7-2016

Remote Access and Computerized User Control of Robotic Micromanipulators

Ryan M. Dunn
rmd8337@rit.edu

Remote Access and Computerized User Control of Robotic Micromanipulators

by

Ryan M. Dunn

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science
in Mechanical Engineering

Supervised by

Assistant Professor Dr. Michael Schrlau
Department of Mechanical Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
July 2016

Approved by:

Dr. Michael Schrlau, Assistant Professor
Thesis Advisor, Department of Mechanical Engineering

Dr. Kathleen Lamkin-Kennard, Associate Professor
Committee Member, Department of Mechanical Engineering

Dr. Mark Kempski, Professor
Committee Member, Department of Mechanical Engineering

Dr. Alan Nye, Professor
Department Representative, Mechanical Engineering

Acknowledgments

I would like to thank my labmates for assisting me with my research and making the NBIL an enjoyable place to learn and work. Masoud has been a valuable member of the lab for the entire duration of my work. Anna, Olivia, and Adeel are great colleagues and sources of both help and entertainment. I wish good luck to Christina, James and Devarsh on the continuation of their research. I hope to remain friends with all of you.

Thanks as well to Sam, my first and oldest friend at RIT. Thank you Mike and Nikki, my parents and role models, for raising me with good values and giving me a great education. And thank you to Ruth Schirmer for giving me a Rochester family.

Most importantly, thank you Dr. Schrlau for introducing me to the NBIL, encouraging me to pursue research and for supporting me all throughout it. Dr. Schrlau provided me with assistance when I needed it, encouragement when I needed it, and independence when I needed it. I could not have hoped for a better mentor and adviser and it will not be forgotten.

Abstract

Remote Access and Computerized User Control of Robotic Micromanipulators

Ryan M. Dunn

Supervising Professor: Dr. Michael Schrlau

Nano- and micromanipulators are critical research tools in numerous fields including micro-manufacturing and disease study. Despite their importance, nano- and micromanipulation systems remain inaccessible to many groups due to price and lack of portability. An intuitive and remotely accessible manipulation system helps mitigate this access problem. Previously, optimal control hardware for single-probe manipulation and the effect of latency on user performance were not well understood. Remote access demands full computerization; graphical user interfaces with networking capabilities were developed to fulfill this requirement and allow the use of numerous hardware controllers. Virtual environments were created to simulate the use of a manipulator with full parametric control and measurement capabilities. Users completed simulated tasks with each device and were surveyed about their perceptions. User performance with a commercial manipulator controller was exceeded by performance with both a computer mouse and pen tablet. Latency was imposed within the virtual environment to study it's effects and establish guidelines as to which latency ranges are acceptable for long-range remote manipulation. User performance began to degrade noticeably at 100 ms and severely at 400 ms and performance with the mouse degraded the least as latency increased. A computer vision system for analyzing carbon nanotube arrays was developed so the computation time could be compared to acceptable system latency. The system characterizes the arrays to a high degree of accuracy and most of the measurement types of obtainable fast enough for real-time analysis.

Contents

List of Tables

List of Figures

Chapter 1

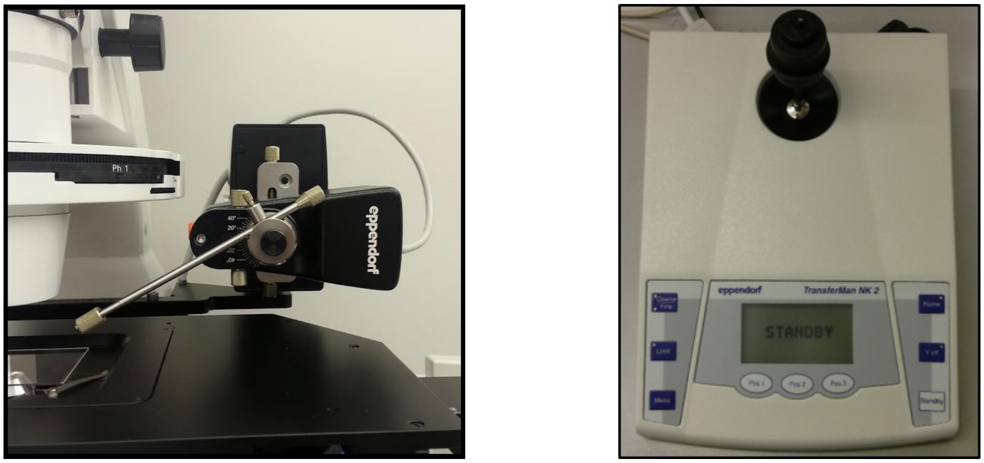
Introduction

Micro- and nanomanipulators are high-precision positioning instruments used in conjunction with high magnification optical and electron microscopes to interact with objects on the nanometer scale. Biomedical science utilizes such equipment to elucidate the behavior of individual living cells at the nanoscale. Many commercial nanomanipulator control systems and devices are unintuitive and expensive, preventing educators and researchers from utilizing them as tools to teach or study nanoscience.

1.1 Motivation

Manipulators are critical tools for a wide variety of applications ranging from cell injection to nano-assembly. Commercial manipulators move at least one end effector typically with three or more degrees of freedom. For example, the Eppendorf Transferman NK2 manipulator shown in Figure 1.1a can move a single end effector in the X, Y, and Z Cartesian co-ordinates and are used in conjunction with optical microscopes. Using two, in a dual manipulator arrangement, allows for cell injection by holding a cell with a suction probe and injecting it with the other. Other micromanipulation systems can support multiple end-effectors each with up to 6 degrees of freedom.

The manipulators themselves have robust and proprietary control systems built in to ensure that their actuators drive the end effectors to the correct position based on user input. These control systems are fast and accurate but external factors like mounting and



(a) Transferman NK2 motor-module.

(b) Transferman NK2 Control Unit

Figure 1.1: Eppendorf Tranksfermank NK2 manipulator (a) and Control Unit (b).

installation can introduce error that the manipulator cannot account for. Many systems are standalone and do not require the use of a computer workstation. The Eppendorf Transferman NK2 can operate as a standalone system and has a proprietary joystick controller shown in Figure 1.1b.

This control system, and others, can receive a command signal from an external source allowing the manipulator to be controlled from a personal computer. The Transferman NK2 utilizes serial communication but other manipulators use a variety of communication protocols and connectors. Control from a personal computer allows for a high degree of customization and automation using software written in any language supported by the PC, including MATLAB. For example, cells can be identified using image processing and probes can be moved to the cell by calculating the difference in their positions. Computerization also allows for additional hardware devices to control a manipulator provided that the hardware can be interfaced with the PC. Because manipulators are critical research tools, manipulator automation with custom software is an active field of research and represents significant potential for manipulator functionality to be expanded.

A field similar to nano- and micromanipulation involves using minimally invasive surgical robots to perform procedures that would otherwise require large incisions. While the importance of nano- and micromanipulators as a research tool should not be understated, they do not have the same immediate lifesaving potential that surgical equipment does. As a result, surgical technology is more advanced in certain areas. Two such areas are user control and remote use over long distances. Due to the complex nature of non-invasive surgical procedures the robotic devices themselves are more articulated and more difficult to control. Additionally, expert surgeons represent valuable human capital and long distance access allows them to perform operations from other countries. Not nearly as much research has been carried out in these areas in regards to nano- and micromanipulators. Robotic telesurgery technology serves as an important reference for this research.

Aside from research applications, manipulators have potential as an education tool. Exposure to STEM fields in high school is important in encouraging students to consider the pursuit of higher education in engineering. In 2012, the Nano-Bio Interface Lab (NBIL) demonstrated rudimentary control of a manipulator in a classroom. Students were able to experience technology hands-on that is usually inaccessible. To date the NBIL has hosted two demonstrations involving manipulators in the Rochester City School District (RCSD). Computerized control and remote user interfaces would facilitate more of these demonstrations as part of the NBIL's outreach through RIT.

Unfortunately, nano- and micromanipulator technology is prohibitively expensive and while attempts are being made to reduce the costs of these devices they still remain inaccessible to high-school classrooms and represent a significant investment for many labs. A method of increasing accessibility is to develop software that allows remote access to these devices. Such software would allow the technology to be used in high school classrooms as a teaching tool and let students manipulate cells and nanoscale objects themselves in real-time. Additionally, a remote manipulator system requires an inexpensive and portable control device for use at the remote workstation. For laboratory use, efficiency in the long term is the primary selection criteria but for hands-on classroom demonstrations an engaging control method is desirable.

In conclusion, the motivation of this research is twofold. First, manipulators are critical research tools but the technology supporting them is not as developed as it could be, as is evident from relatively more advanced similar technologies. Furthermore, manipulators could serve as excellent educational tools both for teaching in fields related to biology or nanotechnology and for demonstrations in primary education. Enabling remote access to manipulators and improving the user interface through hardware and software developments represent significant opportunities to advance nano- and micromanipulator technology.

1.2 Research Goals

A manipulator control system including alternative control hardware and a remote access graphical user interface (GUI) allows users to access a manipulator without physically relocating it. Such a system enables remote use from other laboratories and facilitate hands-on technology demonstrations for education. Figure 1.2 shows the structure of a remote manipulation system with each of the focus areas of this research labeled. Each numerical label corresponds to experimentation towards developing and understanding long-distance manipulation. The first area relates to manipulator control devices and the human-computer interface. The second is latency introduced by the Internet connection and how it effects user performance. The third area involves the developing and testing of GUIs. The final area relates to automatic processing of the data read by the GUI. Specifically, a real-time carbon nanotube (CNT) array characterization algorithm was developed to serve as additional GUI input while operating within the delay ranges identified in the second focus area.

