and a 50200A Telescope 0° (KARL STORZ Endoscopy America Inc., Culver City, California).

The tracking system using for testing was a hybrid POLARIS Optical Tracking System (Northern Digital Inc., Waterloo, Ontario, Canada) with a accuracy specification of 0.35 mm RMS.

The HP005-P pointer and the universal trackers were from Traxtal Technologies, Toronto, Ontario, Canada.

The visualization and navigation software was initially designed and prototyped at the Image Guidance Laboratories [22], and developed into an FDA-cleared, commercial package distributed through CBYON, Inc., Palo Alto, California. The software ran on a dual-processor 866 MHz Pentium III workstation.

The endoscope calibration unit's target features are black dots with diameters of 1 mm and 2 mm, in a precision planar grid with a 3 mm inter-dot spacing as shown in Fig. 2. The grid is mounted in the unit along with a rigidly-attached tracker, and the physical coordinates of the dots are known with respect to this tracker. In order to achieve a distribution of the calibration points in the z-direction, the calibration unit is designed to position the axis of the telescope lens at an angle of 30° to the grid plane normal.

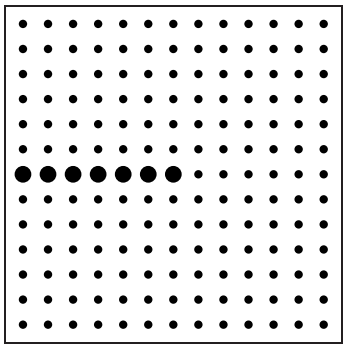


Fig. 2. The calibration target grid.

The physical target used in accuracy testing was a grid similar to that used for calibration, but extending over a region large enough to fill the FOV at all testing distances. The grid was mounted on a moveable plate with a rigidly-attached tracking device fixed on it. There was a registration crosshair at each corner and at the middle of each edge of the grid.

V. Method

A. Endoscope Calibration

An endoscope with its own rigidly-attached tracker is placed in the calibration unit with the tip of its lens approximately 15 mm away from the grid. When the user presses a foot-pedal, a snapshot of the endoscope video output is stored along with the orientation and location of the endoscope and calibration unit tracking devices.

The calibration routine automatically detects and identifies the dots in the endoscope video snapshot, as illustrated in Fig. 3.

1) Feature Detection:

- A 2D Gaussian filter is used to reduce video noise in the captured image, and Canny edge detection is used to find the edges of the dots.
- Closed edge loops with no outliers are isolated and all such loops with perimeters within a fixed range are flagged as dots.
- The center of the dot is computed by finding the mean position of all pixels enclosed by the edge loop. This is robust against edge error and yields sub-pixel accuracy:

$$\mathbf{x}_{\text{dot}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i. \tag{6}$$

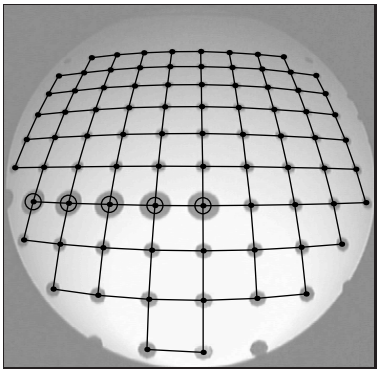


Fig. 3. Detection and identification of the calibration target dots in the endoscope video snapshot.

- The radius of the dot is similarly computed as the radius of an ideal circle with an area equal to the number of enclosed pixels:

$$r_{\text{dot}} = \sqrt{\frac{N}{\pi}}. \tag{7}$$

2) Feature Identification:

- Large dots are separated from small dots based on relative radius. The two large dots furthest from each other in the image are located, and the one that is collinear with one large and one small adjacent dot is determined to be the center dot.
- Since the location of the center dot and the other large dots are now known, the orientation of the dot grid in the image is determined. The radii of the dots also determine the scale of the image.
- Using *a priori* knowledge of the grid layout, the dot identification routine uses the relative locations of identified dots to classify adjacent unknown detected dots. In this way the detected dot locations are mapped to their positions in the grid matrix.
- The set of known physical and detected image coordinates for the identified dots are then passed to the Tsai algorithm to determine the endoscope camera calibration parameters.
- After calibration, the dynamic orientation and position of the modeled endoscope camera relative to the patient is known, and virtual endoscopic images matching the endoscope video images can be rendered using the preoperative volume dataset.

B. Post-Calibration Accuracy Testing

1) Test Configuration: A tracked pointer was used to locate, relative to the attached tracking device, the positions of the eight crosshairs around the grid. The dot positions were computed using the known relationship between the positions of the crosshairs and dots.

The endoscope was calibrated and mounted perpendicularly facing this grid. The grid was sequentially stepped in 1 mm intervals from 5 to 65 mm from the tip of the endoscope's telescope. At each position the physical locations and orientations of the endoscope and of the grid plate tracking device were stored with a snapshot of the endoscope video output.

This process generated a 3D volume of grid points with associated tracking data and endoscope video images.

2) Error Calculation: A point in the video image corresponds to a ray from the center of projection (the camera model's optical origin) passing through that point in the image projection plane (the model's equivalent to the camera CCD). If there is no modeling or tracking error, this ray will also pass through the physical point seen in that image.

If there is residual lens distortion, tracking error or other error unaccounted for in the system model, the ray will miss the corresponding physical point by some distance, denoted e_p. This is defined as the shortest distance between the physical point and the ray, as shown in Fig. 4. Conversely, a ray between the computed center of projection and the physical point will pass through the image projection plane some distance away from where the corresponding point in the video image would be; this error, denoted e_i, is defined as the distance between the intersection of this ray with the image plane and the position of the corresponding video image point in the image projection plane, also shown in Fig. 4.

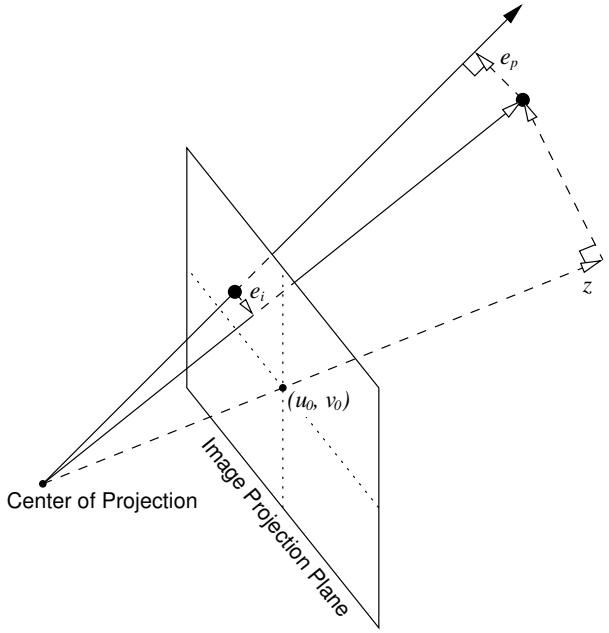


Fig. 4. An image error within the image projection plane and the corresponding projection error in physical space.

The dots in each target position image were located and matched with the corresponding grid dots using the method described for calibration. The position of the physical dots was computed using the tracker-relative co-ordinates of the dots and the stored orientation and position of the target tracker, and the image and projection errors were calculated. The dot radii — the distances of the dots from (C_x, C_y) — were calculated and quantized to 10% intervals for image error charting; the target grid plane z-axis distances were quantized to 5 mm intervals for projection error charting.

3) Endoscope Lens Distortion Compensation: When computing e_i, the position of the ideal projection of the physical dot onto the image plane was computed. This position was then radially distorted to account for the lens distortion in the endoscope video

output image.

Conversely, when determining e_p, the position in the image plane of the dot in the endoscope video image was computed. The radial distortion of this position was calculated and used to position the ray towards the physical point so that its intersection with the image plane fell where it would have been if the video image had been undistorted.

Errors e_i and e_p associated with every detected dot were separately computed using each of the three approaches to lens-distortion compensation.

These errors were also computed using the images of the calibration grid already used for calibration. Re-projecting the calibration points using the calibrated parameters allowed the accuracy of calibration and the fitness of its lens-distortion model to be assessed, since any residual errors were attributable to elements of the system which the calibration model was unable to take into account.

C. Synthetic Calibration Accuracy Testing

In order to compare the effectiveness of 2D planar calibration at providing a representative sample for calibration with that of an optimal 3D volume calibration, accuracy testing was also performed after calibration using the physical and image positions of the points in each of two targets extracted from the 3D grid point volume used for accuracy testing.

1) Synthetic Planar Calibration: Synthetic planar calibration was performed by calibration using the subset of points in the volume that lay on a plane centered at 15 mm from the lens tip and with its normal at 34° to the axis of the endoscope lens.

2) Supercalibration: A reference synthetic "supercalibration" was performed by calibration using the 455 points axially closest (up to 19 mm along the camera z-axis) to the endoscope tip. This represented calibration using the entire near visible point volume, which is theoretically optimal within that region since all available information is used.

VI. Results

A. Residual Calibration Error

The residual mean normalized image error e_i in the calibration image following calibration was 0.0023 (0.23%) of the full diameter of the field of view. With a display 512 pixels in diameter, this equates to an average error of 1.2 pixels.

The residual mean projection error e_p was 0.064 mm.

B. Post-Calibration Accuracy Results

Mean image error, again normalized to the field of view, is shown in Fig. 5.

At points in the image closer to the center than to the edge, there is no significant difference in image error. However as the point radius increases, the uncompensated error grows rapidly. Full radial lens-distortion compensation is shown to be highly effective, as the image error in this case is virtually independent of radius. Linear compensation is about half as effective as full compensation.

At 90% of the maximum radius in a 512-pixel display, this translates to a mean error of 38.1, 8.4 and 25.8 pixels respectively.

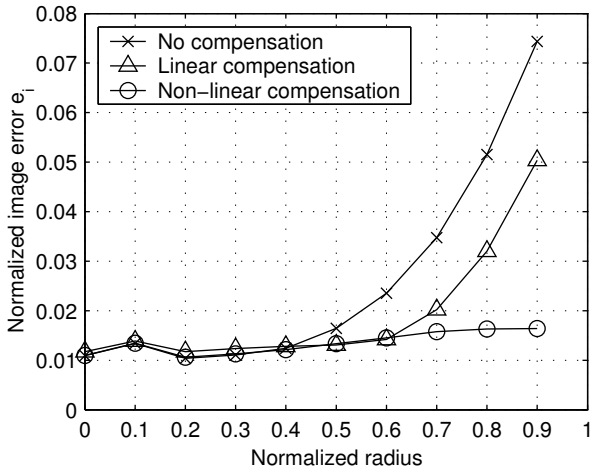


Fig. 5. Mean image error (normalized to the full field of view) versus radial position in the image (normalized to the maximum radius) using each of the three approaches to radial lens-distortion compensation.

Mean projection error as a function of distance along the z-axis away from the endoscope tip (along the axis of the lens telescope) is shown in Fig. 6.

Full radial lens-distortion compensation results in a mean projection error of less than 0.5 mm at distances of up to 25 mm, and halves the uncompensated error at greater distances. At no distance does the mean projection error exceed 1.5 mm.

Linear compensation is again shown to be about half as effective as full compensation.

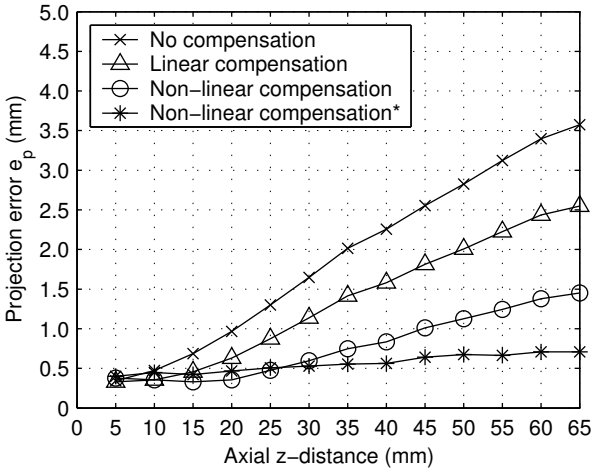


Fig. 6. Mean projection error versus target distance from the lens tip along the camera z-axis (the axis of the endoscope telescope) using each of the three approaches to radial lens-distortion compensation.

*Synthetic supercalibration

C. Synthetic Calibration Accuracy Results

At 0.02, the overall average normalized mean image error for planar calibration was exactly twice the average normalized image error of 0.01 for supercalibration.

Mean projection error as a function of z-axis distance from the endoscope tip using full lens-distortion compensation is shown in

Fig. 7. Compared to planar calibration using the calibration unit, the greater errors for the synthetic version of planar calibration are not unexpected: the synthetic plane was constrained to a steeper angle to the camera z-axis by the 3D grid spacing, and jitter in point position resulted in calibration points not falling precisely on a 2D plane.

It is however apparent that planar calibration is comparable to the optimal supercalibration for z-axis distances of up to 20 mm. At the maximum distance the projection errors grow up to 300% larger.

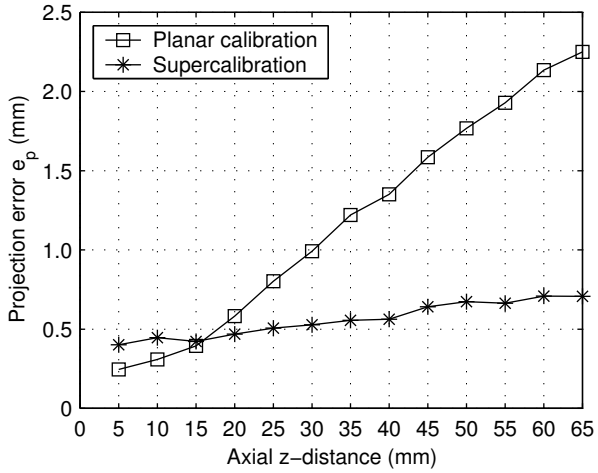


Fig. 7. Mean projection error versus target distance from the lens tip along the camera z-axis (the axis of the endoscope telescope) using synthetic calibration methods.

VII. Discussion

A. Sources of Error

The primary sources of error in the overall system are the precision with which registration is performed, the tracking system errors [32] and, if not fully compensated for, endoscope lens distortion. Measurable positional accuracy of objects in the video image is also limited by the resolution of the image. NTSC video signals generally have a resolution of up to 720×480; in practice this is limited by the lowest resolution element in the signal path from CCD to frame grabber. Since each pixel represents a sample of a frustum of space, this positional uncertainty is also proportional to the object's distance from the endoscope.

The calibration image error was approximately one pixel in size. This indicates that the calibration model is adequate, since this error is of the same order as the spatial quantization error of the image. However, some incremental improvement should be possible by modeling perspective distortion when locating the center of each grid dot.

In most endoscopes it is possible to rotate the optics relative to the imaging sensor. The effect of this rotation was not investigated, but it is expected to be relatively small, since the radial lens distortion model is independent of rotation.

Rotation and translation of the patient is accounted for by using a tracker rigidly attached to the patient, but soft tissue deformation may lead to a situation where the pre-operative data no longer matches the configuration of the patient's anatomy. Since such deformation is outside the domain of the virtual endoscopy system, any error introduced in this way will add to the total system error.

B. Lens-Distortion Compensation

The rendering of a single 512×512 colour frame using the ATI RADEON 8500 64 MB graphics accelerator takes about 60 ms for a typical CT volume ($512 \times 512 \times 128$) and about 40 ms for a typical MRI volume ($256 \times 256 \times 256$).

An efficient implementation of the non-linear lens-distortion compensation algorithm takes approximately 5 ms on a dual 2.2 GHz Xeon workstation, leaving at least 10 ms for other tasks at a video framerate. Thus if the simultaneous computational load is light, full non-linear lens-distortion compensation is quite feasible. Since volume rendering is performed by the graphics accelerator, there is ample time for full lens-distortion compensation if no other tasks are being performed and single-frame latency is acceptable.

As an alternative to full, non-linear distortion compensation where the real-time computational load is high, invariant-radius linear compensation was shown to be an effective method of partially compensating for non-linear lens distortion. It provides 50% of the accuracy improvement yielded by non-linear distortion compensation, without the computational impact and resulting potential performance effects, since additional linear scaling can be performed by the graphics subsystem hardware at no additional cost while displaying the framebuffer.

If non-linear distortion compensation is required when the computational load is high or high-resolution colour video is provided, it is possible to offload distortion compensation to the graphics subsystem hardware at the cost of added complexity [33]–[35].

C. Clinical Applications

Fig. 8 and Fig. 9 illustrate the use of these algorithms in a real clinical setting.

References

[1] P. H. Schonemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, pp. 1–10, 1966.

[2] T. M. Peters and A. Olivier, "CT-aided stereotaxy for depth electrode implantation and biopsy," *Canadian Journal of Neurological Sciences*, vol. 10, pp. 166–169, 1983.

[3] P. L. Gildenberg, H. H. Kaufman, and K. S. Murthy, "Calculation of stereotactic coordinates from the computed tomography scan," *Neurosurgery*, vol. 10, pp. 580–586, 1982.

[4] P. J. Kelly, P. A. Ball, and S. J. Goerss, "Results of computer tomography-based computer-assisted stereotactic resection of metastatic intracranial tumors," *Neurosurgery*, vol. 22, pp. 7–17, 1988.

[5] B. L. Guthrie and J. R. A. Jr., "Frameless stereotaxy: computer interactive neurosurgery," *Perspectives in Neurological Surgery*, vol. 2, pp. 1–22, 1991.

[6] W. B. Legget, M. M. Greenberg, and W. E. J. Gannon, "The viewing wand: a new system for three-dimensional CT correlated intraoperative localization," *Current Surgery*, vol. 48, pp. 674–678, 1991.

[7] P. F. Hemler, T. Koumrian, J. R. A. Jr., and B. Guthrie, "A three-dimensional guidance system for frameless stereotactic neurosurgery," in *Proceedings of the Fifth Annual IEEE Symposium on Computer-Based Medical Systems*, 1992, pp. 309–314.

[8] J. W. Trobaugh, W. D. Richard, K. R. Smith, and R. D. Bucholz, "Frameless stereotactic ultrasonography: method and applications," *Comput Med Imaging Graph*, vol. 18, pp. 235–246, 1994.