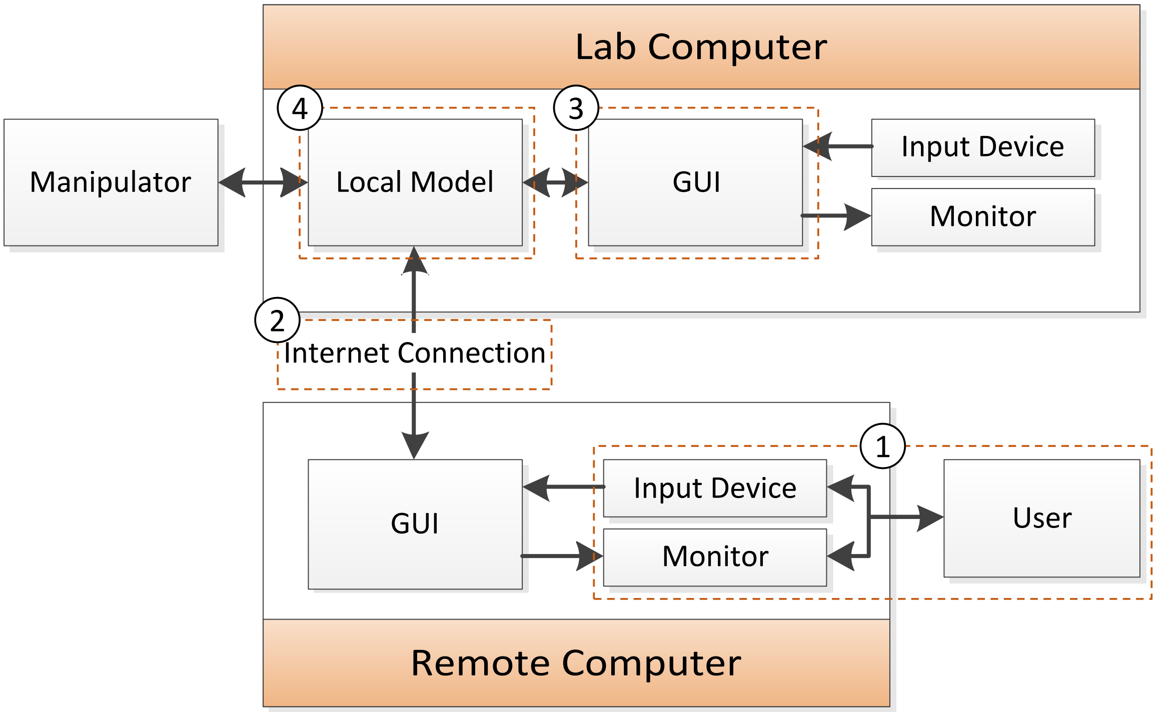


Figure 1.2: Schematic showing information flow between components of a remote nano- or micromanipulation system. Focus areas of this research are labeled.

1. Manipulator Control Devices

Accessing microscope cameras from a personal computer has benefits for both viewing and recording and the ability to drive a manipulator from a computer provides convenience and flexibility depending on the software used to control it. Controlling a manipulator through a MATLAB GUI allows for varying levels of automation and flexibility in control hardware by using different devices as inputs to MATLAB. A complete manipulator GUI has numerous benefits for both frequent and infrequent manipulator users.

It was unclear, and not well researched, if commercial control devices for nano- and micromanipulator control are more effective than alternative devices. A test procedure was established to gather statistically significant information based on metrics for user performance evaluation. These metrics included quantitative measurements like speed, accuracy, precision as well as qualitative metrics like ease-of-use and level of engagement. Qualitative metrics are not trivial to define required investigation to establish them and develop a procedure to measure them. Multiple candidate hardware devices were interfaced with MATLAB and implemented with GUIs. A comprehensive evaluation method utilizing a virtual environment was developed and multi-user tests were carried out to systematically evaluate user performance across each device in the context of both laboratory use and education.

2. Understanding Latency

When utilizing a fully computerized manipulation system with digital optics, and therefore video, the primary performance degradation introduced by long distance access is network latency. The effects of latency have been studied for other applications including robotic telesurgery but these effects are not well understood for micro- and nanomanipulation. This research forms the basis of understanding the effects of latency for micro-manipulation and how it differs from other fields. A multi-device virtual environment test was performed to quantify performance degradation over a range of imposed latencies to develop guidelines for which latencies are acceptable with different hardware controllers. Finally, latency and networking performance is largely dependent on the remote access implementation. Various implementations, enumerated in Chapter 4 were developed or implemented, and then compared.

3. User Interfaces

Significant prior work in the NBIL has been carried out towards the development of manipulator GUIs. The first iteration, completed by Nicholas Hensel and continued by myself is robust, but computationally expensive and difficult to add complex features to. It was possible that this research would benefit from a new, simpler GUI or a lighter version of the existing GUI [1]. A minimal, expandable GUI was developed to be used for testing with the intent of using it for educational purposes.

4. Computer Vision

Hybrid automation is an excellent tool to enhance user performance through computerization. By processing the visual input to a GUI, information about the viewing area can be calculated and used to provide real time information to the user. This information can also be used to drive enhanced control features. For example, image processing can be used to identify cells allowing injection to be carried out automatically.

The NBIL also performs research relating on the development of CNT arrays. When fabricating CNT arrays numerical geometric analysis must be carried out to monitor the manufacturing process. Imagining the arrays and taking a battery of measurements manually is tedious and subject to human error. Performing these measurements automatically at the point of capture improves rapidity and effectiveness of the development process. These computer vision driven calculations can potentially be too computationally expensive to run in real time. Once the impact of performance from latency was well understood, the computations with acceptable processing time can be identified.

1.3 Literature Review

1.3.1 Manipulators and Manipulator Control

Micro- and nanomanipulators provide the ability to maneuver and position micro- and nanoscale objects to precise locations and orientations. Used in conjunction with optical and electron microscopes, these ultrahigh-precision positioning instruments enable functions critical for many micro- and nanoscale applications, such as manufacturing and biomedicine. In manufacturing, manipulator systems have been employed to automate the maneuvering and positioning of small objects in order to produce patterns, structures, and devices. For example, Capelleri *et. al.* developed caging grasps for micro assembly capable of accurately positioning micro objects using an array of probes [2, 3].

In biomedicine, manipulators are commonly utilized to maneuver and position small probes inside living cells and tissue. Cell injection is a large area of research; Wang *et. al.* developed a method of carrying out cellular injection using probes at a high rate and with minimal user intervention [4]. Grippers are used in addition to pipettes for biological applications. For example, Kim *et. al.* developed a nanonewton microgripper to analyze the properties of biomaterials [5].

The goal of micro- and nanomanipulator automation is to create a system requiring only minimalistic user intervention, optimally none, to carry out a specific desired function. In this way, automation of predefined tasks increases precision and throughput while reducing variability and time. Most automation requires computer vision algorithms to access the position of manipulation targets. Wang *et. al.* also developed high-throughput automatic injection systems and demonstrated their use on zebrafish embryos [6–8] and contributed image processing techniques for injection automation [9]. Mattos *et. al.* developed image processing techniques to improve their automated injection process of blastocyst cells [10–14].

However, situations exist, especially in the development and utilization of new tools and techniques, where the automation and control of these manipulators, and other related equipment, needs to be flexible and adaptable. For example, in our own work towards developing carbon nanotube (CNT)-based probes for single cell analysis [15–21], micro- and nanomanipulators are routinely used to maneuver the functional end of the probes in order to interface with single living cells in an undetermined manner, often requiring on-the-fly repositioning or customized movements based on qualitative visual feedback. Here, the tips of CNT-based probes are manually maneuvered in Cartesian space by the manipulators joystick and positioned within the intracellular environments of single living cells under an optical or fluorescence microscope to perform functions or analysis with tertiary instruments. Hensel *et. al.* integrated image processing with a custom, extensible and user-friendly interface to create a robust manipulator GUI [1]. The GUI included automation options and various control methods including cursor following and point-and-click movement.

1.3.2 Latency and Long-Range Control

Latency

Nano- and micromanipulation is a tool critical to numerous scientific fields and has the potential to serve as an educational resource. Many fields including nano-manufacturing and biomedicine are reliant on these manipulators to develop assembly processes and elucidate the behavior of cells on an individual level [22, 23]. Nano- and micromanipulation technology is cost prohibitive for many labs and classrooms, limiting their use. One solution to overcome limited access is to enable manipulators to be accessed remotely. Bolopion *et. al.* developed an environment for SEM control that reproduces the SEM image remotely and allows a user to control a probe using a haptic interface [24]. Bolopion *et. al.* emphasized the negative impact that delays can have on haptic coupling. To reduce latency they transmitted numerical data about the sample and reconstructed an image of it rather than transmitting real-time video. The reported round trip latency of the system over a distance of about 400 miles from Oldenburg, Germany to Paris, France was only 37 ms although the computation time of the image processing and reconstruction added to the total system latency [25]. While some groups have developed long-distance remote manipulation systems, the performance impact from operating remotely is not well understood [26, 27]. There are numerous other examples of remote systems expanding the reach of doctors including ″Eye, Robot,″ a system for remote ophthalmologic (eye) examinations [28]. A joystick was selected as the user control device and latency is less of an issue because the doctor is only viewing the patient. The success of remote applications in medicine and the similarities between them and remote nanomanipulator suggest that a remote nanomanipulator system will be viable.

The most significant performance issue posed by remote micromanipulation is latency. Even small delays from data transmission and computation time can cause performance degradation and user discomfort [29]. Understanding the acceptable range of latencies for remote micromanipulation is critical to the implementation of a successful remote manipulation platform and expanding their accessibility for research and education. Automated systems relying on visual feedback are also susceptible to latency but the effects can be mitigated [30]. Unfortunately these techniques are not directly applicable to human-controlled systems where the software cannot judge the intent of the user beyond the input they provide. Hybrid automation techniques for micromanipulation such as allowing users to specify a series of manipulations to be carried out all at once can help mitigate problems arising from latency by reducing the reliance on live visual feedback [1, 13]. Bohren *et. al.* describes another hybrid automation technique applicable to manipulation that relies on the system recognizing the intent of the user but is only possible for tasks that are well defined and structured in advance [31].

Robotic Surgery

A field similar to nano- and micromanipulation is robotic surgery: the use of minimally invasive surgical robots to perform procedures that would otherwise require large incisions. Surgical technology is more developed than nano- and micromanipulation technology in a few areas including user control hardware and long range remote access [32–35]. Due to the complex nature of non-invasive surgical procedures the robotic devices themselves are more articulated and more difficult to control [36]. Additionally, expert surgeons represent valuable human capital and long distance access allows them to perform operations from other countries [37]. Not nearly as much research has been carried out in these areas in regards to nano- and micromanipulators. Robotic telesurgery technology serves as an important reference for this research.

Like micromanipulation, the most serious problem introduced by remote telesurgery is latency. Significant latency can cripple a surgeons ability to perform precise procedures and inconsistent latency is almost impossible to adjust to. Latency is an inescapable property of communication networks and it can be minimized through robust networks and programming techniques but will always exist. Xu *et. al.* tested medical students with a dV-Trainer telesurgery simulator with latencies ranging from 0ms to 1000 ms [38]. The dV-Trainer is a simulator designed specifically for training with the *da Vinci Si* Surgical System. As anticipated there was a positive correlation between completion time and errors with increased latency. Latencies below 300ms were deemed ”safe” by all participants while less than 20 % deemed latencies above 800 ms safe. Xu *et. al.* concluded that while 300 ms delays are suitable, latencies below 200 ms are ideal for telesurgery. Xu *et. al.* also looked to decrease the negative effects of latency by training users in a high latency environment but found that the acclimation to the latency only lasted a week and warned against excessive confidence from latency training [39]. Other fields are also concerned with human perception of delay. Nakamura *et. al.* found that humans could adjust to delays in prosthetic myoeletric hands up to a delay of 170 ms without feeling uncomfortable with the performance of the prosthetic [40].

Remote systems, including robotic surgery, present complications relating to instrument control. Surgeons have traditional training and experience with hand tools that only translate to robotic controls if the robotic controls mimic the hand tools. Furthermore the surgeries rely on optical tools for visual feedback rather than the naked eye. Control of articulated, multiple DOF actuators and intraoperative imaging has been the focus of remote telesurgery over the past decade [33].

While these drawbacks are severe, they can be overcome. Many successful cases of remote surgery have been reported in the past three years involving manipulators and procedures of varying complexities. Wirz *et. al.* reports a successful endonasal surgery with a latency below 100 ms during which the surgeon reported no discernible difference between local and remote robotic surgery [35].

Control Device Evaluation

Pen tablets are used frequently among professional and amateur digital artists because the similarity to a traditional artistic medium suits their work better than a mouse and keyboard. It is possible that these similarities may also extend to manipulator control or create heightened engagements in educational demonstrations. While investigating the ways users develop an emotional investment in tablet PCs, Zamani *et. al.* found that touch interfaces facilitate a heightened connection to the technology they are controlling [41]. The use of pen tablets for educational purposes has also been investigated. Romney reported an increase in freshman math retention rates with the introduction of Tablet PCs for note taking [42]. Oviatt *et. al.* found that, when solving math problems, low-performing students preferred a digital tablet interface while high-performing students preferred traditional pen-and-paper [43]. Problem solving time also increased with the digital interface likely due to the increased complexity of the system over pen-and-paper. When using a pen tablet to control a manipulator there is no added complexity for the user. The manipulators movement is mapped directly and intuitively with the users movement similar to any hand tool. Due to their adoption in other fields, pen tablets were strong candidates for alternate manipulator control, particularly for students using the device for the first time.

Typically, humans have better control over devices in their bare hands than over devices controlled by hardware devices like a joystick due to years of developing fine motor skills through day-to-day activities. Bare hand tool control is not possible for micromanipulators but it is for surgery. There is much more research centered on achieving fine robotic control for surgery but not for nano- and micromanipulation. However, due to the similarities between the two fields, much of the research is still applicable to micromanipulation. In systems such as these, the target location is identified by the user and their ability to navigate their device to that location needs to be quantified to evaluate the effectiveness of the system. The users performance should be measured by speed and accuracy independently. Human performance with control devices can be evaluated in virtual environments in a number of different ways. When testing motor skills with a haptic joystick, O'Malley *et. al.* tasked users to hit a target with a virtual mass tethered by a spring to another mass they could accelerate using their joystick [45]. Komlodi *et. al.* combined a qualitative and quantitative approach to analyze the viability of a Wii remote controller in 3D virtual environments [46]. After being asked to complete tasks in a virtual environment the users were asked questions with varying specificity. Broad questions like ″how was it″ and more specific questions like ″how would you describe this experience with one word″ help to develop an understanding of the user's qualitative experience with the control device. The virtual environment outlined in [46] is much more complicated than one required to test manipulator control devices but the qualitative analysis is a valuable reference.

At the Italian Institute of Advanced Robotics' Biomedical lab (*incl.* Mattos, Becattini *et. al.*), path tracing tests were used to evaluate hardware control devices for a surgical laser in a virtual environment [47]. Their evaluation method involved following a circular path and measuring the deviation from that path. Circles are common laser ablation paths for surgery. Three non-traditional devices were used: a Microsoft Xbox 360 controller, a Dell Latitude XT2 Tablet PC and a Saitek Cyborg Evo Force velocity joystick. A UniMax Micromanipulator was also used as it is an established device for laser surgery. The Microsoft controller and Saitek joystick both operated as velocity joysticks while the Tablet PC used the absolute position of the electronic stylus. Tasks were completed with visual feedback directly from the microscope and again using a computer display. With every device, error was reduced when using the computer display over the microscope's optical path and nine out of ten users preferred the computer display.

Chapter 2

Control Hardware

Additional control hardware is required for remote workstations. It is unclear if the commercial micromanipulator controllers are as effective as widely available alternatives. A major benefit of controlling a manipulator through a GUI is extensibility to additional hardware controllers. Price and portability were the primary concerns in the selection of the control device at the remote workstation. For these reasons a velocity joystick, a keyboard and mouse, and a pen tablet were selected for consideration. Each of these devices were interfaced with the MATLAB GUI as a control option at both the local and remote workstation. A device evaluation program, also programmed in MATLAB generates a field of targets and obstacles that the user must navigate with each control device. The completion times for each device were recorded to evaluate the efficiency of each control device. Additionally, for educational applications, users were surveyed to determine which devices were the most engaging. Results indicated that mouse control is more effective than joystick control for single probe manipulation. Tablet control is also suitable while velocity joystick control is not.

2.1 Introduction

Nano- and micromanipulator technology is prohibitively expensive and while attempts are being made to reduce the costs of these devices they still remain inaccessible to high-school classrooms and represent a significant investment for many labs. An alternative method of increasing accessibility is to develop software that allows remote access to these devices. Such software would allow the technology to be used in high school classrooms as a teaching tool and let students manipulate cells and nanoscale objects themselves in real-time. Additionally, a remote manipulator system requires an inexpensive and portable control device for use at the remote workstation. There are numerous possible control devices that fit this need, including a joystick, a mouse and keyboard, or a pen tablet touch interface. Each of these control devices are explored for potential use with a remote system. For laboratory use, efficiency in the long term is the primary selection criteria but for hands-on classroom demonstrations an engaging control method is desirable. This chapter describes the methods that were used to identify the most effective control device for laboratory as well as classroom use.

2.2 Experimental Design

Because this experiment involved human subjects, it required Institutional Review Board Approval. Approval was obtained on October 8th 2015.[1]

2.2.1 Control Device Selection

Control devices were selected based on cost and widespread familiarity and are listed in Table 2.1. Joysticks are common manipulator control devices and were natural considerations for the remote workstation, however joysticks vary dramatically in both price and design. Joysticks typically operate using positional or velocity control. A velocity control joystick returns itself to the center position and its deviation from center determines the velocity of the controlled actuators. A position joystick does not return to center and the actuators are driven to match the relative position of the joystick. The proprietary joystick of the Transferman NK2 is a positional joystick seen in Figure 1.1b. The velocity joystick

[1] Exemption 2. 46.101(b)(2) - <https://www.rit.edu/research/hsro/exemption_categories>

selected for consideration is a Logitech 3D Pro, seen in Figure 2.1.

Table 2.1: Summary of selected hardware control devices.

Control Device	Model	Description
Mouse	Dell MS111	Traditional PC mouse, widely available.
Pen Tablet	Wacom Intuos CTH640	Touch-sensitive tablet utilizing an electronic pen for pressure sensitive cursor control.
Velocity Joystick	Logitech 3D Pro	3-DOF (X, Y, Rotation) gaming joystick. Joystick's position corresponds to cursor velocity.
Position Joystick	Transferman NK2	3-DOF joystick with hardware controller for use with Eppendorf micromanipulators. Joystick's position corresponds to cursor position.

It is important to ensure that the velocity joystick control is tuned to provide the best user experience possible as to not introduce avoidable control deficiencies to the system. With the joysticks X and Y positions each ranging from -1 to 1, a dead zone was selected between -0.05 and 0.05 to ensure it was not too sensitive and the cursors position does not change while the user has the joystick centered. A scaling coefficient of 10 is applied to the velocity meaning that the maximum possible speed of the cursor is 10 px/frame while the test program runs at 10 frame/s. These values were selected through qualitative testing. If the user feels that the velocity joystick drives the cursor too rapidly, they may reduce the velocity scaling coefficient using a flap on the joysticks base.

Pen tablets are used frequently among professional and amateur digital artists because the similarity to a traditional artistic medium suits their work better than a mouse and keyboard. We desire for students to have a positive experience during technology demonstrations and it is possible that a tablet will help meet these goals thanks to an increased tactile connection to the manipulator [41–43]. Determining if a pen tablet interface would create a substantially different user experience, particularly for students, than more traditional control systems was worth investigation.

A tablet was selected based on price and portability so it is feasible to transport it for remote use. A Wacom Intuos CTH680 was selected, pictured in Figure 2.2. The pens position on the tablet controls the X and Y position of the manipulator while the pressure of the pen on the tablet controls the Z position between two programmable levels (low

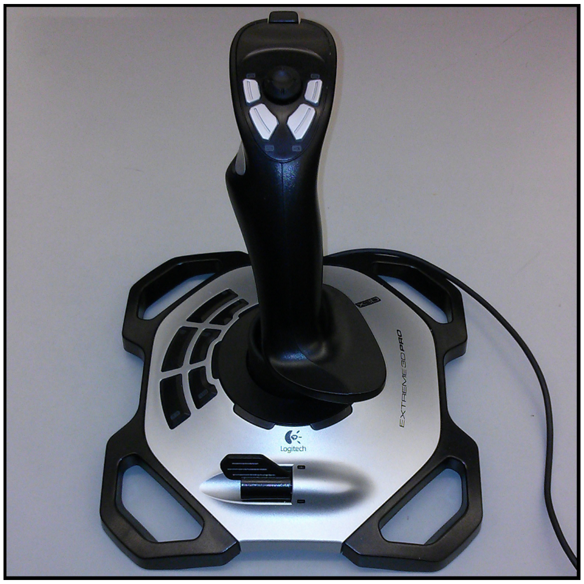


Figure 2.1: Logitech 3D Pro velocity control joystick.