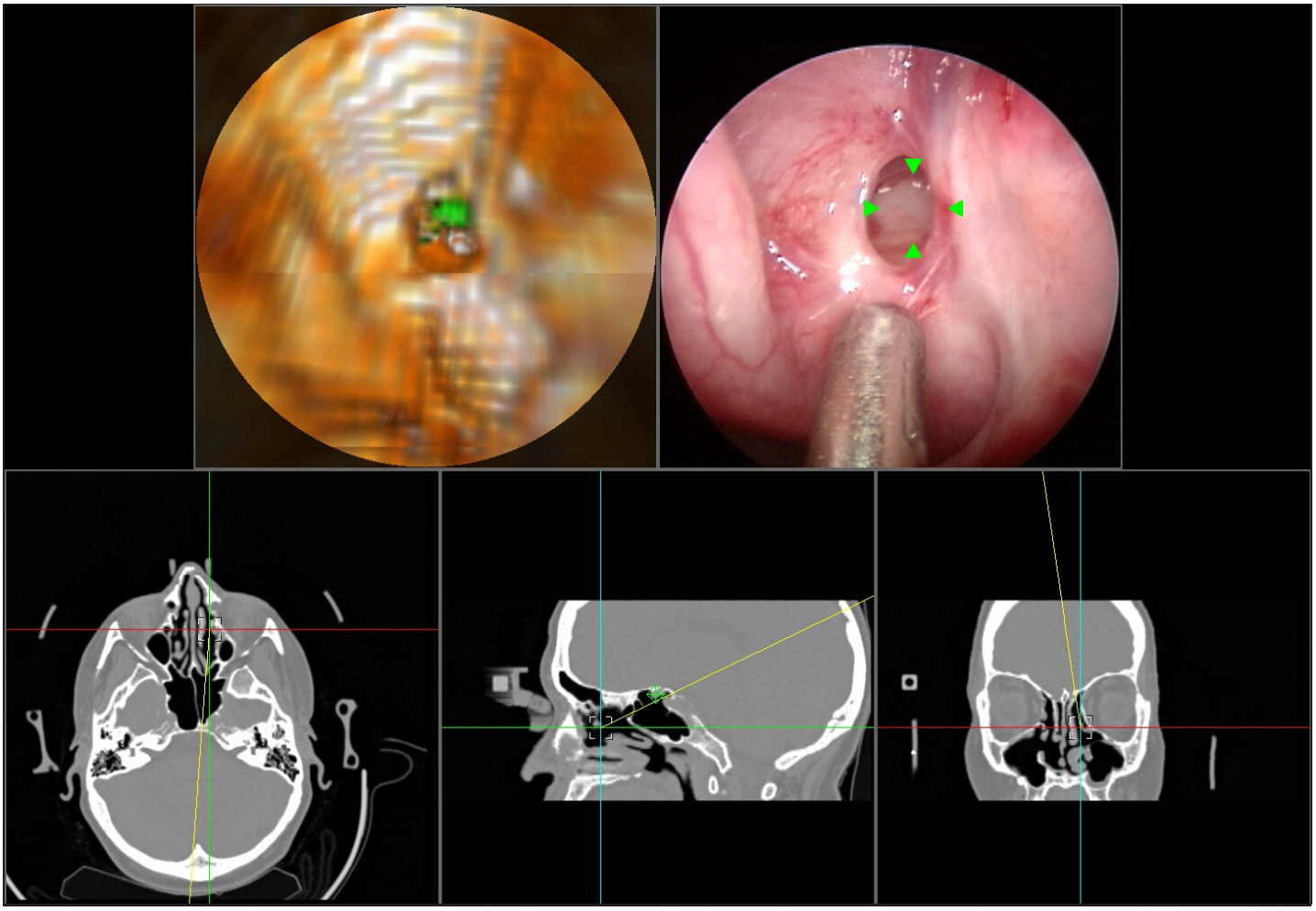


Fig. 8. A pre-operative sinus CT reconstructed into a 3D volume-rendering matching with intraoperative endoscopic images. Rendering revealed sphenoid sinus had pneumatized around left optic nerve (shown in green). Intraoperatively, endoscopic video image and 3D perspective volume-rendered endoscopic image (virtual image) were used to guide surgical dissection without direct entry into sphenoid sinus or hitting the optic nerve (anti-targeting).

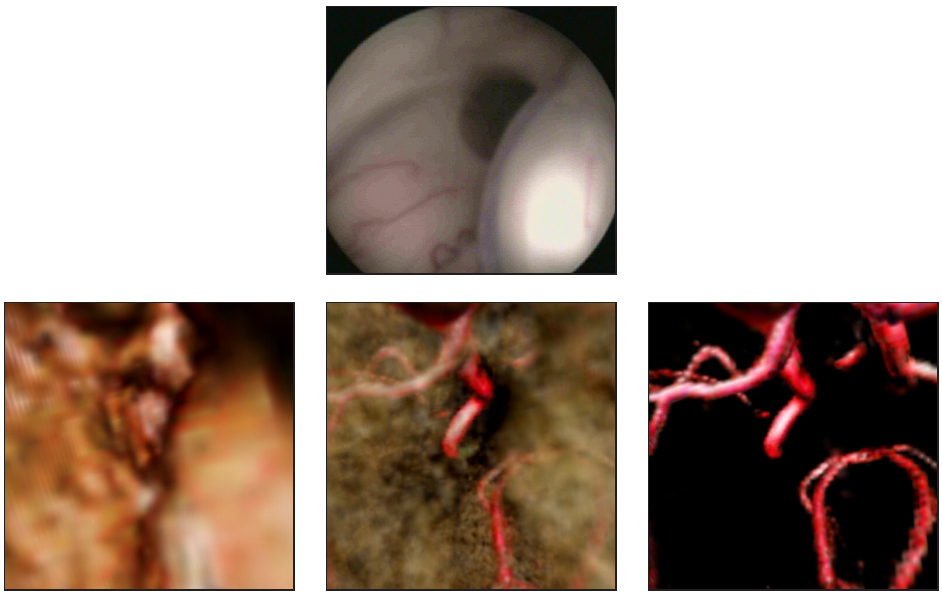


Fig. 9. Image-enhanced endoscopy being used on a patient with myelomeningocele-related hydrocephalus who has had multiple shunt failures. This technique allowed for optimal approach for third ventriculostomy including recognition of an anomalous third ventricle floor anatomy and the relationship to the abnormal position of Circle of Willis vessels. The top image shows the view through the optical endoscope, while the three rendered images below it show tissue being made progressivly more transparent to allow the surgeon to see beyond the surface.

[9] E. Krotkov, K. Henriksen, and R. Kories, "Stereo ranging with verging cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1200–1205, 1990.

[10] N. Alvertos, D. Brzakovic, and R. C. Gonzalez, "Camera geometries for image matching in 3D machine vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 897–915, 1989.

[11] K. R. Smith, K. J. Frank, and R. D. Bucholz, "The Neurostation — a highly accurate, minimally invasive solution to frameless stereotactic neurosurgery," *Comput Med Imaging Graph*, vol. 18, pp. 247–256, 1994.

[12] P. C. et al., "Computer assisted medical interventions: passive and semiactive aids," *IEEE Engineering in Medicine and Biology*, vol. 14, pp. 254–263, 1995.

[13] S. Lavallee, J. Troccaz, P. Sautot, B. Mazier, P. Cinquin, and P. Merloz, "Computer-assisted spinal surgery using anatomy-based registration," in *Computer-Integrated Surgery*, R. H. T. et al, Ed., pp. 425–449. MIT Press, Cambridge, MA, 1995.

[14] L. P. Nolte, H. Visarius, E. Arm, F. Langlotz, O. Schwarzenback, and L. Zamorano, "Computer-aided fixation of spinal implants," *Journal of Image Guided Surgery*, vol. 1, pp. 88–93, 1995.

[15] H. F. Reinhardt, "Neuronavigation: a ten-year review," in *Computer-Integrated Surgery*, R. H. T. et al, Ed., pp. 329–341. MIT Press, Cambridge, MA, 1995.

[16] R. K. et al., "Image-guided techniques for neurosurgery," in *Proceedings of Visualization in Biomedical Computing '94*, 1994, pp. 537–540.

[17] W. E. L. Grimson, G. J. Ettinger, S. J. White, T. Lozano-Perez, W. M. Wells, and R. Kikinis, "An automatic registration method for frameless stereotaxy, image-guided surgery, and enhanced reality visualization," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 129–140, 1996.

[18] A. C. F. C. at al, "Craniotomy simulation and guidance using a stereo video based tracking system (VISLAN)," in *Proceedings of Visualization in Biomedical Computing '94*, 1994, pp. 541–551.

[19] R. Shahidi, R. Mezrich, and D. Silver, "Proposed simulation of volumetric image navigation using a surgical microscope," *Journal of Image Guided Surgery*, vol. 1, pp. 249–265, 1995.

[20] A. C. F. C. et al, "Development and preliminary evaluation of vislan, a surgical planning and guidance system using intra-operative video imaging," *Medical Image Analysis*, vol. 1, pp. 73–90, 1996.

[21] K. Darabi, K. D. Resch, J. Weinert, U. Jendrysiak, and A. Perneczky, "Real and simulated endoscopy of neurological approaches in an anatomical model," in *Proceedings of CVRMED-MRCAS '97*, 1997, pp. 323–326.

[22] R. Shahidi, B. Wang, M. Epitaux, R. Grzeszczuk, and J. Adler, "Volumetric image guidance via a stereotactic endoscope," in *Proceedings of the First Annual Conference on Medical Image Computing and Computer-Assisted Interventions*, W. M. Wells, A. Colchester, and S. Delp, Eds., Cambridge, Massachusetts, Oct. 1998, vol. 1496 of *Lecture Notes in Computer Science*, pp. 241–252, Springer.

[23] P. Jannin, A. Bouliou, E. Journet, and J. Scarabin, "A ray-traced texture mapping for enhanced virtuality in image-guided neurosurgery," in *Proceedings of Medicine Meets Virtual Reality*, S. J. Weghorst, H. B. Sieburg, and K. S. Morgan, Eds., San Diego, California, USA, Jan. 1996, vol. 29 of *Studies in Health Technology and Informatics*, pp. 553–563, IOS Press.

[24] P. Jannin, A. Bouliou, J. Scarabin, C. Barillot, and J. Luber, "Visual matching between real and virtual images in image guided neurosurgery," in *Proceedings of the SPIE Conference on Medical Imaging: Image Display*, Y. Kim, Ed., 1997, vol. 3031, pp. 518–526.

[25] D. Dey, D. G. Gobbi, K. J. Surry, P. J. Slomka, and T. M. Peters, "Mapping of endoscopic images to object surfaces via ray-traced texture mapping for image guidance in neurosurgery," in *Proceedings of the SPIE Conference on Medical Imaging: Image Display and Visualization*, S. K. Mun, Ed., Feb. 2000, vol. 3976, pp. 290–300.

[26] D. Dey, P. J. Slomka, D. G. Gobbi, and T. M. Peters, "Mixed reality merging of endoscopic images and 3-D surfaces," in *MICCAI*, S. L. Delp, A. M. DiGioia, and B. Jaramaz, Eds., Pittsburgh, Pennsylvania, USA, Oct. 2000, vol. 1935 of *Lecture Notes in Computer Science*, pp. 796–803, Springer.

[27] C. McGlone, Ed., *Manual of Photogrammetry*, ASPRS, 5th edition, July 2004.

[28] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1986, pp. 364–374.

[29] W. E. Smith, N. Vakil, and S. A. Maislin, "Correction of distortion in endoscope images," *IEEE Transactions on Medical Imaging*, vol. 11, no. 1, pp. 117–122, Mar. 1992.

[30] H. Haneishi and Y. Miyake, "Distortion correction of electronic endoscope image," in *IEEE Conf. Rec., Med. Imag. Conf.*, San Francisco, California, USA, 1993, vol. 3, pp. 1717–1721.

[31] H. Haneishi, Y. Yagihashi, and Y. Miyake, "A new method for distortion correction of electronic endoscope images," *IEEE Transactions on Medical Imaging*, vol. 14, no. 3, pp. 548–555, Sept. 1995.

[32] R. Khadem, C. Yeh, M. Sadeghi-Tehrani, M. R. Bax, J. A. Johnson, J. N. Welch, E. P. Wilkinson, and R. Shahidi, "Comparative tracking error analysis of five different optical tracking systems," *Computer Aided Surgery*, vol. 5, pp. 98–107, 2000.

[33] M. R. Bax, "Real-time lens distortion correction using texture mapping," in *Proceedings of the SPIE Conference on Medical Imaging: Visualization, Image-Guided Procedures and Display*, R. L. Galloway, Jr., Ed., Feb. 2003, vol. 5029, pp. 249–256.

[34] M. R. Bax, "Real-time lens distortion correction: 3d video graphics cards are good for more than games," *Stanford ECJ*, vol. Spring, pp. 9–13, 2004.

[35] M. R. Bax, C. Maurer, Jr., and R. Shahidi, "Real-time lens distortion correction: Speed, accuracy and efficiency," *SPIE Journal of Biomedical Optics*, 2006, in press.

[36] R. Khadem, M. R. Bax, J. A. Johnson, E. P. Wilkinson, and R. Shahidi, "Endoscope calibration and accuracy testing for 3D/2D image registration," in *Proceedings of the Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2001.

[37] M. R. Bax, R. Khadem, J. A. Johnson, E. P. Wilkinson, and R. Shahidi, "Calibration and accuracy testing for image-enhanced endoscopy," in *Proceedings of Medicine Meets Virtual Reality*, J. D. Westwood, H. M. Hoffman, R. A. Robb, and D. Stredney, Eds., Newport Beach, California, USA, Jan. 2002, vol. 85 of *Studies in Health Technology and Informatics*, pp. 52–56, IOS Press.

[38] M. R. Bax, R. Khadem, J. A. Johnson, E. P. Wilkinson, and R. Shahidi, "Image-enhanced endoscopy calibration and image registration accuracy testing," in *Proceedings of the SPIE Conference on Medical Imaging: Visualization, Image-Guided Procedures, and Display*, 2002.

[39] R. Shahidi, M. R. Bax, C. Maurer, Jr., J. A. Johnson, E. P. Wilkinson, B. Wang, J. B. West, M. J. Citardi, K. H. Manwaring, and R. Khadem, "Implementation, calibration and accuracy testing of an image-enhanced endoscopy system," *IEEE Transactions on Medical Imaging*, vol. 21, no. 12, pp. 1524–1535, Dec. 2002.

Diversity in Rate-distortion Optimized Streaming of Packetized Media

Jacob Chakareski

Abstract. We consider diversity for media streaming in a rate-distortion optimization framework. In a sender-driven transmission, diversity is achieved by using multiple transmission paths over the network. In a receiver-driven transmission, diversity is achieved by requesting media packets from multiple servers. In the first scenario, the proposed framework enables the sender to decide at every instant which packets, if any, to transmit and over which transmission paths in order to meet a rate constraint while minimizing the end-to-end distortion. In the second scenario, the framework enables the receiver to decide at every instant which packets, if any, to request for transmission and from which servers again in order to meet a rate constraint while minimizing the end-to-end distortion. Experimental results demonstrate the benefit of exploiting packet diversity in rate-distortion optimized streaming of packetized media.

Index Terms: rate-distortion optimization, packet path diversity, server diversity, media communication and networking.

I. Introduction

This paper addresses the application of diversity to streaming of packetized media in a rate-distortion optimized way. In a packet switched network, diversity is achieved by using multiple transmission paths over the network, in a sender-driven transmission. Similarly, in a receiver-driven transmission diversity is achieved by simultaneously requesting packets from multiple servers located at different locations in the network. The two prospective diversity scenarios are shown in Figure 1.

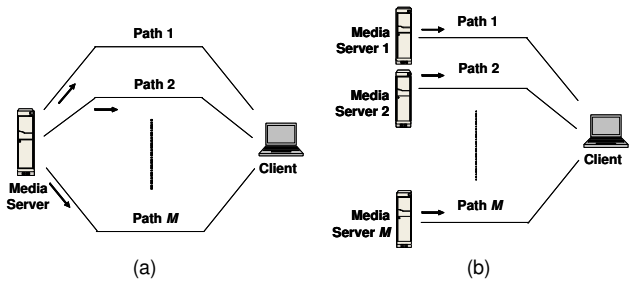


Fig. 1. Diversity in packet networks. (a) Path diversity. (b) Server diversity.

Diversity techniques have been studied for many years in the context of wireless communication. They were introduced in order to exploit the large variability in terms of channel quality when multiple channels are considered for simultaneous transmission. The achieved diversity can be in time, frequency or space, or in any combination of these three [1].

A number of studies have shown that there is an analogous situation in Internet communication: in 30-80% of the cases there is an alternate path that performs significantly better than the default path between two hosts [2]. Performance is measured in terms of round-trip-time, loss rate and bandwidth. These studies have motivated the introduction of packet path diversity for video streaming in [3], where the author proposes to send complementary descriptions of a multiple description (MD) coder through two different Internet paths. The presented experimental results show the potential benefits of the proposed system.

Since then a number of studies have appeared that exploit the concept of packet diversity in media communication. In [4] the authors employ path diversity in the context of video communication using unbalanced MD coding to accommodate the fact that different paths might have different bandwidth constraints. The unbalanced descriptions are created by adjusting the frame rate of a description sent over a particular path. In [5, 6] a framework for real-time voice communication over the Internet is proposed, based on path diversity and MD coding. Significant improvement in performance is reported over existing schemes using a single network path transmission. In [7] the authors study image and video transmission in a multihop mobile radio network. It is shown that combining MD coding and multiple path transport in such a setting provides higher bandwidth and robustness to end-to-end connections. In [8] a receiver-driven framework for distributed streaming of video from multiple senders to a single receiver is proposed. The authors exploit a TCP-friendly protocol and introduce a rate allocation and packet partition algorithms to minimize the packet loss and jitter. It is shown that this results in lower distortion than distributed video streaming in a simplistic, non-adaptive fashion. In a follow up work [9, 10], the authors incorporate Forward Error Correction (FEC) codes into the framework to account for lost packets as retransmissions are not allowed. Receiver-driven streaming of video from multiple servers to a single client is also studied in [11]. The authors propose a network-friendly streaming algorithm based on distributed MD encoding of video. Due to the diversity effect of multiple servers, the algorithm is immune to single points of failure and provides natural load balancing of the servers. In [12] a framework for video transmission over the Internet is presented, based on path diversity and rate-distortion optimized reference picture selection. Here, based on feedback the packet dependency is adapted to channel conditions in order to minimize the distortion at the receiving end, while taking advantage of path diversity. In a related work [13], the authors employ feedback and channel probing within an MD coding framework, to determine which network path should be used for transmission and to adapt the source encoding of the video in order to mitigate error propagation effects. In [14, 15] the performance of path diversity and multiple description coding in Content Delivery Networks (CDN) is studied. 20-40% reduction in distortion is reported over conventional CDNs for the network conditions and topologies under consideration. In [16] the authors examine the performance of MD and of layered coding as two viable strategies for video coding in the context of path diversity in multihop wireless networks. Finally, another related work is [17] where the authors consider rate-distortion optimized streaming over networks with DiffServ support.

In this paper, we present a general framework for rate-distortion optimized streaming of packetized media over multiple network paths. Within the framework, two scenarios are considered: sender-driven and receiver-driven transmission. Media packets are typically characterized by different deadlines, importances and interdependencies, henceforth denoted *source information*. In a sender-driven scenario, using this information and the proposed framework, the sender is able to transmit its media packets over multiple paths based on the feedback it receives in a rate-distortion optimized way, that is, minimizing the expected end-to-end distortion subject to a constraint on the expected overall transmission rate over the paths. Such a rate-distortion optimized transmission

The author is with the Information Systems Laboratory at Stanford. E-mail: cakarz@stanford.edu

algorithm, or transmission policy, results in unequal error protection provided to different portions of the media stream. The core step of the optimization framework involves trading off the expected redundancy (the cost used to communicate a media packet) for the probability that the packet will be communicated in error. Similarly, in a receiver-driven scenario, using the source information and the proposed framework, the receiver is able to request transmission of media packets from multiple servers based on the feedback it receives from the servers in a rate-distortion optimized way. The present work derives from the work in [18], which develops a framework for rate-distortion optimized streaming of packetized media over the Internet. By incorporating models for packet path diversity, error concealment and burst loss channels, we have substantially generalized the framework in [18]. Finally, parts of this work have been presented in [19, 20].

We present the major ideas in our paper as follows. In Section II we define our abstractions of the encoding, packetization, and communication processes. In Section III we show how the entire media presentation can be transmitted in a rate-distortion optimized way, using as a building block an algorithm for rate-distortion optimized transmission of a single media packet. This algorithm is the subject of Section IV. In Section V we report our experimental results. Finally, concluding remarks are provided in Section VI.

II. Source and Channel Characterizations

A. Media Source Model

In a streaming media system, the encoded data are packetized into *data units* and are stored in a file on a media server. All of the data units in the presentation have interdependencies, which can be expressed by a directed acyclic graph. Each node of the graph corresponds to a data unit, where an edge of the graph directed from data unit l' to data unit l implies that data unit l can be decoded only if data unit l' is first decoded.

Associated with each data unit l is a size B_l, a decoding time $t_{DTS,l}$, a set of data units $\mathcal{N}_c^{(l)}$ and an importance $\Delta d_l^{(l_1)}$. Specifically, the size B_l is the size of the data unit in bytes. $t_{DTS,l}$ is the *delivery deadline* by which data unit l must arrive at the client, or be too late to be usefully decoded. Packets containing data units that arrive after the data units' delivery deadlines are discarded. Finally, $\mathcal{N}_c^{(l)} = \{1, \ldots, l\}$ is the set of data units that the receiver considers for error concealment in case data unit l is not decodable by the receiver on time, while $\Delta d_l^{(l_1)}$, for $l_1 \in \mathcal{N}_c^{(l)}$, is the reduction in reconstruction error (distortion) for the media presentation if data unit l is not decodable and is concealed with data unit l_1 that is received and decoded on time.

B. Packet Loss Probabilities

We model a network path between a sender and a receiver, typically a media server and a client, as a burst loss channel using a K-state discrete-time Markov model. The forward and the backward channel make state transitions independently of each other every T seconds, where the transitions are described by probability matrices $\mathcal{P}_{(F)}$ and $\mathcal{P}_{(B)}$, respectively. In each state the forward and the backward channel are characterized as an independent time-invariant packet erasure channel with random delay. Hence, they are completely specified with the probability of packet loss $\epsilon_{F/B}^k$ and the probability density of the transmission delay $p_{F/B}^k$, for $k = 1, \ldots, K$. This means that if the media server sends a packet on the forward channel at time t, given that the forward channel is in state k at t, then the packet is lost with probability ϵ_F^k. However, if the packet is not lost, then it arrives at the client at time t', where the forward trip time $FTT^k = t' - t$ is randomly drawn according to the probability density p_F^k. Therefore, we let $P\{FTT^k > \tau\} = \epsilon_F^k + (1 - \epsilon_F^k) \int_\tau^\infty p_F^k(t)dt$ denote the probability that a packet transmitted by the server at time t, given that the forward channel is in state k at t, does not arrive at the client application by time $t + \tau$, whether it is lost in the network or simply delayed by more than τ. Then similarly, $P\{BTT^k > \tau\} = \epsilon_B^k + (1 - \epsilon_B^k) \int_\tau^\infty p_B^k(t)dt$ denotes the probability that a packet transmitted by the client at time t, given that the backward channel is in state k at t, does not arrive at the server by time $t + \tau$, whether it is lost in the network or simply delayed by more than τ. Finally, we are interested in $P\{RTT^{kj} > \tau\}$, which is the probability that the server does not receive an acknowledgement by time $t + \tau$ for a packet transmitted at time t, given that the forward and the backward channel are respectively in states k and j, at t.

To derive $P\{RTT^{kj} > \tau\}$ assume first that the transmission on the forward channel occurred immediately after the channel made a state transition. If $FTT^k \leq T$, the packet is received by the client before the backward channel makes the next state transition. Then $P\{RTT^{kj} > \tau | FTT^k \leq T\} = P\{FTT^k + BTT^j > \tau | FTT^k \leq T\}$ as the client sends an acknowledgement while the backward channel is still in the current state j. The probability of this event is $P\{FTT^k \leq T\}$. However, if $lT < FTT^k \leq (l+1)T$, for $l \geq 1$, then the state of the backward channel makes l transitions before the packet actually arrives at the client. The probability of this event is $P\{lT < FTT^k \leq (l+1)T\}$. Here the state on the backward channel when the acknowledgement is sent can be any of the K possible values. Hence we compute the desired quantity as the expected value over all of them, i.e., $P\{RTT^{kj} > \tau | lT < FTT^k \leq (l+1)T\} = \sum_{p=1}^K \mathcal{P}_{jp(B)}^{(l)} P\{FTT^k + BTT^p > \tau | lT < FTT^k \leq (l+1)T\}$. Note that $\mathcal{P}_{jp(B)}^{(l)}$ is the probability of making a transition from state j to state p in l transition intervals. These probabilities are obtained using matrix power, i.e., $\mathcal{P}_{(B)}^{(l)} = \mathcal{P}_{(B)}^{l}$. Finally, by averaging over all possible outcomes for FTT^k we write

$$
\begin{aligned}
P\{RTT^{kj} > \tau\} = & \sum_{l=0}^{\infty} \sum_{p=1}^{K} \mathcal{P}_{jp(B)}^{(l)} P\{lT < FTT^k \leq (l+1)T, FTT^k + BTT^p > \tau\} \\
& \sum_{l=0}^{M-1} \sum_{p=1}^{K} \mathcal{P}_{jp(B)}^{(l)} P\{lT < FTT^k \leq (l+1)T, FTT^k + BTT^p > \tau\} \\
& + \sum_{p=1}^{K} \mathcal{P}_{jp(B)}^{(M)} P\{MT < FTT^k \leq \tau, FTT^k + BTT^p > \tau\} \\
& + P\{FTT^k > \tau\} \qquad (1)
\end{aligned}
$$

where the first equality follows from Bayes' rule, while the second one holds since $P\{lT < FTT^k \leq (l+1)T, FTT^k + BTT^p > \tau\} = P\{lT < FTT^k \leq (l+1)T\}$, for $lT \geq \tau$. Finally, $M = \lfloor \tau/T \rfloor$ and $\mathcal{P}_{(B)}^{(0)} = \mathcal{I}$, the identity matrix.

In a receiver-driven transmission the client sends request packets to the server, requesting transmission of particular media packets. The server transmits a media packet only when it receives a request for it. Therefore, for this scenario we define $P\{RTT^{kj} > \tau\}$ as the probability of the event the client sends a request on the backward channel at time t, but the requested media packet does not arrive on the forward channel by time $t + \tau$, given that the forward and the backward channel are respectively in states j and k, at t. Building on (1), it is easy to see that we can write

$$
\begin{aligned}
P\{RTT^{kj} > \tau\} = & \sum_{l=0}^{M-1} \sum_{p=1}^{K} \mathcal{P}_{jp(F)}^{(l)} P\{lT < BTT^k \leq (l+1)T, FTT^p + BTT^k > \tau\} \\
& + \sum_{p=1}^{K} \mathcal{P}_{jp(F)}^{(M)} P\{MT < BTT^k \leq \tau, FTT^p + BTT^k > \tau\} \\
& + P\{BTT^k > \tau\} \qquad (2)
\end{aligned}
$$

III. Rate-distortion optimized policy selection

Suppose there are L data units in the media presentation. Let $\pi_l \in \Pi$ be the transmission policy for data unit $l \in \{1, \ldots, L\}$ and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ be the vector of transmission policies for all L data units. Π is a family of policies defined precisely in the next subsection.

Any given policy vector $\boldsymbol{\pi}$ induces an expected distortion $D(\boldsymbol{\pi})$ and an expected transmission rate $R(\boldsymbol{\pi})$ for the media presentation. We seek the policy vector $\boldsymbol{\pi}$ that minimizes $D(\boldsymbol{\pi})$ subject to a constraint on $R(\boldsymbol{\pi})$. This can be achieved by minimizing the Lagrangian $D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$ for some Lagrange multiplier $\lambda > 0$, thus achieving a point on the lower convex hull of the set of all achievable distortion-rate pairs.

We now compute expressions for $R(\boldsymbol{\pi})$ and $D(\boldsymbol{\pi})$. The expected transmission rate $R(\boldsymbol{\pi})$ is the sum of the expected number of bytes transmitted for each data unit $l \in \{1, \ldots, L\}$, $R(\boldsymbol{\pi}) = \sum_l B_l \rho(\pi_l)$, where B_l is the number of bytes in data unit l and $\rho(\pi_l)$ is the expected number of transmitted bytes per source byte (under policy π_l), called the *expected cost*. The expected distortion $D(\boldsymbol{\pi})$ can be expressed in terms of the probability $\epsilon(\pi_l)$ that data unit l does not arrive at the receiver on time (under policy π_l), called the *expected error*. Let data unit l be not decodable on time at the receiver and let the decoder employ data unit $l_1 \in N_C^{(l)}$ to conceal the loss of l. Note that the decoder always prefers the most recent decodable unit from the concealment set, i.e., for $l_1, l_2 \in N_c^{(l)} : l_1 > l_2$ and both l_1, l_2 decodable on time the decoder always chooses l_1 to conceal the missing data unit l.

The probability of the event data unit l_1 is decodable on time is $\prod_{j \in A(l_1)}(1-\epsilon(\pi_j))$, where $A(l_1)$ is the set of ancestors of l_1, including l_1. Now given that l_1 is decodable, the probability of the event none of the data units $j \in N_C^{(l)} : j > l_1$ are decodable on time is

$$\prod_{l_2 \in C(l,l_1)} \left(1 - \prod_{l_3 \in A(l_2) \setminus A(l_1)} (1 - \epsilon(\pi_{l_3})) \right)$$

where $C(l, l_1)$ is the set of data units $j \in N_c^{(l)} : j > l_1$ that are not mutual descendants, i.e., for $j, k \in C(l, l_1) : j \notin D(k), k \notin D(j)$, where $D(j)$ is the set of descendants of data unit j. The product term within the brackets is the probability that all of the ancestors of $l_2 \in C(l, l_1)$, excluding those common with l_1, are received on time, where "\" denotes the operator "set difference". Note that we only need to account for the event that data units from $C(l, l_1)$ are not decodable, as all the other data units from the concealment set are descendants of one or more data units from $C(l, l_1)$ and hence will definitely not be decodable on time. Finally, note that the case when data unit l is decodable is included using this notation as l is the last data unit in $N_c^{(l)}$.

We use these results first to take an expectation over all possible cases of concealment for data unit l and then to sum over all data units in order to obtain

$$D(\boldsymbol{\pi}) = D_0 - \sum_l \sum_{l_1 \in \mathcal{N}_c^{(l)}} \Delta d_l^{(l_1)} \prod_{j \in \mathcal{A}(l_1)} (1 - \epsilon(\pi_j)) \times \prod_{l_2 \in \mathcal{C}(l,l_1)} \left(1 - \prod_{l_3 \in \mathcal{A}(l_2) \setminus \mathcal{A}(l_1)} (1 - \epsilon(\pi_{l_3})) \right) \quad (3)$$

where D_0 is the expected reconstruction error for the presentation if no data units are received. In deriving (3), we assume statistical independence of the losses affecting separate data units for tractability. However, we still account for the dependencies between different packets associated with same data unit, as shown in Section IV. For a further discussion, see [19].

Finding a policy vector $\boldsymbol{\pi}$ that minimizes the expected Lagrangian $J(\boldsymbol{\pi}) = D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$, for $\lambda > 0$, is difficult since the terms involving the individual policies π_l in $J(\boldsymbol{\pi})$ are not independent. Therefore, we employ an iterative descent algorithm, called Iterative Sensitivity Adjustment (ISA), in which we minimize the objective function $J(\pi_1, \ldots, \pi_L)$ one variable at a time while keeping the other variables constant, until convergence [18]. It can be shown that the optimal individual policies at iteration n, for $n = 1, 2, \ldots$, are given by

$$\pi_l^{(n)} = \arg\min_{\pi_l} S_l^{(n)} \epsilon(\pi_l) + \lambda B_l \rho(\pi_l), \quad (4)$$

where

$$S_l^{(n)} = \sum_{l_1 \,:\, l \in N_c^{(l_1)}} S_{l,l_1}^{+(n)} - S_{l,l_1}^{-(n)} = S_l^{+(n)} - S_l^{-(n)} \quad (5)$$

can be regarded as the *sensitivity* to losing data unit l, i.e., the amount by which the expected distortion will increase if data unit l cannot be recovered at the client, given the current transmission policies for the other data units. Note that differently from [18], the sensitivity here consists of two nonnegative terms $S_l^{+(n)}$ and $S_l^{-(n)}$. The first term increases the sensitivity associated with data unit l in case l is in the ancestor set of data unit l_2 used for concealment of a data unit l_1. On the other hand, the second term reduces the sensitivity associated with l in case l is not in the ancestor set of l_2. This result is intuitive and allows us to better model the situations where data unit l is irrelevant for concealment of another data unit. Expressions for $S_{l,l_1}^{+(n)}$ and $S_{l,l_1}^{-(n)}$ are easily obtained from (3) by grouping terms.

The minimization (4) is now simple, since each data unit l can be considered in isolation. Indeed the optimal transmission policy $\pi_l \in \Pi$ for data unit l minimizes the "per data unit" Lagrangian $\epsilon(\pi_l) + \lambda' \rho(\pi_l)$, where $\lambda' = \lambda B_l / S_l^{(n)}$. Thus to minimize (4) for any l and λ', it suffices to know the lower convex hull $\epsilon(\rho) = \min_{\pi \in \Pi}\{\epsilon(\pi) : \rho(\pi) \leq \rho\}$ of the function, which we call the expected *error-cost* function. In the next section we show how to compute the expected error-cost function for the family of transmission policies corresponding to diversity, first in sender-driven transmission and then in receiver-driven transmission.

IV. Computing the expected error-cost function

A. Sender-driven transmission

Assume that there are M network paths over which the server can simultaneously send a data unit to the client. Furthermore assume that there are N discrete transmission opportunities $t_0, t_1, \ldots, t_{N-1}$ prior to the data unit's delivery deadline t_{DTS} at which the server is allowed to transmit a packet for the data unit on the forward channel of any $m \leq M$ paths. The server need not transmit a packet at every transmission opportunity. The server does not transmit any further packets after an ACK is received on the backward channel of any of the paths.

At each transmission opportunity t_i, $i = 0, 1, \ldots, N-1$, the server takes an action $a_i = [a_{i1}, \ldots, a_{iM}]$, where $a_{im} = 1$ means that a packet is sent on the forward channel of path m and $a_{im} = 0$ means that no packet is sent on the forward channel of path m. Then, at the next transmission opportunity t_{i+1}, the server makes an observation o_i, where o_i is the set of acknowledgements received by the server in the interval $(t_i, t_{i+1}]$. For example, $o_i = \{ACK_{j1}^{m_1}, ACK_{j2}^{m_2}\}$ means that during the interval $(t_i, t_{i+1}]$, ACKs arrived on the backward channels for the packets sent at time t_{j_1} and t_{j_2} on the forward channels of paths m_1 and m_2, respectively. The history, or the sequence of action-observation pairs $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_i, o_i)$ leading up to time t_{i+1}, determines the state q_{i+1} at time t_{i+1}, as illustrated in Figure 2. If the final observation o_i includes an ACK, then q_{i+1} is a final state. In addition, any state at time $t_N = t_{DTS}$ is a final state. Final states in Figure 2 are indicated by double circles.

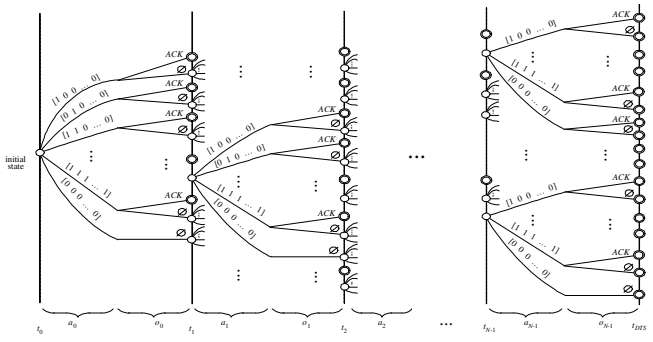


Fig. 2. Markov decision tree for a data unit with packet path diversity.

The action a_i taken at a non-final state q_i determines the transition probabilities $P(q_{i+1}|q_i, a_i)$ to the next state q_{i+1}. Formally, a policy π is a mapping $q \mapsto a$ from non-final states to actions. Thus any policy π induces a Markov chain with transition probabilities $P_\pi(q_{i+1}|q_i) \equiv P(q_{i+1}|q_i, \pi(q_i))$, and consequently also induces a probability distribution on final states. Let q_F be a final state with history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{F-1}, o_{F-1})$, and let $q_{i+1} = q_i \circ (a_i, o_i)$, $i = 1, \ldots, F-1$, be the sequence of states leading up to q_F. Then q_F has probability $P_\pi(q_F) = \prod_{i=0}^{F-1} P_\pi(q_{i+1}|q_i)$, transmission cost $\rho_\pi(q_F) = \sum_{i=0}^{F-1} \sum_{m=1}^{M} a_{im}$, and error $\epsilon_\pi(q_F) = 0$ if o_{F-1} contains an ACK and otherwise $\epsilon_\pi(q_F)$ is equal to the probability that none of the transmitted packets arrives at the client on time, given q_F. Hence, we can express the expected cost and error for the Markov chain induced by policy π: $\rho(\pi) = E_\pi \rho_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\rho_\pi(q_F)$, $\epsilon(\pi) = E_\pi \epsilon_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\epsilon_\pi(q_F)$.

We wish to find the policy π^* that minimizes $\epsilon(\pi) + \lambda' \rho(\pi)$, as discussed in the previous section. We do that by enumerating all possible policies π, plotting the error-cost performances $\{(\rho(\pi), \epsilon(\pi))\}$ in the error-cost plane, and producing an operational error-cost function for our scenario. At every transmission opportunity t_i we find π^*, where $\{(\rho(\pi), \epsilon(\pi)) : \pi \in \Pi\}$ is calculated conditioned on q_i and all the policies under consideration are consistent with the history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{i-1}, o_{i-1})$ leading up to state q_i at time t_i. Then, a_i is set to the first action $\pi^*(q_i)$ of π^*, and the procedure is repeated at each successive transmission opportunity until a final state is reached. Note that it would be sufficient to determine π^* only once at time t_0, except for the fact that λ' may be adjusted by the algorithm from Section III at each transmission opportunity t_i in order to take into account feedback from previous transmissions of other data units.

In the following we explain how $\epsilon(\pi)$ and $\rho(\pi)$ are computed. Let t_i be the current transmission opportunity and let $C^F_{jm}, C^B_{jm} \in \{1, \ldots, K\}$ be respectively the states on the forward and the backward channel of path $m = 1, \ldots, M$ at transmission opportunity $t_j : j \leq i$. We assume that the sender has this information available. It is a reasonable assumption, as any congestion control mechanism employed by a streaming media system, such as those in [21–28], will include some kind of channel estimation. This is because in the absence of explicit feedback from the network, the sender and/or the receiver must infer the state of the network by observing data units as they enter and leave the network.

As explained earlier, the expected error for a policy π is simply the probability that all the transmitted packets from π as well as those from the transmission history do not arrive at the client on time. Hence we write

$$\epsilon(\pi) = \prod_{\substack{j<i,\ m \\ :\ a_{jm}=1}} P\{FTT^{C^F_{jm}} > t_{DTS} - t_j | RTT^{C^F_{jm} C^B_{jm}} > t_i - t_j\} \times \prod_{\substack{j \geq i,\ m \\ :\ a_{jm}=1}} \sum_{k=1}^{K} \mathcal{P}^{(j-i)}_{C^F_{im} k(F)} P\{FTT^k > t_{DTS} - t_j\}. \quad (6)$$

Furthermore, upon receipt of an acknowledgement packet, the server truncates its transmission pattern and does not consider sending any packets afterwards. Therefore, the cost for each transmission $a_{jp} = 1 : j \in \{i, \ldots, N-1\}, p = 1, \ldots, M$ is equal to the probability that none of the previous transmissions results in an acknowledgement packet received by the server by t_j. Hence, the expected cost is simply the sum of the individual costs over all transmission opportunities and paths, i.e.,

$$\rho(\pi) = \sum_{\substack{j \geq i,\ p \\ :\ a_{jp}=1}} \prod_{\substack{l<i,\ m \\ :\ a_{lm}=1}} P\{RTT^{C^F_{lm} C^B_{lm}} > t_j - t_l | RTT^{C^F_{lm} C^B_{lm}} > t_i - t_l\} \times \prod_{\substack{i \leq l < j,\ m \\ :\ a_{lm}=1}} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \mathcal{P}^{(l-i)}_{C^F_{im} k_1(F)} \mathcal{P}^{(l-i)}_{C^B_{im} k_2(B)} P\{RTT^{k_1 k_2} > t_j - t_l\} \quad (7)$$

where the first product term in both $\epsilon(\pi)$ and $\rho(\pi)$ accounts for previous transmissions (if any). Finally, note that the contribution to the error-cost of prospective transmissions in π at opportunities $t_j : j > i$ can be accounted for only as an expected value over all possible channel states at t_j.