Figure 2.2: Wacom model CTH 680 pen tablet.

pressure is fully raised while high pressure is fully lowered.) An option for a higher degree of discretization is included however most single probe manipulator applications involve two z-levels, one for interacting with the same and one for repositioning the manipulator freely.

A traditional mouse is also considered as one of the potential control methods for demonstrations because almost all users will have had experience with them. The mouse is implemented such that its position controls the X and Y position of the manipulator and the scroll wheel controls the Z position. If no scroll wheel is available the Z position may be controlled using a keyboards arrow keys.

2.2.2 Virtual Task Design

In order to determine the effectiveness of manipulator control, certain metrics must be established as a means of evaluation. Speed is naturally desirable and can usually be gained by sacrificing accuracy and precision. We seek a control system that maximizes a combination of these factors. Speed is defined simply as the rate at which operations can be completed. Hardware design and user proficiency are the major limiting factors of speed. Each manipulator has a maximum effective speed and when that speed is faster than comfortable for a particular user they will manually drive the manipulator at a lower speed, hence the need for variable speed control with any controller. Speed can also be limited in the software if the application calls for a lower maximum speed than the hardware's maximum possible speed, for example if the hardware's maximum speed could be damaging to another component or manipulation target.

Various methods of measuring movement accuracy and precision were considered. The first involved specifying a path for the user to follow with the control device. When the user completed the movement, a MATLAB algorithm calculated the total area enclosed by the users path and the specified path. A smaller enclosed area represented a more accurate and controlled motion. Any deviation from the path exceeding a specified distance was considered a failure. Similar task based assessment methods, outlined in Chapter 1, have been implemented that rely on line following tasks. When operating a manipulator during typical operations the user generally only needs to move the end-effector from point to point. The actual path is irrelevant as long as certain obstacles are maneuvered around and following straight lines is unimportant. To simulate these requirements, the test program generates targets and obstacles that the user must traverse to and avoid respectively.

The MATLAB test program was designed to be simple and allow the user to rotate though the devices and form a confident preference. The overall test program flow is

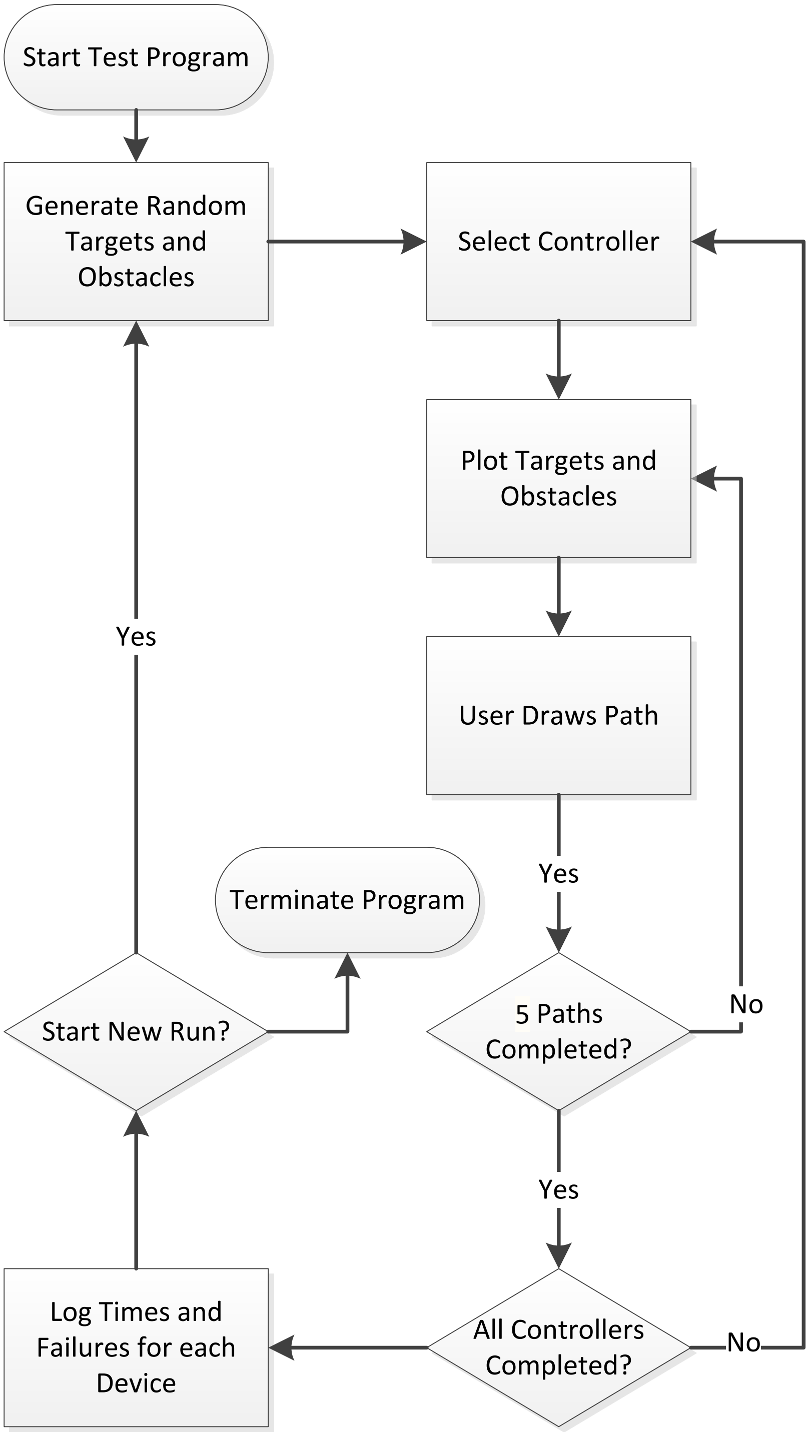
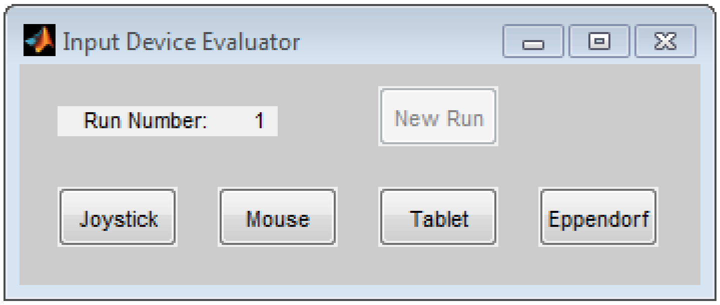

Start Test Program
Generate Random Targets and Obstacles
Select Controller
Plot Targets and Obstacles
User Draws Path
Yes
Yes
Terminate Program
Start New Run?
5 Paths Completed?
No
Yes
Log Times and Failures for each Device
All Controllers Completed?
No

Figure 2.3: Software flow of the hardware performance testing procedure.

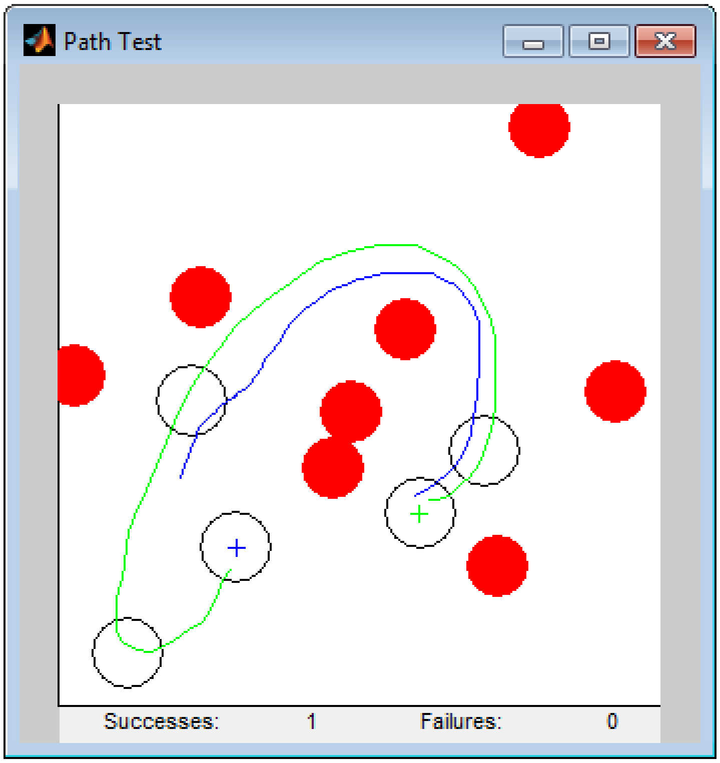
shown in Figure 2.3. When beginning the test process, the user is displayed the launch screen shown in Figure 2.4 accompanied by instructions and may select each of the controllers in any order. 5 random targets and 10 random obstacles are generated when the program is launched and each time a new run is started. When a user selects a device, the random target field seen in Figure 2.5 is displayed with a start target (a) and a destination target (b) and must traverse from the start to the destination target while touching every other target in any order. If the users path reaches each target and crosses no obstacles, the path is considered successful and the time duration of the path is recorded. When the joystick is the active device, the cursor location (c) is plotted by the program because the Windows mouse cursor is not in use. When three paths have been completed the user is prompted to select a new device. When all the devices have been selected and completed, the data for the cycle is logged and a new run is started with newly generated targets and obstacles. The users may select the controllers in any order and are encouraged in the instructions to vary the order. Finding a users preferred device is desirable and requiring the users to make a conscious choice in their device selection may help elucidate this preference.



Run Number: 1

Figure 2.4: Interface for launching virtual environment hardware tests.

To provide the user with feedback during the test, the program reports the number of path successes and failures. If a run is a failure because the path crossed an obstacle, the collision point is highlighted so the user does not have to search for the error and may start a new attempt right away.



Successes: 1 Failures: 0

Figure 2.5: Sample obstacle field for hardware controller tests showing 5 targets and 8 obstacles including a successful path (green) and a path in progress (blue).