B. Receiver-driven transmission

Assume that there are M media servers at different locations in the network with which a client communicates over M independent network paths. Furthermore, assume that there are N discrete transmission opportunities $t_0, t_1, \ldots, t_{N-1}$ prior to a data unit's delivery deadline t_{DTS} at which the client is allowed to transmit a request packet for the data unit on the backward channel of any $m \leq M$ paths. The client need not transmit a request at every transmission opportunity. The client does not transmit any further requests after a packet with the data unit arrives on the forward channel of any of the paths.

At each transmission opportunity t_i, $i = 0, 1, \ldots, N-1$, the client takes an action $a_i = [a_{i1}, \ldots, a_{iM}]$, where $a_{im} = 1$ means that a request is sent on the backward channel of path m and $a_{im} = 0$ means that no request is sent on the backward channel of path m. Then, at the next transmission opportunity t_{i+1}, the client makes an observation o_i, where o_i is the set of packets received by the client in the interval $(t_i, t_{i+1}]$. For example, $o_i = \{DAT^{m_1}_{j_1}, DAT^{m_2}_{j_2}\}$ means that during the interval $(t_i, t_{i+1}]$, packets with the data unit arrived on the forward channels as a response to the requests sent at time t_{j_1} and t_{j_2} on the backward channels of paths m_1 and m_2, respectively. The history, or the sequence of action-observation pairs $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_i, o_i)$ leading up to time t_{i+1}, determines the state q_{i+1} at time t_{i+1}. If the final observation o_i includes a DAT, then q_{i+1} is a final state. In addition, any state at time $t_N = t_{DTS}$ is a final state.

As noted the operation of the client in this scenario is analogous to the one of the server from Section IV-A, with ACKs being replaced with DATs. Therefore, there is no need for describing it again here and we end this section with the corresponding error-cost expressions. We use the same notation and an assumption as in Section IV-A. $\epsilon(\pi)$ is easily obtained from (6) by replacing the forward trip time (FTT) with the round trip time (RTT). Similarly, $\rho(\pi)$ is obtained from (7) by noting the following. A request packet incurs a cost of one, which is the cost of transmission of a single media packet, only if the request arrives eventually at the server. Therefore, the cost of sending a request packet is less than one and equals the probability that the request is not lost on the backward channel of the path. Hence we write

$$\epsilon(\pi) = \prod_{\substack{j<i,\ m \\ :\ a_{jm}=1}} P\{RTT^{C^B_{jm} C^F_{jm}} > t_{DTS} - t_j | RTT^{C^B_{jm} C^F_{jm}} > t_i - t_j\} \times \prod_{\substack{j \geq i,\ m \\ :\ a_{jm}=1}} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \mathcal{P}^{(j-i)}_{C^B_{im} k_1(B)} \mathcal{P}^{(j-i)}_{C^F_{im} k_2(F)} P\{RTT^{k_1 k_2} > t_i - t_j\}$$

$$\rho(\pi) = \sum_{\substack{j \geq i,\ p \\ :\ a_{jp}=1}} \prod_{\substack{l<i,\ m \\ :\ a_{lm}=1}} P\{RTT^{C^B_{lm} C^F_{lm}} > t_j - t_l | RTT^{C^B_{lm} C^F_{lm}} > t_i - t_l\} \times \prod_{\substack{i \leq l < j,\ m \\ :\ a_{lm}=1}} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \mathcal{P}^{(l-i)}_{C^B_{im} k_1(B)} \mathcal{P}^{(l-i)}_{C^F_{im} k_2(F)} P\{RTT^{k_1 k_2} > t_j - t_l\} \times \sum_{k=1}^{K} \mathcal{P}^{(j-i)}_{C^B_{ip} k(B)} P\{BTT^k < \infty\} \quad (8)$$

where the last sum in $\rho(\pi)$ equals the probability that the request packet $a_{jp} = 1$ is not lost on the backward channel of path p.

V. Experimental results

Here we investigate the end-to-end distortion-rate performance for streaming packetized video contents using our framework. The videos are two layer SNR scalable representations of the sequences *Foreman* and *Mother and daughter*, henceforth denoted *MaD*. Using an H.263+ [29] codec the first 130 frames of QCIF *Foreman* have been encoded into a base and enhancement layer with corresponding rates of 32 and 64 Kbps. Similarly, the first 321 frames of QCIF *MaD* have been encoded into two layers with rates 32 and 69 Kbps, respectively. For both videos the frame rate is 10 fps and the size of the Group of Pictures (GOP) is

10 frames, consisting of an I frame followed by 9 consecutive P frames. Performance is measured in terms of the luminance peak signal-to-noise ratio (Y-PSNR) in dB of the end-to-end perceptual distortion, averaged over the duration of the video clip, as a function of the available bit rate on the forward channel(s) of the network path(s).

In the experiments we use $T = 100$ ms as the time interval between transmission opportunities and 600 ms for the playback delay. Furthermore, we employ a $K = 2$ state Markov model for each path. The model parameters are kept same over all paths and are specified in Table 1. In particular, in Table 1a we specify the delay and loss

	Loss	Delay		
	ϵ (%)	κ (ms)	μ (ms)	σ (ms)
State 1	3	25	75	50
State 2	15	25	275	250

(a) Loss and delay parameters.

	π_2	τ_2 (ms)
Model 0	0	0
Model 1	0.2	200
Model 2	0.5	1000
Model 3	0.8	2000

(b) State transitions.

TABLE 1. NETWORK PATH CHARACTERIZATION.

characteristics for a channel state. We keep the same characteristics for the forward and the backward channel. The delay density is modelled using a shifted Gamma distribution specified with three parameters: shift κ, mean μ and standard deviation σ. Finally, the state transitions are modelled using two parameters: the stationary probability of being in State 2, π_2, and the expected duration of stay in State 2, τ_2, once a transition is made to this state. We employ four sets of values for these parameters denoted Model 0 - 3 in Table 1b. Due to the selected values Models 0 - 3 cover a range of possibilities in terms of the loss and delay characteristics exhibited on a network path.

A. Sender-driven transmission

We first study the performance of the proposed framework as a function of the number of paths available. The state transitions are generated using Model 2 in these experiments. It can be seen from

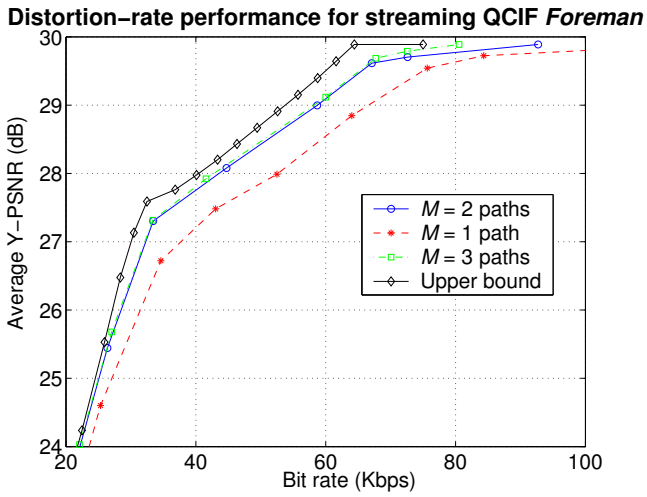


Fig. 3. Sender-driven R-D performance for streaming *Foreman* over $M = 1, 2$ and 3 paths.

Figure 3 that streaming *Foreman* over 2 network paths can improve performance compared to the case of streaming over a single network path. An improvement is observed over the whole range of available rates. The gains in performance are most significant for the range of rates 30 - 70 Kbps and reach up to 0.65 dB. The difference in performance decreases as we move towards very low or very high transmission rates. The improved performance is due to the fact that having an alternative path for transmission reduces dramatically the probability of having to transmit on a forward channel that features degraded quality (State 2) at transmission. This ultimately contributes to a higher likelihood of delivering the media packets on time. Furthermore, it can be seen from Figure 3 that using further paths for streaming does not provide additional gains in performance, since the performances of $M = 3$ and of $M = 2$ are almost identical. As explained above having 2 network paths reduces substantially the likelihood of facing a degraded channel at transmission on every path. Therefore, adding one more path as an alternative does not provide any further benefits, for the selected path model.

Finally, exploiting path diversity in a distortion-rate optimization framework brings the performance of a streaming system closer to that of an "ideal" distortion-rate optimal system working at channel capacity. The performance of such system, shown as "Upper bound" in Figure 3, is computed using the distortion-rate characteristics of the *Foreman* video and the characteristics of the channel. If the sender transmits each data unit an infinite time before its deadline, the channel acts as a binary erasure channel at each state, with a drop probability of ϵ_F^1 and ϵ_F^2, respectively. As the sender has an infinite amount of time available to transmit its data units, it can send all of them during a "good" state (State 1) on the forward channel of a path by simply waiting long enough for a transition to State 1 to occur on the forward channel. As is well known [30], the capacity of an erasure channel with erasure probability ϵ_F^1 is $C = 1 - \epsilon_F^1$. Thus, to successfully transmit each source byte, the sender has to transmit at least $1/C$ bytes over the channel. Therefore, transmission of a data unit with a size B_l requires B_l/C bytes. Knowing ϵ_F^1 and the *source information*, the performance of this system can be computed using an optimal pruning algorithm [31, 32]. Note that for all data units that the system does not transmit, error-concealment is performed at the receiver, as explained in Section III.

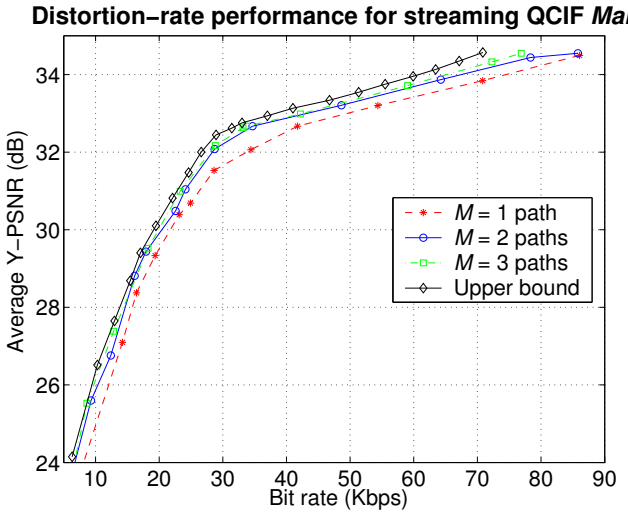


Fig. 4. Sender-driven R-D performance for streaming *MaD* over $M = 1, 2$ and 3 paths.

We observe similar results for streaming *MaD*, shown in Figure 4. As in the case of *Foreman*, having an alternative path for transmission can improve performance over the whole range of available rates. The gains in performance are most significant for the range of rates 25 - 40 Kbps and reach up to 0.55 dB. However, note that the gains here are not as

pronounced as those for the case of *Foreman*. This is due to the nature of the *MaD* video, which exhibits less motion and therefore makes error-concealment more successful in case of packet loss. Furthermore, due to the same reasons explained earlier having more than one alternative path for routing does not provide a significant additional gain in performance, as seen from the case of $M = 3$. Finally, using an alternative path for transmission again narrows the gap in performance between an ideal system and a system that employs only a single path.

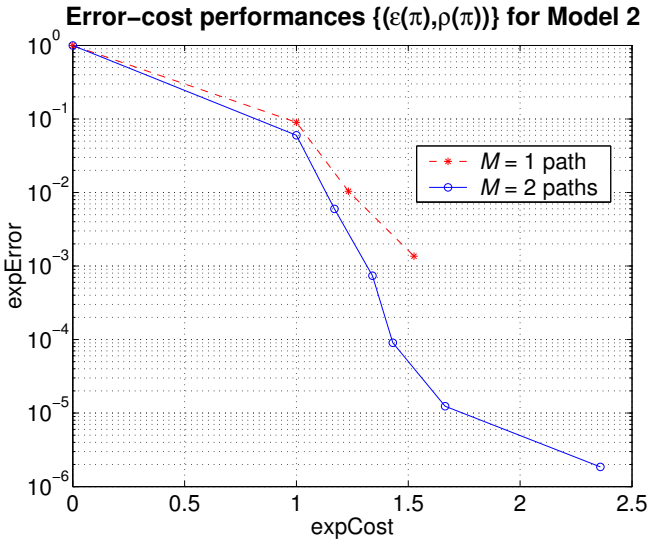


Fig. 5. Error-cost performance for streaming a data unit over $M = 1$ and $M = 2$ paths.

As mentioned before, path diversity provides a higher likelihood of delivering a data unit on time for the same transmission cost. This is illustrated in Figure 5 where we show the error-cost function for streaming over $M = 1$ and $M = 2$ network paths. We explained earlier that this function provides an insight into how efficiently a system trades off redundancy and error on the average for transmission of a single data unit. This ultimately affects the efficiency of the streaming system in terms of the provided trade-off between the amount of transmitted data and the resulting distortion at the client for the whole media presentation. In Figure 5 the functions are computed for the case of no history of previous transmissions and number of transmission opportunities $N = 3$. It can be seen that the error-cost function for $M = 2$ outperforms that for $M = 1$ over the whole range of available transmission costs. Therefore, on the average for $M = 2$ the streaming system transmits less bytes to achieve the same probability of delivering a data unit to the client on time. Note that the error-cost function for $M = 2$ has a longer tail than that for $M = 1$. This is expected as the space of transmission policies π for $M = 2$ is larger and includes policies that do not exist in the policy space of $M = 1$. Consider for example the policy π under which the data unit is transmitted simultaneously on the forward channel of both paths at every transmission opportunity. Certainly, this π has a larger expected cost and conversely a smaller expected error than any transmission policy for streaming the data unit over one of the paths only.

Next, we study the performance of the framework as a function of the quality of the network paths. As explained earlier depending on the state transition model, a path cah exhibit different levels of quality in terms of the loss and delay characteristics of packet transmissions. In Figure 6 we show the performance for streaming *Foreman* over $M = 1$ and over $M = 2$ network paths in case of Models 0, 1 and 3. It can be seen that streaming over two paths does not offer any advantages in case of Model 0. This is expected, as a path here does not switch between states and hence there is no need for an alternative routing of

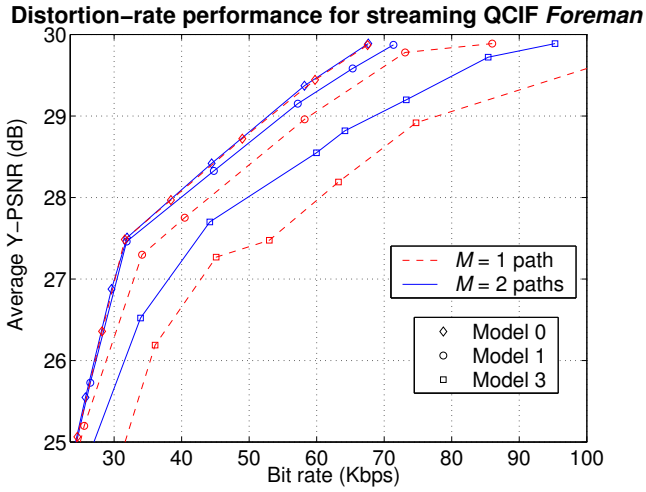


Fig. 6. Sender-driven R-D performance for *Foreman*, $M = 1, 2$ and different state transition models.

packets over another path. However, as the frequency of state transitions and the duration of stay in State 2 for a path increase on the average, the need for an alternative routing in order to avoid a bad quality path steadily increases. Thus, the difference in performance between $M = 2$ and $M = 1$ is largest when the state transitions on a path are governed by Model 3.

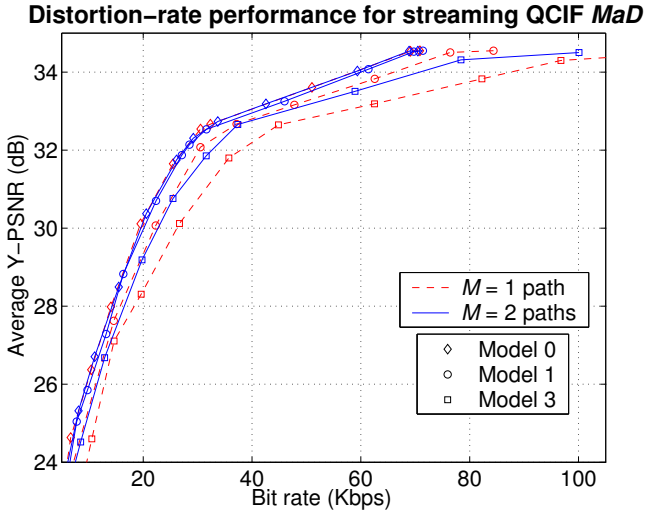


Fig. 7. Sender-driven R-D performance for *MaD* $M = 1, 2$ and different state transition models.

Similar performance is observed for streaming *MaD*, as shown in Figure 7. The only notable difference here is that the gains in performance for the case of $M = 2$ are not as significant as the corresponding gains for streaming *Foreman*, for each of the state transition models. As explained earlier, this is due to nature of the *MaD* video. Finally, it is interesting to note that the performance of $M = 2$ for Model 1 approaches that of $M = 1$ and $M = 2$ for Model 0, as seen from Figure 7.

B. Receiver-driven transmission

We observed that the performance of our framework in the case of receiver-driven transmission is analogous to that for the sender-driven case for each of the experiments conducted in Section V-A. Therefore, we present here only a smaller set of experimental results.

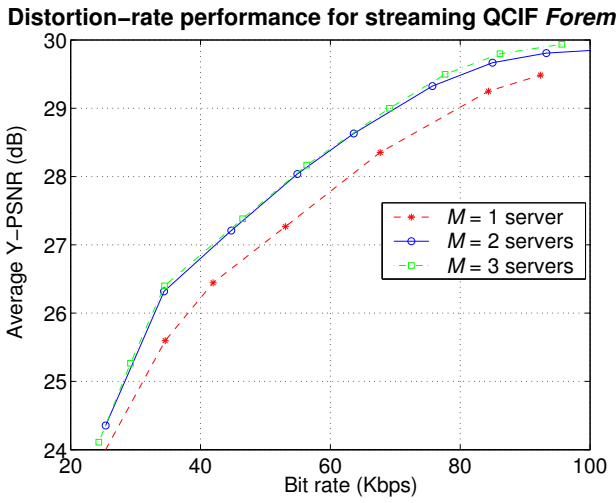


Fig. 8. Receiver-driven R-D performance for streaming *Foreman* over $M = 1, 2$ and 3 paths.

We examine first the performance of the proposed framework as a function of the number of servers available. It can be seen from Figure 8 that streaming *Foreman* from two servers can improve performance compared to the case of streaming from a single server. Model 2 is used to generate the state transitions in these experiments. An improvement is observed over the whole range of available rates. The gains in performance reach up to 0.6 dB in the range of transmission rates 30 - 90 Kbps and decrease as we move towards very low or very high rates. Similarly to the sender-driven case, the improved performance is due to the fact that streaming from two servers dramatically reduces the probability of not delivering the data units to the client on time. Furthermore, it can be seen from Figure 8 that using further servers for streaming does not provide additional significant gains in performance, since the performances of $M = 3$ and of $M = 2$ are almost identical. As the likelihood of not delivering the data units on time to the client is already quite reduced for $M = 2$, streaming from yet one more server does not provide further benefits.