2.2.3 Numerical Analysis

10 users completed 10 obstacle fields with all 4 devices. The users path is fully recorded however only the overall completion time for successful paths are utilized because they user may complete the path differently if they find a more efficient path meaning that the time between each target is inconsistent and non-uniform.

The experimental design allows for three major hardware control performance metrics. The first and most important is the relative performance with each device. This metric is the average task completion time with a device over all trials and can be measured on a per user basis or across all users representing the user's most effective device and the most effective device overall respectively. A relatively low task completion time represents good performance. Figure 2.6 shows the average task performance for a single user. While the difficulty of a particular obstacle field greatly influences the completion time, each obstacle field was completed with all 4 devices.

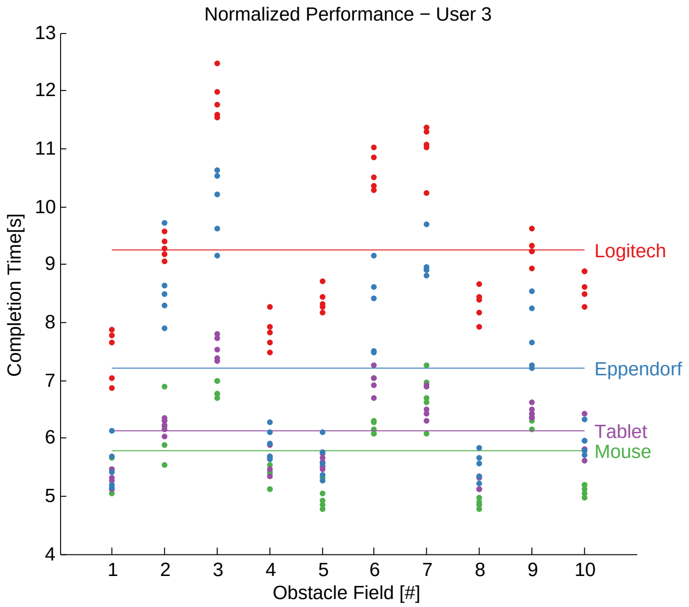


Figure 2.6: Single user hardware test performance data with average performance for each device.

During the course of the experiment users typically operate each hardware device for under 15 minutes. During this time a user may develop familiarity with the device. All users were already familiar with a mouse and a pen tablet is analogous to writing. Joysticks are less common, making it more likely that a user is not already proficient with one. Because the obstacle fields are not uniformly difficult it is insufficient to compare absolute completion time against obstacle field number for a single device. The completion times must be normalized by obstacle field difficulty. A sufficient measurement of obstacle field difficulty is the average completion across all four devices because it accounts for any issues that particular user had with completing the obstacle field. Normalizing a user's absolute times for each device by dividing them by their average completion time for that obstacle field has the added benefit of allowing data to be compared between users. Figure 2.7 demonstrates this normalization. The largest changes in actual task completion time are due to obstacle field difficulty masking trends in user performance. Once the data is normalized by difficulty, trends in relative performance over time can be observed. Figure 2.8 shows normalized task completion for a single user with linear fitted slopes of task completion time vs. obstacle field number. A negative slope indicates that the user adjusted to the hardware device faster than with other devices while a positive slope indicates slower acclimation relative to the other devices. Because the data is normalized based on performance, if a user did not gain any proficiency with the mouse but gained proficiency with the other devices the normalized mouse data will increase with obstacle field number.

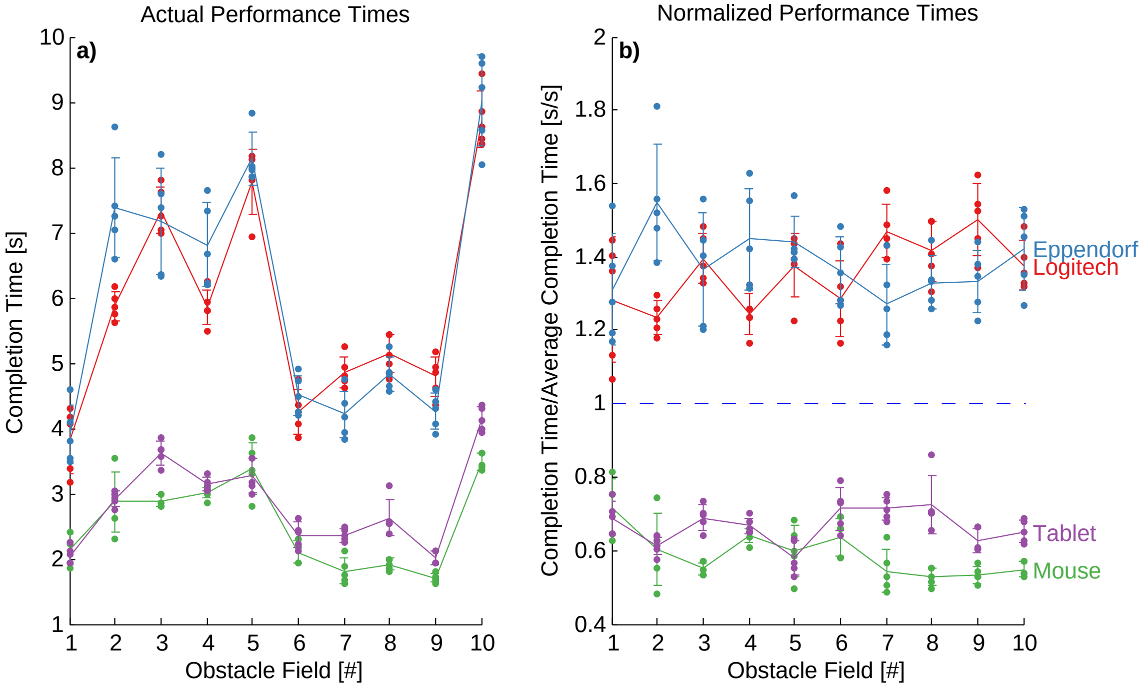


Figure 2.7: Single user task completion data not controlled by relative task difficulty (left) and controlled by relative task difficulty (right). Error bars represent one standard deviation.

Further tests including Analysis of Variance (ANOVA), measurements of standard deviation, and analysis of trends within trials were performed and discussed in the results section.

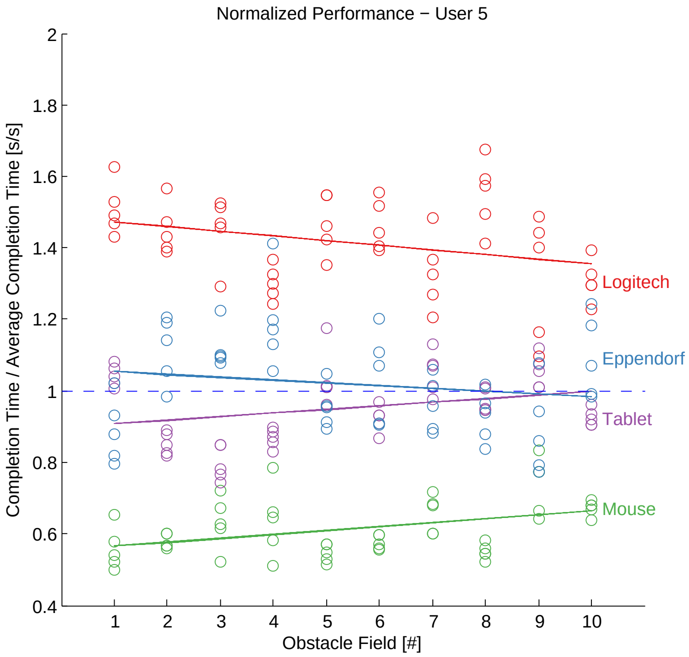


Figure 2.8: Single user hardware test performance data with linear fit for each device.

2.2.4 Qualitative Survey

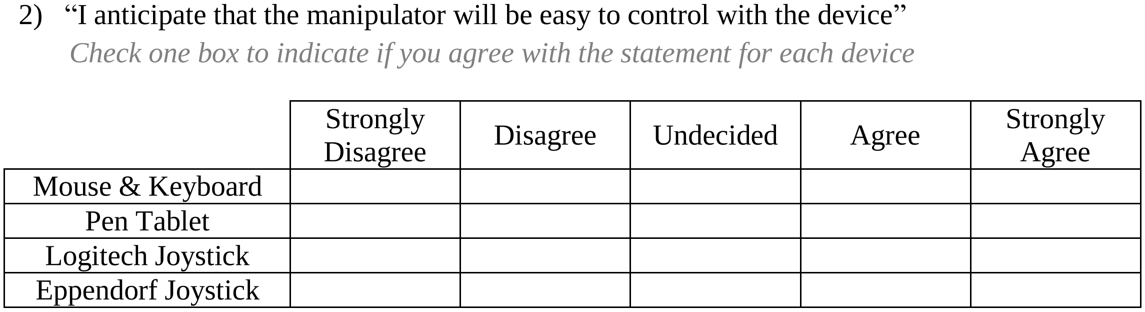
Effectiveness of movement is very important when performing laboratory operations. For a hands-on classroom demonstration however, students might only operate the manipulator for a brief period of time in which case qualitative first impressions are important. To understand the initial impression of the user they were given a pre-survey and post-survey. Three styles of questions were used. The first, shown in Figure 2.9, is a Likert scale that asks the subject to indicate their agreement with a statement. The second, shown in Figure 2.10, presents a statement and asks the user to rank the devices from 1st to 4th based on the given criteria. The final type was a single question that asked the user to circle their preferred device.

The pre-survey contained the following questions:

1. Rank the devices in order of anticipated effectiveness
2. I anticipate that the manipulator will be easy to control with the device
3. I anticipate that the device will be enjoyable to use

The post-survey contained the following questions:

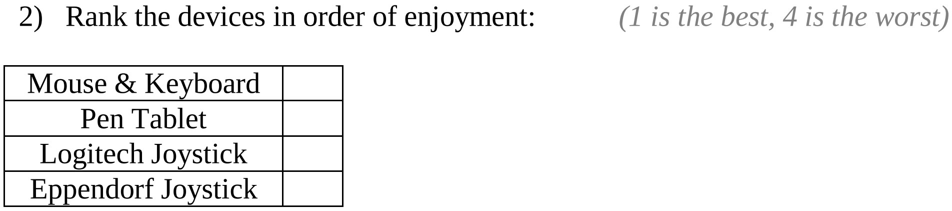
4. Rank the devices in order of effectiveness
5. Rank the devices in order of enjoyment
6. The manipulator was easy to control with the device
7. I got used to using the device quickly
8. This device is suitable for lengthy work requiring manipulation
9. Which device did you prefer overall?