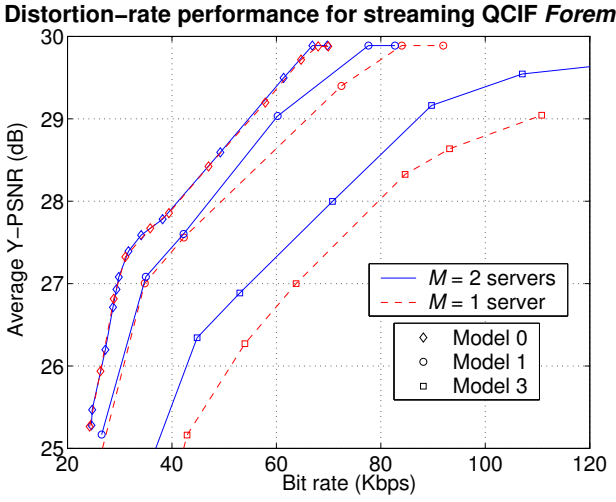


Fig. 9. Receiver-driven R-D performance for *Foreman* $M = 1, 2$ and different state transition models.

Next, we study the performance of the framework as a function of the quality of the network paths between the client and the servers. In Figure 9 we show the performance for streaming *Foreman* from $M = 1$ and from $M = 2$ media servers in case of Models 0, 1 and 3. It can be seen that streaming from two servers does not offer any advantages in case of Model 0. As explained earlier in Section V-A, a network path here does not switch between states and hence there is no need for streaming from another server. However, as we move from Model 0 towards Model 3 the need for streaming from an alternative server, in order to avoid communicating with a server over a network path with bad quality, steadily increases. Hence, the difference in performance between $M = 2$ and $M = 1$ is largest when the state transitions on a path are governed by Model 3.

VI. Conclusions

A framework has been presented that incorporates network path diversity in a rate-distortion optimized streaming of packetized media. The framework consists of 3 key components. The first one is the computation of the packet loss probabilities for a burst loss channel model that are used afterwards by the second component, which is an algorithm for sending a single data unit over multiple network paths. This algorithm achieves optimal error-cost performance by selecting the transmission policy π that minimizes the "per data unit" Lagrangian. Finally, the third component is an algorithm that jointly chooses the set of transmission options for all data units minimizing the Lagrangian $D + \lambda R$, while accounting for the effects of error concealment in case of packet loss. Using our framework a server can exploit the availability of multiple paths over which media packets can be transmitted in order to obtain an improved performance over the case when only a single path is used. Similarly, in a receiver-driven scenario, using the framework a client can request media packets from multiple servers again to obtain an improved performance over the case when only a single server is used. Experimental results for streaming different video contents demonstrate the benefit of using the proposed framework. The gains in performance are dependent on the quality of the network paths in terms of loss and delay. Finally, we show that in a sender-driven scenario exploiting diversity in a distortion rate optimization framework can bring the performance of a streaming system closer to that of an "ideal" distortion-rate optimal system.

References

[1] H. Poor and G. Wornell, *Wireless Communications: Signal Processing Perspectives*. Englewood Cliffs, NJ: Prentice Hall, 1998.

[2] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of internet path selection," in *Proc. Data Communication, Ann. Conf. Series*. Cambridge, MA, United States: ACM, Aug. 1999, pp. 289–299.

[3] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. Visual Communications and Image Processing*, vol. 4310. San Jose, CA: SPIE, Jan. 2001, pp. 329–409.

[4] J. Apostolopoulos and S. Wee, "Unbalanced multiple description video communication using path diversity," in *Proc. Int'l Conf. Image Processing*, vol. 1. Thessaloniki, Greece: IEEE, Oct. 2001, pp. 966–969.

[5] Y. J. Liang, E. Steinbach, and B. Girod, "Real-time voice communication over the Internet using packet path diversity," in *Proc. Int'l Conf. Multimedia*. Ottawa, Canada: ACM, Oct. 2001, pp. 431–440.

[6] ——, "Multi-stream voice over IP using packet path diversity," in *Proc. Workshop on Multimedia Signal Processing*. Cannes, France: IEEE, Oct. 2001, pp. 555–560.

[7] N. Gogate, D.-M. Chung, S. Panwar, and Y. Wang, "Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 9, pp. 777–792, Sept. 2002.

[8] T. Nguyen and A. Zakhor, "Distributed video streaming over internet," in *Proc. Multimedia Computing and Networking*, vol. 4673. San Jose, CA: SPIE, Jan. 2002, pp. 186–195.

[9] ——, "Distributed video streaming with forward error correction," in *Proc. Int'l Packet Video Workshop*, Pittsburg, PA, Apr. 2002.

[10] ——, "Protocols for distributed video streaming," in *Proc. Int'l Conf. Image Processing*, vol. 3. Rochester, NY, USA: IEEE, Sept. 2002, pp. 185–188.

[11] A. Majumdar, R. Puri, and K. Ramchandran, "Distributed multimedia transmission from multiple servers," in *Proc. Int'l Conf. Image Processing*, vol. 3. Rochester, NY, USA: IEEE, Sept. 2002, pp. 177–180.
[12] Y. Liang, E. Setton, and B. Girod, "Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," in *Proc. Workshop on Multimedia Signal Processing*. St. Thomas, US Virgin Islands: IEEE, Dec. 2002, pp. 420–423.
[13] E. Setton, Y. Liang, and B. Girod, "Adaptive multiple description video streaming over multiple channels with active probing," in *Proc. Int'l Conf. Multimedia and Exhibition*, vol. 1. Baltimore, MD: IEEE, July 2003, pp. 509–512.
[14] J. Apostolopoulos, T. Wong, W.-T. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proc. Infocom*, vol. 3. New York City, NY, USA: IEEE, June 2002, pp. 1736–1745.
[15] J. Apostolopoulos, W.-T. Tan, and S. Wee, "Performance of a multiple description streaming media content delivery network," in *Proc. Int'l Conf. Image Processing*, vol. 2. Rochester, NY, USA: IEEE, Sept. 2002, pp. 189–192.
[16] Y. Wang, S. Panwar, S. Lin, and S. Mao, "Wireless video transport using path diversity: multiple description vs. layered coding," in *Proc. Int'l Conf. Image Processing*, vol. 1. Rochester, NY: IEEE, Sept. 2002, pp. 21–24.
[17] A. Sehgal and P. A. Chou, "Cost-distortion optimized streaming media over DiffServ networks," in *Proc. Int'l Conf. Multimedia and Exhibition*, vol. 1. Lausanne, Switzerland: IEEE, Aug. 2002, pp. 857–860.
[18] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, 2001, submitted.
[19] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. Data Compression Conference*. Snowbird, UT: IEEE Computer Society, Mar. 2003, pp. 203–212.
[20] ——, "Server diversity in rate-distortion optimized streaming of multimedia," in *Proc. Int'l Conf. Image Processing*, vol. 3. Barcelona, Spain: IEEE, Sept. 2003, pp. 645–648.
[21] J. Madhavi and S. Floyd, "TCP-friendly unicast rate-based flow control," *Technical note sent to the end2end-interest mailing list*, Jan. 1997, http://www.psc.edu/networking/papers/tcp_friendly.html.
[22] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, vol. 27, no. 3, pp. 67–82, July 1997.
[23] T. Turletti, S. Parisis, and J.-C. Bolot, "Experiments with a layered transmission scheme over the Internet," INRIA, Sophia Antipolis, France, Tech. Rep. 3296, Nov. 1997.
[24] D. Sisalem and H. Schulzrinne, "The loss-delay adaptation algorithm: a TCP-friendly adaptation scheme," in *Proc. Network and Operating System Support for Digital Audio and Video (NOSSDAV)*. Cambridge, UK: ACM, July 1998.
[25] W.-T. Tan and A. Zakhor, "Internet video using error resilient scalable compression and cooperative transport protocol," in *Proc. Int'l Conf. Image Processing*, vol. 3. Chicago, IL: IEEE, Oct. 1998, pp. 458–462.
[26] R. Rejaie, M. Handley, and D. Estrin, "RAP: an end-to-end based congestion control mechanism for realtime streams in the Internet," in *Proc. Conf. on Computer Communications (INFOCOM)*, vol. 3. New York, NY: IEEE, Mar. 1999, pp. 1337–1345.
[27] S. Floyd, M. Handley, and J. Padhye, "Equation-based congestion control for unicast applications," International Computer Science Institute, Berkeley, CA, Tech. Rep. TR-00-03, Mar. 2000.
[28] Q. Zhang, Y.-Q. Zhang, and W. Zhu, "Resource allocation for audio and video streaming over the Internet," in *Proc. Int'l Symp. Circuits and Systems*, vol. IV. Geneva, Switzerland: IEEE, May 2000, pp. 21–24.
[29] Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *Draft ITU-T Recommendation H.263 Version 2*, Sept. 1997.
[30] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 1991.
[31] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoustics Speech and Signal Processing*, vol. 36, no. 1, pp. 1445–1453, Sept. 1988.
[32] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree structured source coding and modeling," *IEEE Trans. Information Theory*, vol. 35, no. 2, pp. 299–315, Mar. 1989.

Analysis of the LMS speed of convergence based on spectral information

Aaron E Flores

Abstract. The LMS algorithm is used to find the optimal minimum mean-squared error (MMSE) solutions for a wide variety of problems. Unfortunately, its convergence speed depends heavily on its initial conditions when the autocorrelation matrix R of its input vector has a high eigenvalue spread. In many applications such as system identification or channel equalization, R is Toeplitz. In this paper we exploit the Toeplitz structure of R to show that when the weight vector is initialized to zero, the convergence speed of LMS is related to the weighted average of its input PSD with weights given by the spectrum of the optimum solution. This result may allow for the prediction of the LMS algorithm performance in practice when approximate spectral information for the problem at hand is available.

Index Terms: LMS, LMS/Newton, convergence speed, efficiency, power spectrum, Wiener solution, Toeplitz.

I. Introduction

THE LMS algorithm is widely used to find the minimum mean-squared error (MMSE) solutions for a variety of linear estimation problems, and its efficiency has been extensively studied during the last decades [1]– [22]. The convergence speed of LMS depends on two factors: the eigenvalue distribution of the input autocorrelation matrix R, and the chosen initial condition. Modes corresponding to small eigenvalues converge much more slowly than those corresponding to large eigenvalues, and the initialization of the algorithm determines how much excitation is received by each of these modes. In practice, it is not known how the chosen initial conditions excite each of the modes, making the LMS speed of convergence difficult to predict. Our analysis of the convergence speed is concerned with applications where the input vector comes from a tapped delay line, inducing a Toeplitz structure in the R matrix. Using the Toeplitz nature of R, we show that the speed of convergence of the LMS algorithm can be estimated from its input power spectral density and the spectrum of the difference between the initial weight vector and the optimum solution. The speed of convergence of LMS was qualitatively assessed by comparing it to its ideal counterpart, the LMS/Newton algorithm, which is often used as a benchmark for adaptive algorithms [17], [18]. In section II we describe both the LMS and LMS/Newton algorithms and derive approximations to their learning curves on which our transient analysis is based. In section III we define the performance metric used to evaluate LMS speed of convergence, and in section IV we show how that metric can be estimated from spectral information. We illustrate our result with simulations in section V and summarize our conclusions and future work in Section VI.

II. The LMS and LMS/Newton algorithms

Adaptive algorithms such as LMS and LMS/Newton are used to solve linear estimation problems like the one depicted in Fig. 1, where the input vector $\boldsymbol{x}_k = [x_{1k}x_{2k}\cdots x_{Lk}]^T$ and desired response $d_k \in \Re$ are jointly stationary random processes, $\boldsymbol{w}_k = [w_{1k}w_{2k}\cdots w_{Lk}]^T$ is the weight vector, $y_k = \boldsymbol{x}_k^T\boldsymbol{w}_k$ is the output, and $\epsilon_k = d_k - y_k$ is the error. The Mean Square Error (MSE) is defined as $\xi_k = E[\epsilon_k^2]$ and it is a quadratic function of the weight vector. The optimal weight vector that minimizes ξ_k is given by $\boldsymbol{w}^* = \boldsymbol{R}^{-1}\boldsymbol{p}$, where $\boldsymbol{R} = E[\boldsymbol{x}_k\boldsymbol{x}_k^T]$ is the input autocorrelation matrix (assumed to be full rank), and $\boldsymbol{p} = E[\boldsymbol{x}_k d_k]$ is the crosscorrelation vector. The minimum MSE (MMSE) obtained using $\boldsymbol{w}^*$ is denoted by ξ^*.

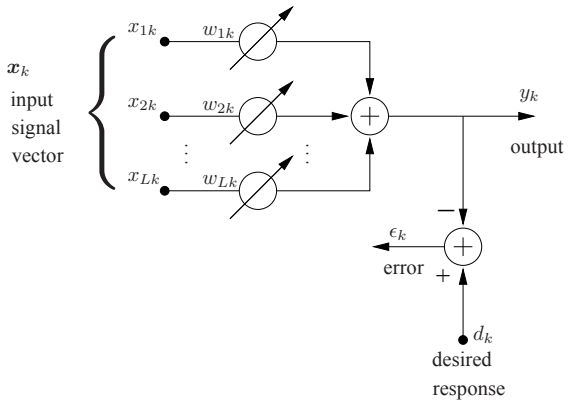


Fig. 1. Adaptive Linear Combiner

A. The LMS algorithm

Often in practice $\boldsymbol{R}^{-1}\boldsymbol{p}$ cannot be calculated due to the lack of knowledge of the statistics $\boldsymbol{R}$ and $\boldsymbol{p}$. However, when samples of $\boldsymbol{x}_k$ and d_k are available, they can be used to iteratively adjust the weight vector to obtain an approximation of $\boldsymbol{w}^*$. The simplest and most widely used algorithm for this is LMS [1]. It performs an instantaneous gradient descent adaptation of the weight vector:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + 2\mu\epsilon_k\boldsymbol{x}_k. \quad (1)$$

The step size parameter is μ and the initial weight vector $\boldsymbol{w}_0$ is arbitrarily set by the user. The MSE sequence ξ_k corresponding to the sequence of adapted weight vectors $\boldsymbol{w}_k$ is commonly known as the learning curve.

The author is with the Information Systems Laboratory at Stanford. Email: aflores@stanford.edu

Next we derive an approximation for ξ_k providing the basis of our speed of convergence analysis. Since $\boldsymbol{R}$ is symmetric, we can diagonalize it as $\boldsymbol{R} = \boldsymbol{Q}^T \boldsymbol{\Lambda} \boldsymbol{Q}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix of the eigenvalues λ_i of $\boldsymbol{R}$, and $\boldsymbol{Q}$ is a real orthonormal matrix with the corresponding eigenvectors of $\boldsymbol{R}$ as columns. Let $\boldsymbol{v}_k \triangleq \boldsymbol{w}_k - \boldsymbol{w}^*$ be the weight vector deviation from the optimal solution and $\boldsymbol{v}'_k \triangleq \boldsymbol{Q}^T \boldsymbol{v}_k$. Let $\boldsymbol{F}_k$ be the vector obtained from the diagonal of $E[\boldsymbol{v}'_k \boldsymbol{v}'^T_k]$, and $\boldsymbol{\lambda} \triangleq [\lambda_1 \lambda_2 \dots \lambda_L]^T$. Assuming the random processes $\{\boldsymbol{x}_k, d_k\}$ to be independent, it can be shown [3] that the MSE at time k can be expressed as

$$\xi_k = \xi^* + \boldsymbol{\lambda}^T \boldsymbol{F}_k. \tag{2}$$

If we further assume $\boldsymbol{x}_k$ to be gaussian, it was shown in [5] and [3] that $\boldsymbol{F}_k$ obeys the following recursion

$$\boldsymbol{F}_{k+1} = (I - 4\mu\boldsymbol{\Lambda} + 8\mu^2\boldsymbol{\Lambda}^2 + 4\mu^2\boldsymbol{\lambda}\boldsymbol{\lambda}^T)\boldsymbol{F}_k + 4\mu^2\xi^*\boldsymbol{\lambda}, \tag{3}$$

where $\boldsymbol{F}_k$ is shown [5] to converge if $\mu < \frac{1}{3\,\mathrm{Tr}\,(R)}$. This condition in μ allows us to approximate (3) by

$$\boldsymbol{F}_{k+1} \approx (I - 4\mu\boldsymbol{\Lambda})\boldsymbol{F}_k + 4\mu^2\xi^*\boldsymbol{\lambda}. \tag{4}$$

Denoting by $\mathbf{1} \in \Re^L$ a vector of ones and using (4) we arrive at

$$\boldsymbol{F}_k \approx (I - 4\mu\boldsymbol{\Lambda})^k(\boldsymbol{F}_0 - \mu\xi^*\mathbf{1}) + \mu\xi^*\mathbf{1}. \tag{5}$$

Substituting (5) in (2), we obtain the following approximation for the LMS learning curve

$$\xi_k \approx \xi_\infty + \boldsymbol{\lambda}^T(I - 4\mu\boldsymbol{\Lambda})^k(\boldsymbol{F}_0 - \mu\xi^*\mathbf{1}). \tag{6}$$

$$\text{where} \quad \xi_\infty \triangleq \lim_{k\to\infty} \xi_k \approx \xi^*(1 + \mu\,\mathrm{Tr}\,(\boldsymbol{R})). \tag{7}$$

A derivation of (6) under the more general conditions specified in [16] was done in [20].

B. The LMS/Newton algorithm

The LMS/Newton algorithm [6] is an ideal variant of the LMS algorithm that uses $\boldsymbol{R}$ to "whiten" its input. Although most of the time it cannot be implemented in practice due to the lack of knowledge of $\boldsymbol{R}$, it is of theoretical importance as a benchmark for adaptive algorithms [17], [18]. The LMS/Newton algorithm is the following

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + 2\mu\lambda_{\text{avg}}\boldsymbol{R}^{-1}\epsilon_k\boldsymbol{x}_k, \tag{8}$$

where $\lambda_{\text{avg}} \triangleq \frac{\sum_{i=1}^N \lambda_i}{n}$. It is well known [6] that the LMS/Newton algorithm is equivalent to the LMS algorithm with learning rate $\mu\lambda_{\text{avg}}$ and $\boldsymbol{x}_k$ previously whitened and normalized such that $\Lambda = I$. Therefore, assuming the same independence and gaussian statistics of $\boldsymbol{x}_k$ and d_k as with LMS, we can replace μ by $\mu\lambda_{\text{avg}}$, Λ by I, and $\mathbf{1}$ by $\boldsymbol{\lambda}^{-1}$ in (6) to obtain the following approximation for the LMS/Newton learning curve

$$\xi_k \approx \xi_\infty + (1 - 4\mu\lambda_{\text{avg}})^k\boldsymbol{\lambda}^T(\boldsymbol{F}_0 - \mu\lambda_{\text{avg}}\xi^*\boldsymbol{\lambda}^{-1}), \tag{9}$$

where the asymptotic MSE ξ_∞ is the same as for LMS. For a detailed derivation of (9), and under the more general conditions especified in [16], the reader is referred to [20].
Notice how the asymptotic MSE for both LMS and LMS/Newton is approximately the same when they use the same value for μ. This is an important fact we take advantage of in the next section.

III. LMS Transient Efficiency

Examining (6) and (9) we can see that the learning curve for LMS is a sum of geometric sequences (modes) with $1 - 4\mu\lambda_i$ as geometric ratios, whereas for LMS/Newton it consists of a single geometric sequence with geometric ratio $1 - 4\mu\lambda_{\text{avg}}$. If all the eigenvalues of $\boldsymbol{R}$ are equal, the learning curves for LMS and LMS/Newton are the same. Generally, the eigenvalues of $\boldsymbol{R}$ are not equal, and LMS consists of multiple modes, some faster than LMS/Newton and some slower than it as depicted in Fig. 2

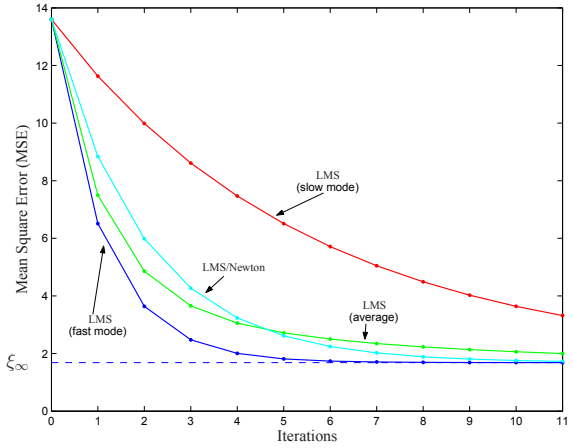


Fig. 2. LMS and LMS/Newton modes

To evaluate the speed of convergence of a learning curve we use the natural measure suggested in [5],

$$J \triangleq \sum_{k=0}^{\infty} \xi_k - \xi_\infty. \tag{10}$$

Small values of J indicate fast convergence and large values indicate slow convergence. Using (6) and (9) to obtain J for LMS and LMS/Newton respectively we get

$$J^{LMS} \approx \frac{\mathbf{1}^T\boldsymbol{F}_0 - \mu L\xi^*}{4\mu} = \frac{\boldsymbol{v}_0^T\boldsymbol{v}_0}{4\mu} - \frac{L\xi^*}{4}, \tag{11}$$

$$J^{LMS/Newton} \approx \frac{\boldsymbol{\lambda}^T\boldsymbol{F}_0 - \mu\lambda_{\text{avg}}L\xi^*}{4\mu\lambda_{\text{avg}}} = \frac{\boldsymbol{v}_0^T\boldsymbol{R}\boldsymbol{v}_0}{4\mu\lambda_{\text{avg}}} - \frac{L\xi^*}{4}. \tag{12}$$

It is clear from (11) and (12) that the larger μ is the faster the algorithms converge; however, equation (7) reminds us that we pay the price of a higher asymptotic MSE. Due to this trade-off between transient and steady state performance, it is costumary to compare the transient speed of these algorithms under the condition that both have the same steady state MSE, which is achieved when both algorithms use the same value for μ. Assuming this condition, we compare the speed of convergence of LMS to the one of LMS/Newton defining what we call the LMS Transient Efficiency

$$\text{LMS Tansient Efficiency} \triangleq \frac{J^{LMS/Newton}}{J^{LMS}}. \tag{13}$$

If the Transient Efficiency is bigger/smaller than one, then LMS performs better/worse than LMS/Newton in the sense of the J metric. Assuming that the initial weight vector $\boldsymbol{w}_0$ is far enough

from $\boldsymbol{w}^*$ such that $\boldsymbol{v}_0^T\boldsymbol{v}_0 \gg \mu L\xi^*$ and $\frac{\boldsymbol{v}_0^T\boldsymbol{R}\boldsymbol{v}_0}{\mu\lambda_{\text{avg}}} \gg \mu L\xi^*$, we obtain from (11), (12) and (13)

$$\text{LMS Transient Efficiency} \approx \frac{1}{\lambda_{\text{avg}}}\frac{\boldsymbol{v}_0^T\boldsymbol{R}\boldsymbol{v}_0}{\boldsymbol{v}_0^T\boldsymbol{v}_0}. \tag{14}$$

Since (14) is invariant to scaling of $\boldsymbol{v}_0$ and $\boldsymbol{R}$, we can assume without loss of generality $\|\boldsymbol{v}_0\| = \lambda_{\text{avg}} = 1$, obtaining the following compact expression:

$$\text{LMS Transient Efficiency} \approx \boldsymbol{v}_0^T\boldsymbol{R}\boldsymbol{v}_0. \tag{15}$$

This is the first contribution in this paper and it will be used to obtain an approximation of the LMS Transient Efficiency in terms of spectra in the next section.

IV. LMS Transient Efficiency in terms of spectral information

In this section we show that in the case that $\boldsymbol{R}$ is Toeplitz, the LMS Transient Efficiency can be expressed in terms of the Fourier spectrum of $\boldsymbol{v}_0$ and the input power spectral density. The Toeplitz structure of $\boldsymbol{R}$ arises from applications where the input vector $\boldsymbol{x}_k$ comes from a Tapped Delay Line (TDL) as shown in Fig. 3, i. e. $\boldsymbol{x}_k = [x_k\ x_{k-1}\ x_{k-2}\ \dots\ x_{k-L+1}]^T$, where x_k is a stationary scalar random process with autocorrelation sequence $\phi_{xx}[n] \triangleq E[x_k\, x_{k+n}]$. The input autocorrelation matrix is therefore given by $\boldsymbol{R}_{k,l} = \phi_{xx}[k-l]$.

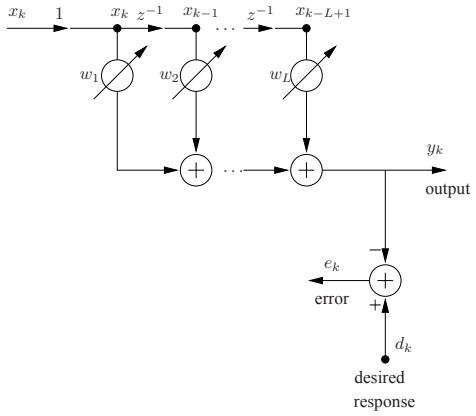


Fig. 3. Tapped Delay Line

We define the Input Power Spectral Density vector $\boldsymbol{\Phi}_{\boldsymbol{xx}} \in \Re^M$ as the M uniformly spaced samples of the DTFT of the N-point truncation of the input autocorrelation function $\phi_{xx}[n]$, i. e.

$$\boldsymbol{\Phi}_{\boldsymbol{xx}}[m] \triangleq \sum_{n=-N+1}^{N-1} \phi_{xx}[n]e^{-2\pi j\frac{mn}{M}} \tag{16}$$
$$m = 0, 1, \dots M-1, \quad N \geq L, \quad M \geq N+L-1.$$

In order to show how $\boldsymbol{\Phi}_{\boldsymbol{xx}} \in \Re^M$ is related to the LMS Transient Efficiency, the following Lemma will be needed

Lemma 1 (Spectral Factorization of $\boldsymbol{R}$): Let $\boldsymbol{\Psi}$ be a $M \times M$ diagonal matrix with $\boldsymbol{\Psi}_{m,m} = \boldsymbol{\Phi}_{\boldsymbol{xx}}[m]$ as its diagonal entries, and let $\boldsymbol{U}$ be a $L \times M$ matrix with elements $\boldsymbol{U}_{l,m} = \frac{1}{\sqrt{M}}e^{2\pi j\frac{ml}{M}}$ with $0 \leq l \leq L-1$ and $0 \leq m \leq M-1$, then $\boldsymbol{R}$ can be factored in the following way

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{U}^*. \tag{17}$$

Proof: Let $\boldsymbol{T} \triangleq \boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{U}^*$, then

$$\boldsymbol{T}_{k,l} = \sum_{m=0}^{M-1} \frac{1}{\sqrt{M}}e^{2\pi j\frac{km}{M}}\boldsymbol{\Psi}_{m,m}\frac{1}{\sqrt{M}}e^{2\pi j\frac{-ml}{M}} \tag{18}$$
$$= \frac{1}{M}\sum_{m=0}^{M-1} \boldsymbol{\Phi}_{\boldsymbol{xx}}[m]e^{2\pi j\frac{m(k-l)}{M}} \tag{19}$$
$$= \frac{1}{M}\sum_{m=0}^{M-1}\sum_{n=-N+1}^{N-1} \phi_{xx}[n]e^{-2\pi j\frac{mn}{M}}e^{2\pi j\frac{m(k-l)}{M}} \tag{20}$$
$$= \sum_{n=-N+1}^{N-1} \phi_{xx}[n]\frac{1}{M}\sum_{m=0}^{M-1} e^{-2\pi j\frac{m(k-l-n)}{M}} \tag{21}$$
$$= \sum_{n=-N+1}^{N-1} \phi_{xx}[n]\delta[k-l-n] \tag{22}$$
$$= \phi_{xx}[k-l] = \boldsymbol{R}_{k,l} \tag{23}$$

where the step from (21) to (22) was done using the following inequalities: $-L+1 \leq k-l \leq L-1$, $-N+1 \leq n \leq N-1$, and $M \geq N+L-1$. ■

Let $\boldsymbol{V}_0 \triangleq \boldsymbol{U}^*\boldsymbol{v}_0$ be the M-point DFT of $\boldsymbol{v}_0$, i. e.

$$\boldsymbol{V}_0[m] \triangleq \frac{1}{\sqrt{M}}\sum_{n=0}^{L-1} \boldsymbol{v}_0[n]e^{-2\pi j\frac{mn}{M}} \quad m = 0, 1, \dots M-1, \tag{24}$$

Using Lemma 1 and (24) we obtain

$$\boldsymbol{v}_o^T\boldsymbol{R}\boldsymbol{v}_0 = \boldsymbol{v}_o^T\boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{U}^*\boldsymbol{v}_0 = \sum_{m=0}^{M-1} \Phi_{xx}[m]|\boldsymbol{V}_o[m]|^2, \tag{25}$$

furthermore, inserting (25) into (15), we arrive to the following approximation for the LMS Transient Eficciency in terms of the input psd and the spectrum of $\boldsymbol{v}_o$

$$\text{LMS Transient Efficiency} \approx \sum_{m=0}^{M-1} \Phi_{xx}[m]|\boldsymbol{V}_0[m]|^2. \tag{26}$$

An analagous expression to (26) in terms of continuous Fourier spectra can be also obtained [20]. We should note that the independence assumption under which the approximation in (15) derived in this paper is violated when a tapped delay line is used. To avoid this problem, the more general conditions specified in [16] (which can be applied to tapped delay lines) are used in [20] to derive (6) and (9) on which (26) relies on.

Since $\boldsymbol{v}_0$ and $\boldsymbol{R}$ are scaled such that $\|\boldsymbol{v}_0\| = \lambda_{\text{avg}} = 1$, we can show that

$$\frac{\sum_{m=0}^{M-1}\Phi_{xx}[m]}{M} = 1, \quad \sum_{m=0}^{M-1}|\boldsymbol{V}_0[m]|^2 = 1. \tag{27}$$

These normalizations on $\Phi_{xx}[m]$ and $|\boldsymbol{V}_0[m]|^2$ allow us to interpret the right hand side of (26) as a weighted average of $\Phi_{xx}[m]$ where the weights are given by $|\boldsymbol{V}_0[m]|^2$. Therefore, if $\Phi_{xx}[m]$ is large in the frequency ranges where $|\boldsymbol{V}_0[m]|^2$ is also large, the LMS Transient Efficiency will likely be bigger than one, implying that LMS will outperform LMS/Newton under the J metric. On the other hand, if $\Phi_{xx}[m]$ tends to be small in the frequency ranges where $|\boldsymbol{V}_0[m]|^2$ is large, the LMS Transient Efficiency will likely be less than one indicating that LMS will perform worse than LMS/Newton in the J metric sense.

An important application of (26) is to the special case when the weight vector is initialized to zero $\boldsymbol{w}_o = 0$, hence $\boldsymbol{v}_o = -\boldsymbol{w}^*$, which results in

$$|\boldsymbol{V}_0[m]|^2 = |\boldsymbol{W}^*[m]|^2, \tag{28}$$

where $\boldsymbol{W}^*[m]$ is the M-point Discrete Fourier Transform of the optimal or Wiener solution $\boldsymbol{w}^*$ as defined in (24). Therefore, when the weight vector is initialized to zero, the speed of convergence of the LMS algorithm can be estimated by the weighted average of the input psd with the weights given by the spectrum of the Wiener solution. This observation can be very useful for system identification applications, where prior approximate knowledge of the input psd and frequency response of the plant to be identified may be used to estimate LMS's performance before implementing it. For example, if the plant has a low-pass frequency response and the input psd has very little power in the frequency range above the cut-off frequency, LMS will converge very fast. On the other hand, if the plant has a high-pass nature and the input psd is small for high frequencies, LMS will be very slow. In channel equalization applications the spectrum of the optimum solution is the opposite of the channel frequency response; hence, if we initiliaze the weight vector to zero, (26) will be small implying a slow convergence of LMS, a phenomenom often observed when using LMS in equalization or adaptive inverse control problems. We illustrate these ideas with two examples in the next section.

V. Simulations

Consider the system identification problem depicted in Fig. 4. The additive plant noise is independent of the input. Both, the adaptive filter and the plant consists of 16 taps, so the spectrum of the optimum solution coincides with the plant frequency response. The weight vector is initialized to zero and adapted for 800 iterations. The learning rate is set so that $\mu \,\mathrm{Tr}\,(R) = 0.05$. The power spectrum of the input and the frequency response of the plant to be identified are shown in Fig. 5, where a value of $M = 200$ was used to calculate $\Phi_{xx}[m]$ and $|\boldsymbol{W}^*[m]|^2$. The eigenvalue spread for this exercise was 5709. By inspection of the input psd and the frequency response of the plant we expect the LMS Transient Efficiency to be bigger than one, indicating that LMS should perform better than LMS/Newton, which is confirmed by their learning curves shown in Fig. 6. The estimate for the LMS Transient efficiency obtained from the simulation was 2.064, very close to the approximation of 2.069 given by (26).

To illustrate what happens if we use zero initial conditions when the input psd and the Wiener solution spectrum are very distinct, consider the simple equalization problem depicted in Fig. 7. The input to the channel is a white signal, hence the input psd for the adaptive filter is given by the magnitude squared of the channel frequency response. The channel frequency response and spectrum of the wiener solution for the adaptive equalizer are shown in Fig. 8 where a value of $M = 200$ was used. The eigenvalue spread of the input to the adaptive filter was 75.4. As intuitively expected, the input psd and Wiener spectrum have opposite shapes resulting in a small LMS Transient Efficiency,indicating that LMS should perform worse than LMS/Newton, which was corroborated by their learning curves shown in Fig. 9. The estimate for the LMS Transient

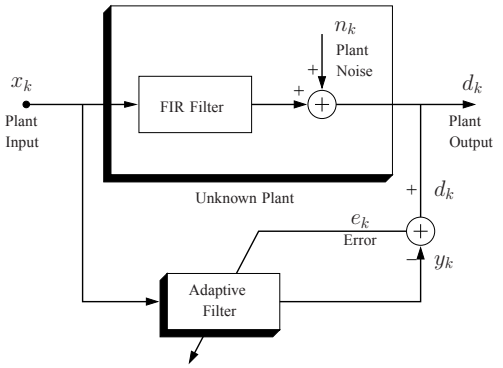


Fig. 4. System Identification Example

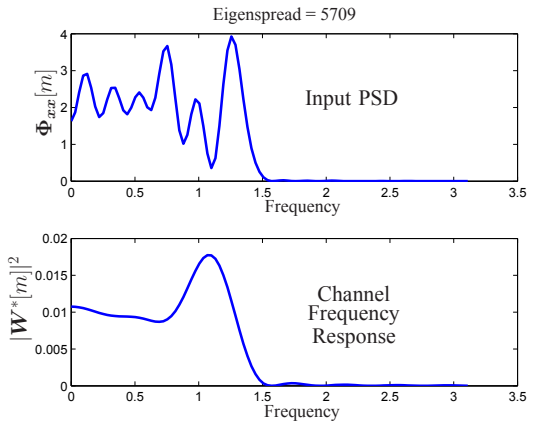


Fig. 5. Input Power Spectrum and Plant Frequency Response

efficiency obtained from the simulation was 0.132, very close to the approximation of 0.137 given by (26).

VI. Conclusions and Future Work

We have shown that when the LMS algorithm is used to train a transversal adaptive filter with jointly stationary input and desired response signals, its transient performance can be assesed by the inner product between the input psd and the spectrum of the initial weight vector deviation from the Wiener solution. This implies that when the initial weight vector is set to zero, the transient efficiency of the LMS algorithm can be predicted from approximate knowledge of the input psd and the frequency response of the Wiener solution. We described how this theory can be applied to system identification and equalization tasks; however, our results can be useful in predicting the LMS performance in any application where the input autocorrelation matrix is Toeplitz and spectral information about the input and optimum solution is available.

In [20], an extension of this analysis is made to certain nonstationary conditions, where the spectral characteristics of the input and the spectrum of the changes of the optimum solution keep a general shape, say both being always low pass. A similar analysis based on the Mean Square Deviation (MSD) instead of the MSE is also done in [20].

It is currently investigated the application of these results to a variety of applications such as adaptive echo cancellation,

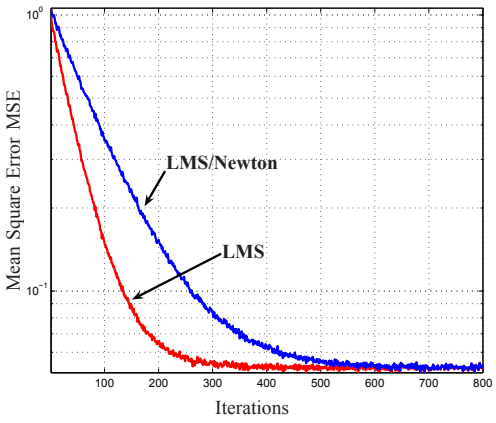


Fig. 6. Learning curves for System Identification example

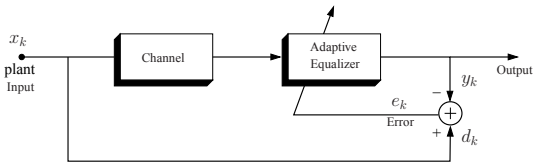


Fig. 7. Equalization Example

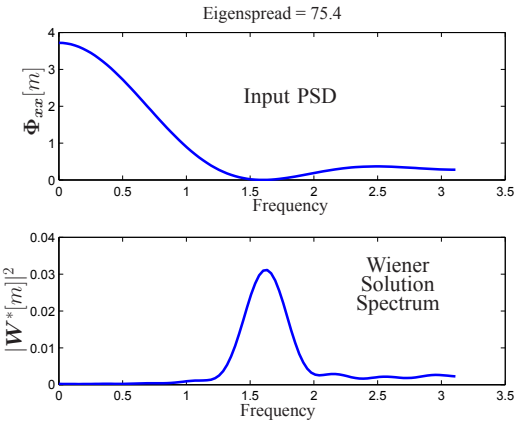


Fig. 8. Frequency response of plant and optimum equalizer

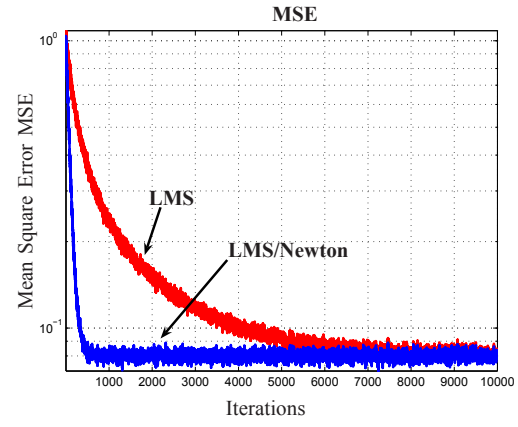


Fig. 9. Learning curves for equalization example

the adaptive line enhancer, adaptive noise cancelling, adaptive inverse control, etc, where further insight can be gained on the performance of the LMS algorithm. Adaptive prediction is of particular interest, since in such application, the approximate relantionship between the input psd and the spectrum of the Wiener solution is likely to be known a priori due to the nature of the problem.

References

[1] B. Widrow and M. E. Hoff, Jr, "Adaptive switching circuits," in *IRE WESCON Con. Rec., pt. 4*, 1960, pp. 96–104.

[2] B. Widrow, J. M. McCool, M. G. Larimore, and J. C. R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter.," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151 – 62, Aug. 1976.

[3] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, no. 3, pp. 722–736, June 1981.

[4] B. Widrow and E. Walach, "On the statistical efficiency of the lms algorithm with nonstationary inputs.," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 211 – 221, 1984.