2) "I anticipate that the manipulator will be easy to control with the device"
Check one box to indicate if you agree with the statement for each device

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
Mouse & Keyboard					
Pen Tablet					
Logitech Joystick					
Eppendorf Joystick					

Figure 2.9: Example of Likert scale question from hardware device study post-survey



2) Rank the devices in order of enjoyment: *(1 is the best, 4 is the worst)*

Mouse & Keyboard	
Pen Tablet	
Logitech Joystick	
Eppendorf Joystick	

Figure 2.10: Example of Ranking question from hardware device study post-survey

While these questions include self-evaluation of user performance, they do not replace the actual numerical data. Instead, these questions will elucidate how accurately the users perceive their personal effectiveness with the device. A Likert scale allows for ties between devices and rankings do not reveal the magnitude of the difference between devices. By asking the subject to answer both style of questions a ranking with magnitudes is obtained. All of the questions asked in the pre-survey have corresponding questions in the post-survey to obtain information on how the subject's opinion of the devices changed after the performance tests. The survey was designed in accordance with suggestions from *Research Methods in Psychology* [48]. Examples of both surveys are included in supplemental material.

Due to the relatively small sample size of the study, demographic information was omitted. Measuring the performance difference across variables like age and gender would require a much large sample size and is worth considering for future studies. Additionally, the users were not surveyed about their prior experience with each device besides the Eppendorf joystick. This omission may have prevented trends based on particular experience with the pen tablet or Logitech joystick to emerge in the data.

2.3 Results

8 subjects without prior experience with micro-manipulation participated in the study, completing 10 obstacle fields 5 times with all 4 devices representing 400 measurements of completion time per device. Data from subjects that had prior experience with manipulation or that did not complete the entire test sequence were not included; fortunately remaining data were sufficient to obtain statistically significant results ($p < 0.001$).

The most significant observation is relative device performance. The adjusted device performance of an individual user was averaged (P_{adj}) and ranked (P_r). The average of rankings for all users ($\overline{P_r}$) represents which devices consistently performed better or worse than others while the average performance across all users ($\overline{P_{adj}}$) provides a direct comparison. Table 2.2 shows these values alongside the unadjusted mean performance $\overline{P}$. The mouse was consistently ranked 1st while the pen tablet was consistently ranked 2nd. Despite their clear rankings, $\overline{P}$ was similar for the tablet and mouse. The Logitech 3D Pro was most often ranked 4th.

To ensure that the differences in means are statistically significant, a series of one-way analysis of variance (ANOVA) tests were applied to the data set. An ANOVA test between P_{adj} is conservative ($df = 31$) yet no $\overline{P_{adj}}$ overlaps with the range of another

Table 2.2: Average performance time $\overline{P}$, adjusted performance time $\overline{P_{adj}}$, and device performance rank $\overline{P_r}$, across all users.

Device	$\overline{P}$	$\overline{P_{adj}}$	$\overline{P_r}$
Mouse	3.009	0.687	1.125
Pen Tablet	3.514	0.807	1.875
Transferman NK2	5.105	1.142	3.250
Logitech 3D Pro	6.036	1.363	3.750

device. The mouse and tablet had the closest $\overline{P_{adj}}$ and an ANOVA test between them yields $p = 0.026$. An ANOVA test between devices using unadjusted data ($df = 1599$) yields clearly unique means with $p < 0.001$ between the closest means. Figure 2.11 shows a graphical representation of the ANOVA tests. Note that none of the device means (or medians) fall within another device's expected performance range represented by the indented region of the box plot. The results of these ANOVA tests indicated that the differences between device performance are statistically significant with at least 95 % confidence.

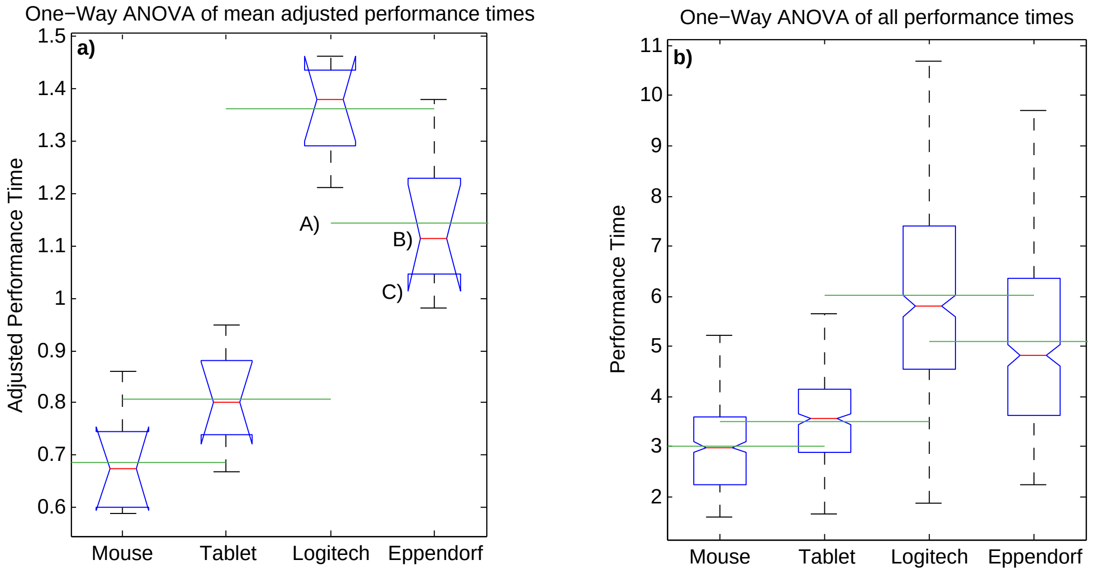


Figure 2.11: One-Way ANOVA test of user's mean adjusted performance between each device P_{adj} and all performance times between each device. Arithmetic means (A), medians (B), and range boundaries (C) are shown along with 25th and 75th percentiles and maximum and minimums (bars).

Standard deviations of (P_{adj}) were calculated for each user (σ_{adj}) and device and represent the consistency with which the user completed tasks with that device. $\overline{\sigma_{adj}}$, the mean of σ_{adj} across all devices is shown in Table 2.3. Notably $\overline{\sigma_{adj}}$ was the lowest for the tablet indicating that it was the most consistently performing device despite not being the best performing device. Additionally the Eppendorf joystick had the highest $\overline{\sigma_{adj}}$ with the 3rd best performance. One-way ANOVA tests were used to compare $\overline{\sigma_{adj}}$ between devices. The ANOVA test indicated that $\overline{\sigma_{adj}}$ for the mouse and tablet were not significantly different ($p = 0.177$) suggesting that they are comparably consistent.

Table 2.3: Averages of user performance standard deviations with each device $\overline{\sigma_{adj}}$ and the number of users that exibited relative performance with each device I_N.

Device	$\overline{\sigma_{adj}}$	I_N
Mouse	0.217	1
Pen Tablet	0.193	3
Logitech 3D Pro	0.337	5
Transferman NK2	0.407	6

There was no significant statistical positive or negative correlation between the user's qualitative assessment of how fast they adjusted to a device and their actual performance.

Accessibility was a primary focus of this experiment and as such it was designed to measure initial performance. However, proficiency can be gained with each device and at different rates. Because obstacle fields were randomly generated, transient trends in performance time P were not expected. Controlling for the obstacle field difficulty, as shown in Figure 2.7, allows relative improvement to be reflected in the slope of P_{adj} vs. obstacle field number. A negative slope indicates that the user improved more with that device than their average performance. A slope was considered confident if the goodness of linear fit between P_{adj} and obstacle field number exceeded 0.9. The number of times a user exhibited relative improvement with a device (I_N) is shown in Table 2.3. Notably more than half of the users showed more improvement with the Logitech 3D Pro and the Transferman NK2 than the mouse and pen tablet suggesting that the mouse and tablet are more intuitive.

Finally, Table 2.4 shows the aggregated results of the pre-survey and post-survey. Likert scale questions were aggregated using the sum of scores with 5 corresponding to ″strongly disagree″ and 1 corresponding to ″strongly agree.″ Similarly ranking questions were aggregated using the sum of rankings. In both cases a low Q_N corresponds with positive responses. Q_8 is simply the number of users that selected the device as their preferred one. As expected most users closely identified which devices they performed well with. The mouse and tablet were well received both before and after. The Logitech 3D Pro was very poorly received and user perception notably improved for the Eppendorf NK2 ($Q_2 < Q_4$). There was no significant correlation between the user′s qualitative assessment of how fast they adjusted to a device Q_7 and their actual performance. The notable results of the qualitative survey indicate that the Logitech 3D Pro is not a suitable device for educational where first impressions are important and the Transferman NK2 may not be effective for very short use durations where users are unable to obtain familiarity with the device.