[5] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, no. 1, pp. 222–229, Feb. 1985.

[6] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 1985.

[7] F. M. Boland and J. B. Foley, "Stochastic convergence of the lms algorithm in adaptive systems.," *Signal Processing*, vol. 13, no. 4, pp. 339 – 52, DEC 1987.

[8] B. F. Boroujeny, "On statistical efficiency of the lms algorithm in system modeling.," *IEEE Transactions on Signal Processing*, vol. 41, no. 5, pp. 1947 – 51, MAY 1993.

[9] D. R. Morgan, "Slow asymptotic convergence of lms acoustic echo cancelers.," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 2, pp. 126 – 36, MAR 1995.

[10] S. C. Douglas and W. M. Pan, "Exact expectation analysis of the lms adaptive filter.," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2863 – 71, DEC 1995.

[11] B. Hassibi, A. H. Sayed, and T. Kailath, "h^∞ optimality of the lms algorithm.," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 267 – 80, FEB 1996.

[12] M. Reuter and J. R. Zeidler, "Nonlinear effects in lms adaptive equalizers.," *IEEE Transactions on Signal Processing*, vol. 47, no. 6, pp. 1570 – 9, JUN 1999.

[13] K. J. Quirk, L. B. Milstein, and J. R. Zeidler, "A performance bound for the lms estimator.," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 1150 – 8, MAY 2000.

[14] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyses of adaptive filters.," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 314 – 24, FEB 2001.

[15] H. J. Butterweck, "A wave theory of long adaptive filters.," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 6, pp. 739 – 47, JUN 2001.

[16] O. Dabeer and E. Masry, "Analysis of mean-square error and transient speed of the LMS adaptive algorithm," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1873–1894, July 2002.

[17] B. Widrow and M. Kamenetsky, "Statistical efficiency of adaptive algorithms.," *Neural Networks*, vol. 16, no. 5/6, pp. 735 – 44, JUN-JUL 2003.

[18] S. Haykin and B. Widrow, *Least-mean-square adaptive filters*, Wiley-Interscience, Hoboken, N.J., 2003.

[19] M. I. Troparevsky and C. E. D'Attellis, "On the convergence of the lms algorithm in adaptive filtering.," *Signal Processing*, vol. 84, no. 10, pp. 1985 – 8, October 2004.

[20] A. Flores, *Spectral Effects on the Rate of Convergence of the LMS Adaptive Algorithm*, Ph.D. thesis, Stanford University, December 2005.

[21] M. Kamenetsky, *Accelerating the Convergence of the LMS Adaptive Algorithm*, Ph.D. thesis, Stanford University, June 2005.

[22] K. Mayyas, "Performance analysis of the deficient length lms adaptive algorithm.," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2727 – 34, AUG 2005.

Evaluating Scheduling Algorithms on Distributed Computational Grids

Ryan J. Wisnesky

Abstract. This paper presents the results of a simulation study of a heterogeneous computational grid using different scheduling algorithms. After a definition of robustness based on the concept of work completion latency is discussed, a method to simulate grids based on Estimated Time to Compute matrices is presented. Three well-known scheduling algorithms are then evaluated against each other, and the highest-performing scheduler is then analyzed in detail. The notion of ETC perturbation is presented, and this high-performing scheduling algorithm is found to be relatively robust against uncertainties in estimated task completion times.

Index Terms: grid computing, scheduling, heterogeneous systems, distributed systems

I. Introduction

COMPUTATIONAL grids are becoming more prevalent as the cost of bringing together disparate computing resources declines. However, a number of challenges remain before these grids can be utilized efficiently. This paper explores the results of using several well-known scheduling algorithms to schedule work on a grid under probabilistic work arrival rates and varying task completion times.

First, we give a definition of computational grids and argue that robustness is an important feature of economical grid computing. Then we proceed to develop a rigorous definition of robustness based on the concept of work completion latency. With this definition as our basis, we present a method for modeling computational grids and describe a software simulation framework we developed to analyze different scheduling algorithms under a variety of workloads. We then describe three scheduling algorithms and give the results of two experiments: the first to investigate the performance of the three algorithms relative to each other, and the second to investigate the effects of variation in work completion times on a specific scheduler.

A. What are Grids?

The definition of a *computational grid* is still a subject of some debate. What follows here is a short definition of a computational grid sufficient to give an adequate background for the rest of the paper; for a more in-depth definition, see [4]. Let us begin by giving an intuitive definition of what we mean by a computational grid: a computational grid is a collection of *nodes*, each of which may be thought of as a system that can perform work and has access to a network. Many systems share this property, including computer clusters. However, a grid is unique in that nodes on the grid vary in capability, and that the grid may provision more nodes to do work, or release nodes from the grid at any time.

In addition to these unique properties, the grid also has several properties stemming from its distributed nature, namely, that nodes may fail (become unable to perform work due to software or hardware problems) at any time, and communication efficiency between nodes can vary widely. In addition, nodes are typically geographically widely separated (i.e., across continents) and are typically owned by different entities. Rather than completely "owning" a node, a grid may use excess computing capacity from machines also doing other work to utilize spare computational power (for instance, the SETI@Home project uses a similar approach). A computational grid, then, can be seen as an adaptive system that provisions extra computational capacity as demand requires or as machines fail, and assigns work to nodes where the work can be done most effectively.

As promising as such a description of a full-fledged grid sounds, a number of obstacles remain before systems that have all the above properties can even be constructed, much less utilized effectively. For our purposes, we will study computational grids that are one step removed from their cluster counterparts: our grids will not expand or contract over time, nor will nodes fail, nor will there be significant communication delays between nodes; however, our grids will be made of nodes of heterogeneous capability. By studying grids of this type, we hope to enable an evolutionary approach to studying more complex grids.

B. Costs and Benefits

Why would anyone consider using a computational grid instead of a more traditional system to solve a particular problem? Simply put, grids have one main advantage over their cluster cousins: the provisioning ability of the grid enables more cost-effective solutions. The advantage stems from both the ability of a grid to use only sufficient computing capacity as demands require and the ability of a grid to use specialized hardware. Consider the case of a grid that needs to process a task involving massive amounts of vector math: the grid could provision a computer with specialized hardware to solve the problem quickly.

However, there are also certain costs to using grids. Especially in an industrial setting, there are several problems that must be overcome before grids can be adopted for widespread use. Namely:

- Given the heterogeneous and transient nature of resources on the grid, is there any way to ensure a certain level of minimum throughput, or *robustness*, against variability in the environment, like node failures?
- How is the cost of provisioning extra resources balanced against the benefit of having extra resources?

This was part of the author's CS191 project. Email: wisnesky@cs.stanford.edu

- How is amount of data that must flow between nodes minimized?
- How does the grid make decisions about where to schedule tasks?

Because we are studying grids that do not provision extra nodes, do not have nodes that fail, and have infinitely fast communication links, the question we are investigating is this: *how are tasks scheduled to a grid of heterogeneous resources to ensure a certain level of robustness?*

Ian Foster, of distributed computing fame, goes so far as to believe that a grid is not really a grid unless it is robust. In [3], one of his three requirements for a grid is that it

> *...delivers nontrivial qualities of service.* (A grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than the sum of its parts.) [Parentheses are Foster's]

Creating a robust scheduling mechanism, even for our case of limited grids, is beyond the scope of this paper. Rather, here we will discuss the performance of scheduling algorithms under assumptions of node heterogeneity, probabilistic task arrival rates, and uncertain task completion times. By investigating well-known algorithms, we can find a certain base level of robustness that specialized scheduling algorithms should be able to improve upon. This analysis, and its accompanying simulation framework, is offered as a baseline for future evolutionary steps toward constructing robust schedulers for more complex grids.

II. Problem Setup

For the rest of this paper, we may assume that a grid refers to a member of the special class of grids defined above that lacks any type of dynamic provisioning: all nodes on the grid are allocated at startup and remain that way throughout the lifetime of the grid. Furthermore, we may assume that all the communication links between nodes are reliable, and that communication latency is zero and bandwidth infinite.

Define a *path* to be an ordered sequence of *applications* $\langle A_1, \ldots, A_N \rangle$ such that the output from application A_{n-1} flows to application A_n. Intuitively, we can think of tasks as consisting of several pipeline stages, where the output from one pipeline stage flows directly into exactly one next stage, and each stage receives input from exactly one prior stage. In practice, each application has a different degree of parallelism, meaning that some applications may begin executing before all the data from the previous application is available. However, for the purposes of our simulation, we assume that there is no such overlap: a path's application A_n must finish before its next application A_{n+1} can start. An example of a task could be the multiplication of ten matrices, where each application multiplies the result from its previous application by a matrix and then feeds the result to the next application. Note that an application is the smallest unit of work that must be run completely on one node.

Define a *job* or *task* to consist of a path, an *initial data resource* (the initial data for the first application in the path), and a *customer*. The term "customer" is used to partition the set of jobs into separate classes which can then be used to develop a system performance feature. The term "customer" comes from the industry perspective of customers submitting work, but in reality any other sort of identifier could be used, or a single customer could be used. Also note that for convenience we often speak of a job's applications rather than a job's path's applications.

The grid's *scheduler* accepts jobs and distributes their constituent applications among nodes. For our purposes, we are assuming a central, omniscient scheduler that may schedule work as it becomes available; in practice, such a scheduler might be a distributed algorithm without access to all the information our scheduler has. Investigating robust, distributed schedulers is one area for future research.

A. Toward a Definition of Robustness

A definition of *robustness* makes sense only relative to a perturbation parameter. In other words, a grid might be robust against uncertainty in job completion times, or robust against uncertain work arrival rates, meaning that variations, or uncertainty, in these estimated times or changing work arrival rates do not severely affect the grid's *performance feature* (thus, robustness is necessarily grid and customer specific, because "severely" is a relative term). The performance feature is some performance metric on the grid as it is running. For instance, one interesting performance feature might be the number of jobs completed per second; another might be the amount of data processed.

1) Our Performance Feature: Latency: To evaluate robustness, we must choose our performance feature first. We proceed to give a formal definition of our performance feature for this simulation; informally, our performance feature is the percentage of jobs that "do not take too long" to finish.

Order the set of all jobs so that $job(j)$ denotes the j-th job in the ordering. Then, define $start(j,i)$ to be the time that the i-th application of $job(j)$ begins executing and define $finish(j,i)$ to be the time that the i-th application ceases executing. We will assume that

$$start(j,i) < finish(j,i)$$

and that

$$finish(j,i) \leq start(j,i+1)$$

although some realistic paths might violate the second inequality (for instance, an application that only needs the first part of the previous application's data might conceivably behave in this manner). Finally, define $finish(j, |job(j)|)$ to be the execution time of the entire job, and note that it is equal to the completion time of the last application in the job's path. Define the *latency* $L(j)$ of job j with its initial resource r_{init} to be

$$L(j) = finish(j, |job(j)|) - arrival(r_{init})$$

where $arrival(r_{init})$ denotes the time that the initial resource is first available. Let $L_c^{max}(j)$ denote the maximum acceptable latency $L(j)$ of $job(j)$ for customer c (this number, specified by the customer, simply denotes how long the customer is willing to wait for this job to complete. This notion of acceptable latency is a key component of the performance feature: for modeling, we can assume it to be a parameter, but in practice, this number

would likely be defined by the customer as part of a service level agreement with the company providing the grid). Then, we define $J_c(t)$ to be the set of all jobs that arrive at time t for customer c. We also define the helper function

$$1(x, y) \equiv \begin{cases} 1 & \text{if } x > y \\ 0 & \text{otherwise.} \end{cases}$$

Now, we define $\alpha_c(t)$ to be the *job latency failure rate*, meaning that

$$\alpha_c(t) = \sum_{\forall j \in J_c(t)} 1\big(L(j), L_c^{max}(j)\big)$$

Intuitively, $\alpha_c(t)$ is the number of jobs, submitted by customer c at time t, that are taking too long to finish. The *performance feature* $f(T)$ of the grid over a discrete interval T is then simply

$$f(T) = \sum_{\forall c} \sum_{\forall \tau \in T} \alpha_c(\tau)$$

Intuitively, the performance feature of a grid is the number of jobs missing their acceptable latencies. Generally, we will divide $f(T)$ by the number of jobs arriving during the interval T to obtain the performance as a more useful percentage.

2) Adequate Performance: Before we can define what we mean for our grids to be robust against variations in a perturbation parameter, we must first define what it means for the grid to be *performing adequately*. We will then say that the grid is robust when it still performs adequately when probabilistic variations in the environment, expressed through the perturbation parameter, are introduced.

Let the function $\beta_c^{max}(t)$ be the *maximum job latency failure rate* for customer c for the jobs submitted at time t. This number, specified by each customer, indicates the number of jobs they are willing to accept taking longer than their maximum acceptable latency at a given time. In practice, this will probably be specified as a percentage, but without loss of generality we can assume it to be an integer. This quantity necessarily varies with time because it could be the case that a customer is willing to accept more failures at certain times than others. The *adequate performance metric* $\delta_c(t)$ for customer c is then defined as

$$\delta_c(t) = \beta_c^{max}(t) - \alpha_c(t)$$

The system is *performing adequately* when for every customer c, and at every moment of time t, $\delta_c(t)$ is positive. Intuitively, a system is performing adequately when, at every moment, each customer is satisfied that not more than some given percentage of their jobs are taking too long.

Of course, this definition of adequate performance is a boolean condition: either the system is adequately performing or it is not. We can use the adequate performance metric $\delta_c(t)$ to create other, more nuanced definitions of adequate performance; a number of examples come readily to mind. We could say that a system is maximally adequately performing when $\delta_c(t)$ is always the maximum possible over any t, or that a system is averagely adequately performing when the average value of $\delta_c(t)$ over the system lifetime is greater than zero. Investigating applications of the adequate performance metric and associated performance features is one of the important directions for future work.

Note that for our simulation, we report the performance features for grids rather than their adequate performance. We do this because the notion of adequate performance is tied to the customer-specified maximum job latency failure rate; the definition of the performance feature is not. This definition of adequate performance is offered as the basis for a definition of adequate performance that might be used in practice as the foundation of a service level agreement by the grid provider and customer.

3) Our Robustness: As stated earlier, a system must be robust relative to some perturbation parameter. There are many such parameters, including

- node failures
- variations in job arrival rates
- variations in estimated job completion times
- time (i.e., a robust system's performance does not vary with time)

We have chosen variation in job completion times as our perturbation parameter for this simulation. The simulation framework itself also could support an evaluation of robustness against job arrival rates; this is another area for future work.

B. Calculation of ETC Values

In order to evaluate the robustness of a scheduling algorithm against variation in estimated job completion times, we must first find a way to model these completion times. We use the method described in [5] to model the application execution times for our simulation. A brief synopsis of the relevant points, along with our modifications to the method, is described in this section.

1) ETC Matrices: The term *ETC value* is short-hand for "**E**stimated **T**ime to **C**ompute value." An ETC value represents the amount of time that an application needs to run on a given node in order to complete – every ETC value must be relative to some node; there is no notion of how long an application takes in the abstract. An *ETC matrix* is a matrix that has an ETC value for every application (the rows) and node (the columns). This definition can best be described by an example; see Table I. Note that because of the requirement that every entry in the matrix have a value, we are assuming that every application can be run on every node. Extending this method to take into account applications that may only run on certain nodes is an interesting extension.

TABLE I
AN EXAMPLE ETC MATRIX FOR A PATH WITH 7 APPLICATIONS ON A GRID OF 5 NODES

	n_1	n_2	n_3	n_4	n_5
a_1	21	7	4	5	9
a_2	6	14	11	18	9
a_3	17	13	13	15	11
a_4	8	19	12	16	15
a_5	13	5	9	10	7
a_6	10	12	13	21	5
a_7	4	15	17	11	9

2) Creating ETC Matrices: [5] gives a generous development of the procedure we use to generate ETC matrices. In essence, the procedure used to create an ETC matrix for a given path takes as input parameters

Mutask measures the central tendency of the ETC values for applications on "average" nodes
Vtask measures the variation of ETC values from Mutask
Vmach measures node heterogeneity

The procedure then uses the parameters to create several gamma distributions, which are then sampled to create the values in the matrix. The algorithm is given in Figure 1.

Fig. 1. The CVB Method for generating ETC Matrices. Summarized from [5].

```
Inputs: Vtask, Vmach, Mutask
Outputs: e[][] // the ETC matrix
Begin:
  Let Atask = 1/(Vtask*Vtask)
  Let Amach = 1/(Vmach*Vmach)
  Let Btask = Mutask/Atask
    For each task i:
      Let q = Gamma(Atask, Btask)
      Let Bmach = q/Amach
      For each node j:
        e[i,j] = Gamma(Amach, Bmach)
```

This method is capable of generating ETC matrices that have high-task, high-machine heterogeneity; high-task, low-machine heterogeneity; and low-task, low-machine heterogeneity. Unfortunately, this method cannot generate low-task, high-machine heterogeneity matrices. A related algorithm can generate exactly those cases that the original algorithm cannot supply. However, for our simulation, we will be using medium-task, medium-machine heterogeneity matrices.

Unfortunately, this method does not generate *consistent* ETC matrices. A consistent matrix has the property that *if a node n has a lower ETC value than a node m for any task, then the same is true for all tasks*. This is a desirable property is many situations; as the authors of [5] discuss, an Intel Pentium III would likely be faster than an Intel 286 for every conceivable task. There is a method for extracting consistent matrices from inconsistent matrices; however, because the generated matrices tend to be *partially consistent*, meaning that they have at least one consistent sub-matrix, and because inconsistent and partially-consistent matrices tend to occur in environments where tasks have vastly different computational needs (i.e., some are floating-point intensive, others need multiple threads, etc.), we feel that using inconsistent matrices is justified.

Our extension of the method given in [5] is to simply apply the method to generate application ETC values for paths. Thus, a path *is* an ETC matrix.

C. Scheduling

The grid scheduler assigns applications to nodes. Each node can run one application at a time, and must run that application to completion. (By splitting multi-cpu machines into a set of single-cpu nodes, powerful machines may in fact run more than one job at a time.) The scheduler maintains a queue of work for each node, and may re-order the queue at any time. An application that is being run but whose previous application has not finished will wait until the previous application has finished before starting. (This is a simplifying assumption: in reality, some applications may begin before their previous application has finished. For instance, a sequence of applications that are streaming data to each other has this property.)

The scheduler is invoked when *mapping events* occur. During a mapping event, the scheduler may reorder each node's work queue (thereby *mapping* applications to nodes), and assigns new jobs that have arrived to various nodes. In general, mapping events occur whenever

- a new job arrives
- a node enters or exits the grid
- an application finishes or is aborted
- acceptable robustness/performance changes

However, for the purposes of this simulation, mapping events are only fired when new jobs arrive, and when applications finish. In general, a robust scheduler will need to take robustness into account when making mapping decisions. It is interesting to ask, however, how well scheduling algorithms which are not aware of robustness will perform. We have selected the following three scheduling algorithms to study:

- *FCFS*: First come, first serve. Maintains a queue of applications, in the order they arrive at the grid, and assigns them to the nodes in the order the nodes become available. When an application finishes running, if there is another application to run to complete that application's associated job, then that application will be added to the end of the incoming work queue. Thus, this is a FCFS algorithm on applications, not jobs.
- *RR*: Round-Robin. Runs all the applications from the first job on the first node, all the applications from the second job on the second node, and so on. It is also possible to round robin applications, rather than paths, to nodes, but path-based round robin was chosen for simplicity.
- *MCT*: Minimize Completion Time. Assigns each application in a job to the set of nodes that will spend the least time performing computation (i.e., the set of nodes with the shortest latency, but not necessarily the set of nodes that will produce the result first). This is achieved by keeping a priority queue for each node that holds its work to be done, ordered such that jobs that entered the grid earlier are given higher priority. When an application finishes running, the next application that must run to complete the job is placed at the appropriate spot in the appropriate queue. *Note that this is the only scheduler that uses ETC values when making scheduling decisions*. Also note that a priority queue is used because although a regular queue would also minimize completion time, it would do so without regard to latency. In other words, our heuristic is that to help minimize latency, older applications should be run before new applications, which is implemented by the priority queue. This heuristic does not hold in all cases, but it is a good rule of thumb that makes for a more realistic (and interesting) scheduler.

Each of these schedulers assumes perfect knowledge of the grid. This means that, when a node completes a task, the scheduler is notified; the scheduler itself maintains a queue of work for each

node. In practice, with a grid distributed over continents and with large communication latencies between nodes, implementing a centralized, omniscient scheduler might be impractical. However, it is useful to know how an idealized scheduler performs in simulation before attempting to implement any scheduler in practice.

III. Simulation Description

A. Overview

Our goals for the simulation were two-fold. First, we measured the performance features of each of the three schedulers, under a variety of conditions, to get a general idea of how the schedulers perform. Then, we introduced variations in the actual completion times for tasks and measured the robustness of the MCT scheduler to these perturbations in ETC values. (We only examined MCT under perturbation because RR and FCFS do not use ETC values. Although perturbing task completion times for these two schedulers would likely cause variations in robustness from simulation to simulation, the changes in robustness are uninteresting: because ETC values are not used when making scheduling decisions, a perturbed system is essentially equivalent to another system with different ETC matrices; this is not the case in an MCT system.)

B. General Framework

The simulation framework, written in Java, is relatively simple. The simulation runs a series of *ticks*, simulating the passage of time. A `Scheduler` is notified about mapping events and assigns applications to nodes. The `Applications` then spin repeatedly as ticks occur, notifying the framework when they are complete. Each `Job` is given an ETC matrix using the method described above.

C. Parameters

Because it is computationally expensive to generate ETC matrices for hundreds of jobs, the simulation framework creates a number of *template jobs*. Each template job has an associated *arrival rate*, which determines the average number of instances of that job that enter the simulation per tick.

In order to simulate realistic environments, we must decide on values of Vtask, Vmach, and Mutask for each template job so that we can generate ETC matrices by the method in Figure 1. (Of course, "realistic" is a relative term; our grids are already highly idealized. By realistic ETC values, we simply mean ETC values that might conceivably be encountered in practice.) To do this, we created the following initial values from which we deviate randomly as we create template jobs (how we do so will be described afterward):

- vmachinit = .4 (The central tendency of Vmach)
- vtaskinit = .4 (The central tendency of Vtask)
- mutaskinit = 20 (The central tendency of Mutask)
- vtaskvar = .2 (Variation for vtaskinit)
- mutaskvar = .1 * mutask (Variation for mutaskinit (2 ticks))
- numapps = 10 (The central tendency for number of apps per job)
- numappsvar = 7 (Variation for number of applications per job)
- ma = .01 (The central tendency for one job arrival per tick)
- mavar = .005 (Variation for ma (50%))

For each template job, we use the following algorithm to generate the three parameters for its ETC matrix (the Gaussian function samples a Gaussian curve over $(-1..1)$, and the uniform random function samples over the interval in the call's parameters):

```
vtask = nextGaussian()*vtaskvar+vtaskinit
mutask = nextGaussian()*mutaskvar+mutaskinit
numtasks = numapps +
 uniformRandom(-numappsvar, numappsvar)
```

The acceptable latency for each template job, and thus for each job instance of that template, is defined to be 1.5 * Mutask * numapps. (Note that as different template jobs have different Mutask and numapps values, not every job has the same acceptable latency.) This choice of acceptable latency is arbitrary, as in practice it would be specified by the customer; this value seems like a reasonable one. Also note that by defining the acceptable latency this way, we are essentially simulating a system with a single customer. Finally, note that we simulating a system that has large numbers of a few types of identical jobs. An example of such a system might be encountered on a grid that has to process thousands of database reads and writes.

In addition, we say that for each template job, its arrival rate is a Poisson function with mean

```
mean = nextGaussian() * mavar + ma
```

Note that each template job has its own arrival function; as the number of template jobs is increased, the system will become more loaded. For all the simulation runs, we use 20 template jobs.

D. Simulation Body

The body of the simulation is as follows:

1) Create the 20 template jobs and associated ETC matrices.
2) For each tick, do the next 3 steps:
   a) For each template job, query its arrival function to determine if an instance of it should arrive this tick.
   b) If a job needs to arrive, clone an instance of the job from the template and notify the scheduler.
   c) If applications have finished, notify the scheduler.
3) Finally, at the end of 10,000 ticks, count the number of jobs that have finished within their acceptable latency, those that have violated that latency, and those still in progress. 10,000 ticks were chosen to give the system enough time to enter a steady-state and to simulate long running grids.

Each simulation run is given a specified number of nodes; we vary the number of nodes while keeping the workload constant and measure performance. An average job has around seven applications that each take around 20 ticks; therefore a steady-state will certainly have been reached by 10,000 ticks.

E. How Realistic is this Environment?

Given our already highly idealized conception of grids (no node failures, no communication latency, etc), it is natural to ask how realistic this environment is. If we look at the sample ETC matrix in Figure 1, we see that the times are not implausible for

certain situations. Especially on grids where nodes are running other applications concurrently (i.e. the node is not "owned;" the grid is using "spare cycles"), large variations in ETC values would be expected. The issue of communication link latency is more difficult. Presumably, for certain grids and applications, we can simulate communication link latency by adding appropriate extra time to a job's ETC values. For other grids, allowing each application to pull data as soon as it is available from its predecessor application could hide communication latency. Developing an abstraction to allow use of ETC matrices with parallelizable jobs, and with varying communication latency and bandwidth, is an important direction for future research. However, we feel that this environment is still worth investigating, especially given the youth of the field.

IV. Results

A. Results for Adequate Performance

1) Preliminaries: Each run of the simulation consists of ten runs of the simulation body above. The statistics are totaled across the loop bodies, but the template applications are regenerated and the simulation reset during each execution of the loop body. The results are summarized in Tables 2-5. "OK," "Late," and "IP" refer to the number of jobs that finished within their acceptable latencies, finished outside their acceptable latencies, and did not finish as of 10,000 ticks, respectively. Because the performance feature is meant to capture the steady-state of the system, jobs in-progress at the end of 10,000 ticks were ignored while calculating performance. The performance is the percentage of jobs that finished within their acceptable latency, as defined earlier.

2) Data: See Figures 2-4 and Tables II-IV.

TABLE II
FCFS Results

Nodes	OK	Late	IP	Perf (%)
1	0	0	1820	0.000
2	0	0	1793	0.000
4	2	24	1762	0.077
8	15	93	1672	0.139
16	108	314	1415	0.256
20	271	394	1194	0.408
24	504	464	829	0.521
28	866	346	674	0.715
32	1397	6	362	0.996
64	1557	0	323	1.000

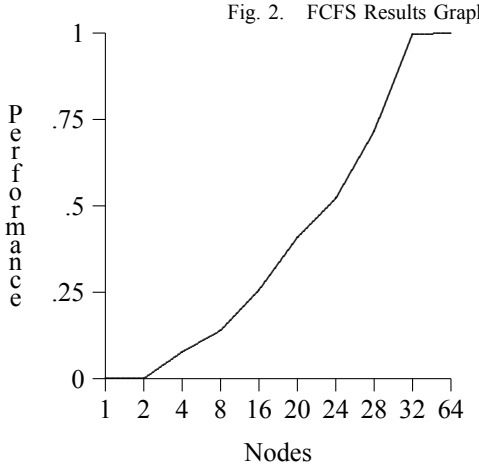
TABLE III
RR Results

Nodes	OK	Late	IP	Perf (%)
1	4	15	1810	0.211
2	25	73	1755	0.255
4	72	179	1560	0.287
8	134	271	1380	0.331
16	375	432	1046	0.465
20	576	413	874	0.582
24	837	350	682	0.705
28	1029	310	574	0.768
32	1256	116	442	0.915
64	1511	0	327	1.000

TABLE IV
MCT Results

Nodes	OK	Late	IP	Perf (%)
1	2	14	1739	0.125
2	26	91	1647	0.222
4	103	262	1416	0.282
8	401	405	1013	0.496
16	1471	62	351	0.960
20	1648	0	134	1.000
24	1697	0	120	1.000
28	1653	0	95	1.000
32	1756	0	107	1.000
64	1799	0	63	1.000

Fig. 2. FCFS Results Graph



3) Discussion: When interpreting the data, it is important to remember that achieving a steady-state has nothing to do with the performance features. For instance, for FCFS with small number of nodes, very few jobs are finishing; this is because the scheduler is assigning applications to nodes in a way that precludes jobs from finishing. This is, in fact, a steady-state, where there simply aren't enough nodes available to finish jobs quickly enough.

We see that both FCFS and RR have similar performance; both algorithms required a large number of nodes before maximum performance was achieved. MCT attained maximum performance extremely quickly, as we would expect for an algorithm that schedules jobs to minimize completion time.

It is interesting to note that FCFS consistently tended to perform slightly more poorly than RR. This can be explained by remembering that FCFS, when an application completes, will add the next application in the associated job to the end of the work queue, thereby requiring that application to wait for some time before starting. In effect, FCFS adds a long delay between applications in a job when the system is overburdened. Both RR and FCFS achieved a steady state maximum performance at around 30 nodes, however, because both FCFS and RR tend to execute applications on average nodes (that is, FCFS will assign essentially a random node to an application, and RR will assign a different random node). Because both assignments are random and average, we would expect maximum performance to be achieved in roughly the same spot.

Fig. 3. RR Results Graph

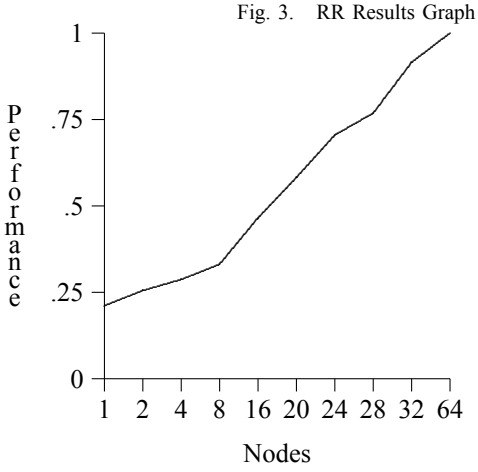
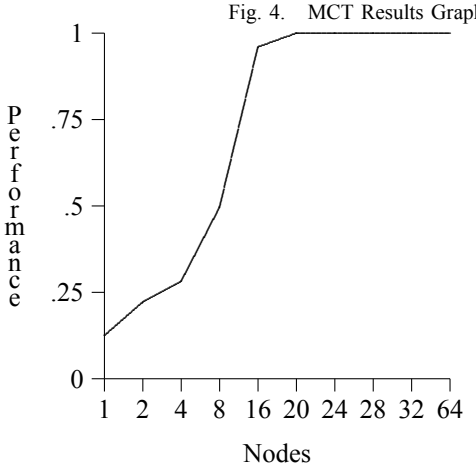


Fig. 4. MCT Results Graph



B. Results for Robustness

1) Preliminaries: In a sense, the above analysis simply confirmed what we might have expected: MCT performs much better than the other algorithms. We saw that MCT achieves perfect adequate performance for our environment when it has around 16 nodes. It is then natural to ask what MCT's robustness against variations/perturbations in ETC values is. In other words, how does MCT perform when the ETC values it is given no longer match the actual time it takes tasks to complete?

To introduce uncertainty into ETC values, we calculate ETC matrices as usual. Then, when a job begins running, we assign it a different time to completion based on a parameter `etcvar`:

```
actualetc = ((1-etcvar)*(etcvar)
            * nextGaussian(-oldetc, oldetc)
            + oldetc
            + (etcvar)*nextUniform(-oldetc,
            oldetc) + oldetc)/2
```

Thus, the scheduler assigns jobs to nodes based on *actual estimates* of completion times; before, when ETC values were the *real* completion times, the ETC values were perfect estimates. Now the ETC values truly do estimate uncertain completion times.

The intuition behind the formula is straightforward: in a system that is unperturbed (`etcvar=0`), the ETC values will be true estimates of job completion times; therefore, the actual distribution of job completion times for that job template should be a Gaussian centered at the ETC value. In a completely perturbed system (`etcvar=1`), the ETC value is the center of a uniform distribution whose average is the ETC value. For systems in between these extremes, a weighted average of the two values is used. We define this variation from a Gaussian distribution to be the *ETC perturbation*. Figures 5 and 6 demonstrate high and low perturbation.

Other definitions of perturbation are possible. For instance, rather than using uniform and Gaussian distributions, other distributions could be used; or, perturbation could correspond to the correlation of ETC values and the actual samples from the distribution. Our definition is offered as a simple, intuitive starting point.

Fig. 5. Low Perturbation

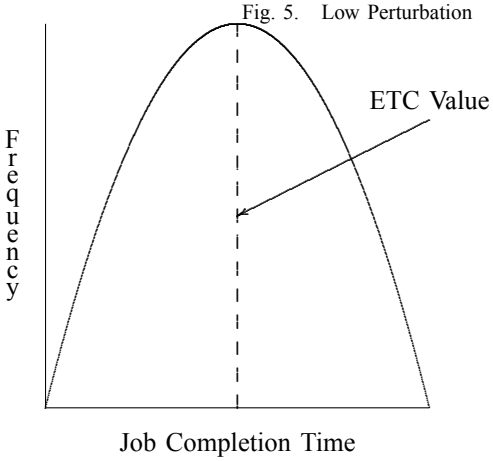
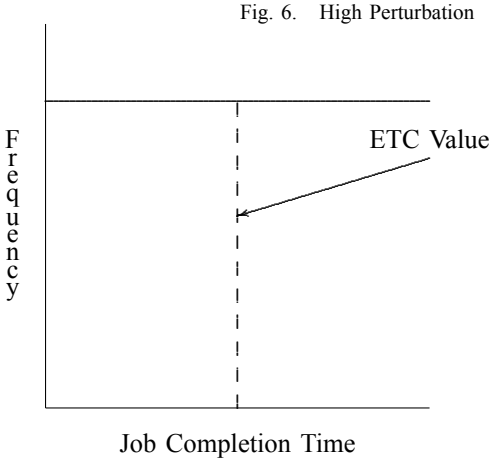


Fig. 6. High Perturbation



To determine how robust MCT is against ETC variation, we use `etcvar` as the perturbation parameter and examine adequate performance as above. We vary `etcvar` is increments of .1 (this increment is arbitrary; it is a compromise between too few and too many data points). Each simulation runs for 10,000 ticks, as before; however, these simulations are run three times and then their results averaged. For the sake of space, all the results are not included here.

2) Data: The data are available in Table V.

TABLE V
MCT VARIATION RESULTS

nodes	0	.2	.4	.6	.8	1	Max - Min
1	.13	.19	.12	.20	.20	.21	.15
2	.21	.24	.21	.21	.19	.20	.05
4	.21	.29	.29	.31	.30	.31	.10
8	.47	.44	.47	.46	.49	.43	.06
16	.94	.96	.96	.97	.97	.95	.04
20	1.0	1.0	1.0	1.0	1.0	1.0	0.0

3) Discussion: We see that MCT is remarkably stable against perturbation in ETC values. In fact, all degrees of perturbation seemed to make very little difference. However, performance values did tend to be slightly more diffuse than in the first experiment, hence the need to repeat the trials more than once. This stability suggests a more general result: as long as ETC values represent the average actual completion times of the associated jobs, then the actual underlying distribution of the job completion times does not matter to MCT. This result could potentially be generalized to any scheduler, indicating that ETC matrices might be useful as inputs to actual robust schedulers, rather than only being useful as tools for a simulation. This possibility would drastically increase the utility of using ETC values for scheduling decisions, as it demonstrates that a concept as simple as an ETC value could be useful in practice. A mathematical proof of this result for MCT is one direction for future work.

Of course, caution is urged. It is impossible to tell, from this experiment alone, whether this result is general or if it is the result of this particular system and its associated ETC matrices. For instance, we cannot know whether this stability will change if jobs are inserted at a slightly faster rate; we also cannot know whether this result holds for any scheduler besides MCT. The stability of MCT might be related to the MCT algorithm itself rather than ETC values being averages of actual completion times. Verifying this result for other environments and with other schedulers is yet another direction for future work.

In addition, even if ETC matrices do turn out to be useful in practice, effectively calculating the ETC matrices for a given grid with a given workload is still an open problem. Approximations created with the described method would be an excellent first step, but the problem is not trivial, and should prove to be an interesting area to investigate. Scheduling methods that do not require estimates of completion times are investigated in [2].

V. Conclusion

We have presented the results of a simulation study of a heterogeneous computational grid. After discussing a definition of robustness based on work completion latency and presenting a method using ETC matrices to model heterogeneous systems, we found that the MCT scheduling algorithm – an algorithm which attempts to minimize the total computational time required for any job – performed the best out of a set of well-known scheduling algorithms. To analyze MCT further, we presented the concept of ETC perturbation, and found that the MCT algorithm was quite stable against variations in job completion times.

However, the most interesting result of the simulations was not the data itself, but the intriguing possibility that ETC values might be useful as inputs to robust schedulers. That is, it could be the case that ETC matrices, even when they are only rough estimates of underlying job completion times, could be treated as true estimates without affecting robustness. Although more work is required to generalize our result, if ETC matrices could be used in this way, then we are one step closer to creating robust scheduling algorithms.

VI. Related Work

The concept of computational grids, and grid computing in general, is being studied by researchers in many fields, including high-performance computing, networking, distributed systems, and web services. [4] is an extensive introduction to what a computational grid actually consists of, and what is required to implement it. The Globus Consortium (`http://www.globus.org`) is a consortium of dozens of companies, government agencies, and universities that is creating an open standard for grid development using web-services as an RPC mechanism.

The modeling of computational grids with heterogeneous resources is just beginning to be explored. [5], published in 2002, can point to no directly related work in the field.

Scheduling tasks of unknown duration on distributed systems is investigated in [2].

The evaluation of scheduling algorithms focused on efficiency rather than robustness is explored in [1].

The construction of actual grids for industrial and scientific work has been undertaken by many companies and scientific groups. One particular success story is the Grid 2003 project (`http://www.ivdgl.org/grid2003`), which has developed a grid consisting of 2000 CPUs spread across the world.

The development of robust scheduling mechanisms is being investigated by IBM. Jay Smith (`bigfun@us.ibm.com`) is one contact point for this work.

Acknowledgements

I would like to thank Jay Smith of IBM for his help in creating the simulation framework and writing this paper. His investigations of ETC matrix methods and modeling techniques proved invaluable, and his definition of robustness was central to this research.

I would also like to thank Professor Mendel Rosenblum for graciously agreeing to supervise my work this quarter.

References

[1] C. Boeres et al. *A tool for the Design and Evaluation of Hybrid Scheduling Algorithms for Computational Grids*. Proceedings of the 2nd workshop on Middleware for grid computing, ACM International Conference, October 2004.

[2] Harchol-Balter, Mor. *Task Assignment with Unknown Duration*. Journal of the ACM, Volume 49 Issue 2, March 2002.

[3] Ian Foster. *What is the Grid? A Three Point Checklist* Grid Today, volume 1 number 6, July 22 2002.

[4] Ian Foster, Carl Kesselman. *Computational Grids*. Chapter 2 of *The Grid: Blueprint for a Future Computing Infrastructure*. Ian Foster and Carl Kesselman, Morgan Kaufman, 1998.

[5] Shoukat Ali, Howard Jay Siegel, Muthucumaru Maheswaran, Debra A. Hensgen, Sahra Ali. *Task Execution Time Modeling for Heterogeneous Computing Systems*. Heterogeneous Computing Workshop, 2000: 185-199.

http://ieee.stanford.edu/ecj