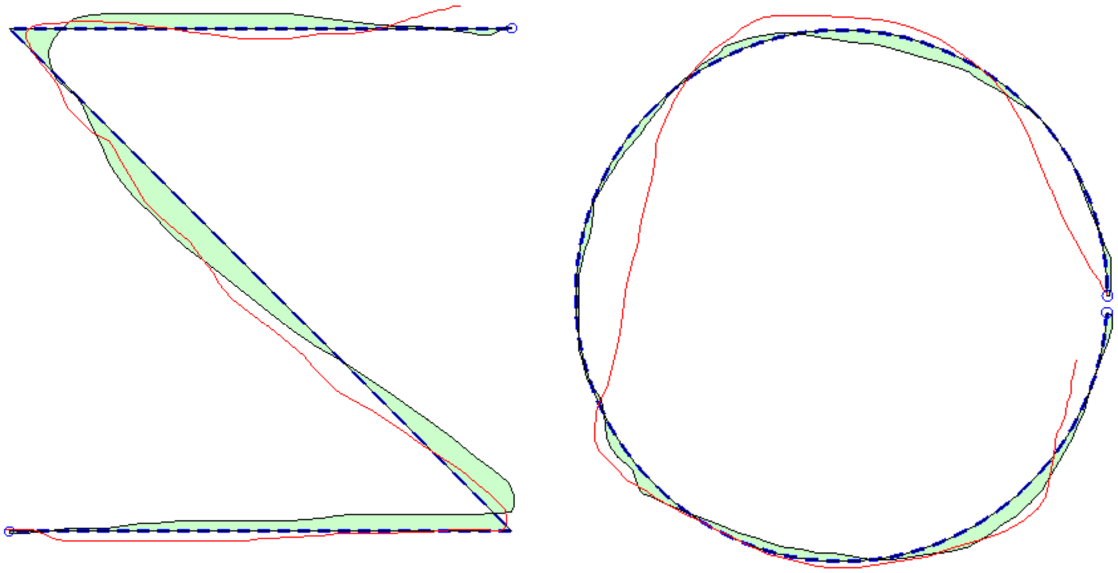
Table 2.4: Aggregated results of pre-and post survey. Questions correspond to those enumerated in Section 2.2.4

Device	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
Mouse	10	14	22	11	13	12	16	11	4
Tablet	26	23	19	13	14	21	19	20	2
Logitech	28	27	24	31	31	28	24	34	0
Eppendorf	16	20	24	25	22	17	17	11	2

2.4 Unused Tasks

As mentioned previously in Section 2.2.2, the design of the tasks in a virtual environment must closely match the use in a physical system. Some tasks influenced by prior research were fully implemented by ultimately not used because they were not indicative of normal use. The original implemented task was path following similar to the task outlined in [47].

The user is presented with a virtual environment containing a path defined by an ordered series of connected points. The user must navigate their cursor from start to finish while following the path as closely as possible. The deviation from the path can either be calculated as the total area formed by enclosing the user's path and the target path or by the RMS deviation at each point along the user's path. Additional thresholds can be set to monitor if the user deviate far enough from the path that the trial would be considered a failure. Figure 2.12a shows a completed trial with the enclosed area represented path deviation in green. Figure 2.12b shows a similar trial with a circular target path.



(a) Z

(b) Circle (n-gon)

Figure 2.12: Example tasks for path following hardware conroller performance test. Shown is the current path (red), the approximate error of the previous path (green) and the target path (blue)

The implementation of the task was successful but preliminary trials showed that results were greatly dependent on the chosen shape. The velocity joystick snapped to horizontal and vertical lines making paths shaped like rectangles very easy to follow. The snapping also made splines harder to follow for the velocity joystick. The pen tablet facilitated small splines to be followed precisely but, similar to hand-writing, straight lines were hard to follow.

Non-random obstacle field tasks were also implemented and once again the arrangement of the targets significantly biased the results with some devices but not others. Strong performance could be achieved with a particular device by tailoring the task to it. For this reason random target generation was used for this experiment and following experiments. Random path following tasks were not used because accurate path following is just as important as simply reaching a manipulation destination for most applications. More complex physics based tasks, similar to those used in [45], but performance of complex task are dependent on the individual users cognitive ability.

Chapter 3

Latency

3.1 Introduction

When utilizing a fully computerized manipulation system with digital optics the primary performance degradation introduced by long distance access is latency. The effects of latency have been studied for other applications including robotic telesurgery but these effects are not well understood for micro- and nanomanipulation. This chapter explores the effects of latency on the effectiveness of single probe manipulation with a variety of common hardware control devices. A virtual environment was created where an artificial latency can be imposed for the completion of a task simulating single probe manipulator use. The virtual environment can be operated without the use of a physical manipulator or with the use of the manipulator utilizing image processing to track the location of the probe. Users completed a variety of manipulation tasks with a position joystick, a velocity joystick, a computer mouse and a pen tablet while their performance was quantified over a range of imposed latencies. Qualitative satisfaction was recorded to determine acceptable latency ranges for remote operation of nano- and micromanipulators. The study revealed that as latency increased, user performance degraded the most for the joysticks and degraded less for both the mouse and pen tablet.

This chapter outlines the process and conclusion of studies designed to establish the effects of latency on nano- and micromanipulation. Three potential low-cost control devices for remote workstations were selected and compared against a proprietary manipulator control joystick. Each device was tested at a range of blind and random sampled latencies in a controlled virtual environment using three tasks that simulate normal manipulator use. The performance with each device was quantified and evaluated to observe trends in performance with increased latency and users were polled about which latencies they considered acceptable. This chapter concludes with guidelines for handling latency and selecting a hardware controller for remote workstations.

3.2 Methods

3.2.1 Control Device Selection

The latency testing was carried out with three hardware devices that are candidates for remote access workstations and one proprietary micromanipulator controller. The three main criteria considered when selecting a control device for a remote workstation were familiarity, portability, and cost. When using a familiar device, users are more likely to perform manipulations successfully [49]. Portability and low cost facilitate the implementation of a remote workstation with minimal difficulty. The candidate devices, outlined in Table 3.1, include a joystick, a mouse, and a pen tablet. These devices are the same as those in Chapter 2 with the exception of the Standalone Joystick. Each candidate device is low-cost and portable which is beneficial for remote access.

Joysticks are common manipulator control devices making them natural options for remote workstations. Joysticks vary dramatically in both price and design and typically operate using position or velocity control. A velocity (relative) control joystick returns itself to the center position usually by spring action and its displacement determines the velocity of the controlled actuators. A position (absolute) joystick does not return to center and the actuators are driven to match the relative position of the joystick.

The Transferman NK2, a position joystick, contains the control hardware for operating

Table 3.1: Summary of selected hardware control devices.

Control Device	Model	Description
Mouse	Dell MS111	Traditional PC mouse, widely available.
Pen Tablet	Wacom Intuos CTH640	Touch-sensitive tablet utilizing an electronic pen for pressure sensitive cursor control.
Custom Joystick	4Site 4SJ300	3-DoF (X, Y, Rotation) joystick modified to operate as a position joystick utilizing the same hardware as the Transferman NK2.
Standalone Joystick	Transferman NK2	3-DoF joystick with hardware controller for use with Eppendorf micromanipulators.

Eppendorf micromanipulators and has a 3 DOF joystick. A modified standalone joystick was created for latency testing utilizing similar hardware to the Transferman NK2. It consists of the same 3 DOF joystick as the NK2 but is interfaced with MATLAB by Serial connection to an Arduino programming board. For latency testing with both joysticks a mapping was assigned such that the edges of the joysticks range of motion corresponds with the edges of the virtual environment. The computer mouse was selected due to wide familiarity and availability. The pen tablet was selected because it was shown to be particularly effective at offering a high level of engagement when used for educational purposes [41–43]. Other manipulation systems utilize other controllers, such as haptic feedback devices [25, 50], but these devices were not utilized in this study because they are more expensive and less intuitive than the selected devices making them less suitable for wide adoption in the future.

3.2.2 Task Selection and Design

Virtual environment testing is a common tool for evaluating task performance and allows all aspects of a task to be quantified and timed while artificially imposing precise latencies. A virtual environment was developed to study the effect of latency on micromanipulator tasks. It is important for tasks in a virtual environment to simulate real tasks users perform during micromanipulator applications. Tasks were selected based on two primary requirements. The first requirement was that the task is similar to actual micromanipulation because the virtual environment must closely resemble the physical system.

The second requirement was that the performance of the task is quantifiable for numerical analysis. Furthermore, tasks that were easy to understand with minimal explanation were preferred. Three independent tasks were selected for this purpose and are listed in the order of least complex to most complex, as shown in Figure 3.1: Approach, Slow Approach, and Nudge. Other tests were considered and implemented but ultimately were not used in the study because they did not simulate realistic uses of the manipulators. An example of such a task was a line-following test with performance measured as the total deviation from the target line. Line following is quantifiable however manually driving a manipulator in perfectly straight lines is not a typical use scenario. In every task the visual feedback provided to the user was delayed by the imposed latency.

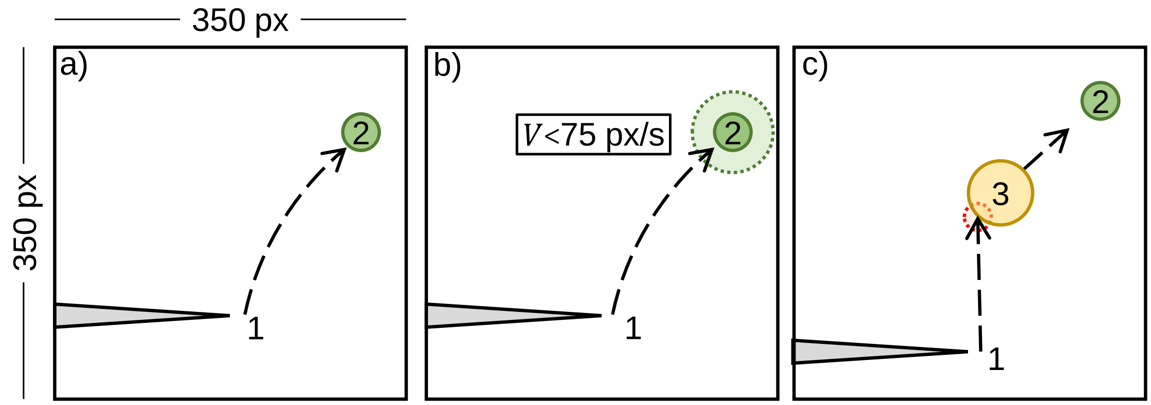


Figure 3.1: Illustration of the Approach test (a), Slow Approach test (b) and the Nudge test. The cursor (1), the current randomly generated target (2) and the virtual object used in the Nudge test (3) are shown.

The **Approach** task prompts the user to drive the cursor to a randomly generated target location. This task is the simplest and there is no way for the user to fail an attempt. The user must navigate the cursor to the randomly generated target at which time a new target is generated. The new target always appears completely within the visible region and to prevent the user from hitting it accidentally rather than deliberately, it is restricted to appearing away from the cursor by at least 20 % of the regions width. The duration of each completion is recorded.

The **Slow Approach** task is identical to the Approach task except that the target must be approached slowly. The velocity of the cursor is checked when it reaches the target and this velocity must be below a threshold of 75 px/s. If an approach exceeds this velocity a failure is logged and a new target is generated. Tasks simulated by a Slow Approach include some cell manipulations where a fast collision may damage the cell. It is important that the calculated velocity is independent of hardware and software factors like framerate. An internal clock is used so processing speed does not impact the calculation and a moving average of the last five frames is used to calculate the velocity because hardware devices like optical mice can ″jump″ resulting in large changes in velocity for single frames that are not desired by the user. The target generation restrictions are the same as for the approach test. The duration of each completion and number of failures at each latency are recorded.

The **Nudging** task prompts the user to maneuver a simulated mass to a randomly generated target by contacting it with the cursor. This task simulates the physics of moving a micro-object, such as a cell, with a manipulator end-effector. The velocity of the object decays in a manner consistent with viscous friction and when the cursor contacts the object it is imparted with a velocity proportional to the contact velocity. Similar to the Slow Approach, the physics of the simulation were implemented such that they were independent of the processing speed of the simulation to prevent undesired behavior of the update frequency of the simulation drops.

When imposing latency in a virtual environment it is important that the latency remain consistent regardless of system performance and framerate. The true location of the pointer was recorded in real time and each time the virtual environment updates the user was shown the cursor position from the frame with the Δt closest to the latency. The simulation variables were stored in a rotating array with enough elements to span the current imposed latency.

3.2.3 Testing Procedure

All four hardware devices were operated with blind and random sampled latencies ranging from 0 ms to 300 ms at 50 ms intervals in addition to 400 ms, 500 ms, 600 ms, 800 ms and 1000 ms representing a total of 12 latencies for each iteration. Each user trial consisted of 40 task completions of the Approach and Slow Approach tasks and 20 task completions for the Nudge task. Figure 3.2 illustrates the software flow of the test procedure.

When performing tests over a range of latencies, it is possible for the user to become more proficient at the task over short periods of time, or conversely to become fatigued by repetitions. These effects would bias the completion times based on the order that the latencies were testing. The significance of this transient effect can be observed by ordering the data by testing order rather than by latency. Figure 3.3 shows the trend from a random sampling of latencies ranging from 0 ms to 200 ms arranged by test order and a trend is clear. For this reason, a single random sampling of latencies was insufficient. To further decouple test order from latency, only 10 tasks were completed at a sampled latency and the same latency was selected multiple times throughout the process. This method allows for potentially uncontrolled transient variables to be evenly distributed across the range of latencies and not produce endogeneity. Figure 3.4 illustrates the difference between an ordered testing procedure where the user starts at a low latency and finishes at a high latency and the randomized procedure used in this study. In summary, it is important to randomly and blindly sample latencies multiple times when studying latency effects.

For tests with the mouse, pen tablet, and velocity joystick, the cursor location was recorded directly by the testing software. When performing tests with the Transferman NK2, the cursor was treated as the location of the probe tip, which is identified using real time image processing. The cursor location was then fed into the same virtual environment as the other tests. In order to not artificially impede task performance with the Transferman NK2, it was important that the image processing technique is fast enough

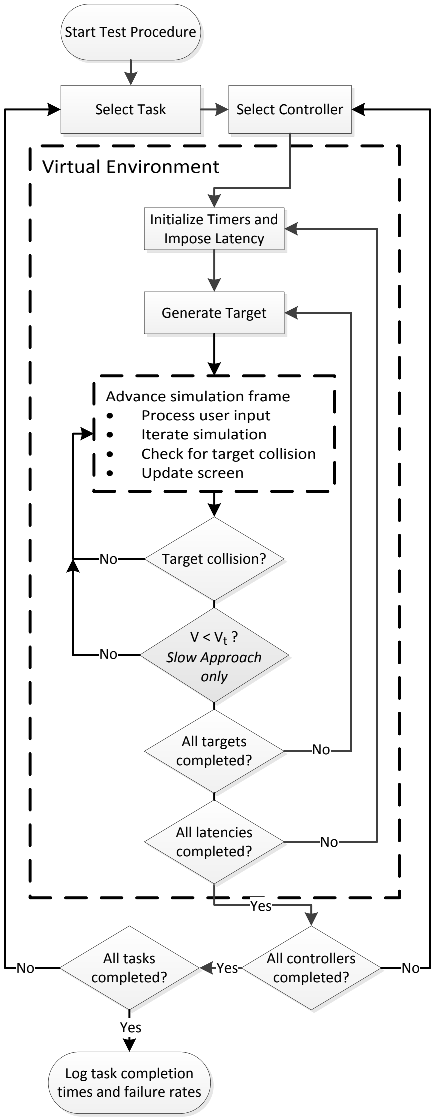


Figure 3.2: Overall test procedure including software flow for latency testing.

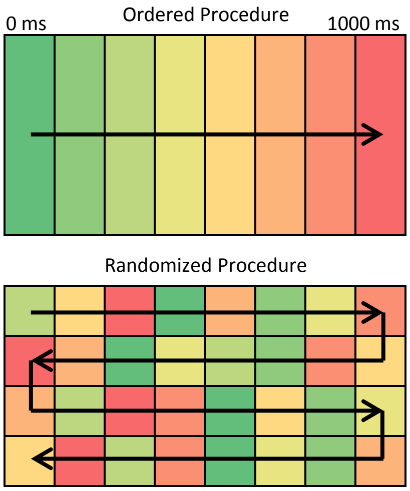


Figure 3.3: Illustration of an ordered test procedure (top) and a randomized test procedure designed to eliminate bias from fatigue (bottom)

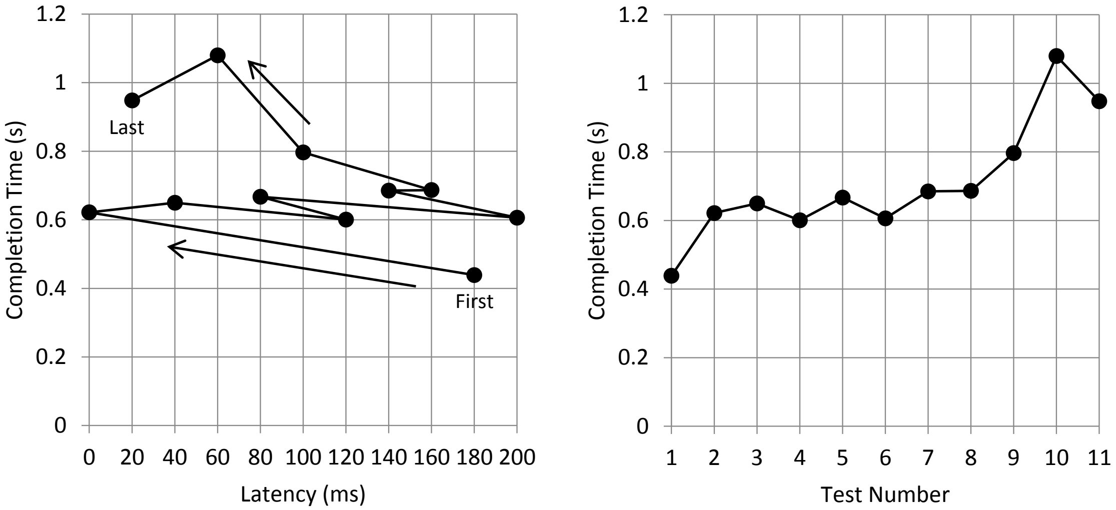


Figure 3.4: Average completion time for approach test at randomly ordered 20 ms latency intervals arranged by latency (left) and arranged in order of completion (right).

to sample at least as frequently as the virtual environment updates. Various image processing techniques were implemented and the fastest sufficiently accurate technique was selected. For each frame, the image was converted to binary matrix based on an intensity threshold and the rows and columns were summed to identify the top location based on the maximum row and right-most minimum column. This process required a horizontal probe arrangement like the one shown in Figure 3.1.

3.2.4 Data Analysis

The individual task completion times for each latency were used to generate an average completion time and a 95 % confidence interval for trend analysis. Because each latency was blindly sampled multiple times it was possible for the means to differ between tests at the same latency. Analysis of variance (ANOVA) tests were used to identify that these differences were statistically significant but no trend was observed within a single latency based on testing order. A lack of trend based on testing order indicated that transient effects such as fatigue did not contribute to the trend between latency and task performance. Every task completion data point for the Approach tests and Nudge tests were categorized by latency to find the average completion time at each latency with the corresponding 95 % confidence interval for the mean. The confidence intervals presented in this chapter were computed assuming a normal distribution, $\sigma_M = 1.98 * \sigma\sqrt{N}$, where N is the number of task completion times in the sample and σ is the standard deviation of the task completion times in the sample.

At each latency there was a minimum additional completion time added by the latency itself. For example at 100 ms of latency, even if the task was completed instantly, the completion time would still be 100 ms. For this reason the latency at every point was subtracted from the completion time yielding an adjusted completion time. If latency had no effect on task performance then the adjusted completion times would form a horizontal line and the absolute completion times would have a slope equal to 1 s of completion time per 1 s of latency. This adjustment eliminates the minimum possible performance decrease that would still be recorded by perfect control if latency was imposed.

Similarly to Chapter 2, demographic surveys were omitted due to insufficient sample size to establish differences across demographics. The users in this survey consisted of students ranging from 20 to 25. Furthermore, long term performance trends were not analyzed in this study. These trends could be assessed if users were to operate the devices at a particular latency on the scale of hours rather than minutes.

3.3 Results

3.3.1 Task Performance

Figure 3.5 shows the adjusted completion times for each device and task over the full range of tested latencies using a randomized procedure. As anticipated, every device exhibited deteriorating effectiveness with increased latency. Table 3.2 shows the results of fitting the data with linear regressions. The Approach task is well modeled by a linear regression and the joystick exhibited the greatest decrease in performance of the three devices, exhibiting a 2.3 s increase in task completion time for each 1 s of imposed latency, despite having relatively similar task completion times at low latencies to the other devices. This observation also occurs for the Slow Approach test and Nudge test, suggesting that the joystick was the worst performing device when operating at a high latency. The trend for the Slow Approaches was significantly less linear particularly for the pen tablet and the joystick. The Slow Approach simulates a task requiring precise and slow movements and these movements become much harder to achieve at high latencies.

Table 3.2: Linear regression slope and goodness of fit for completion times for each test and device.

	Approach		Slow Approach		Nudge	
Device	Slope	R^2	Slope	R^2	Slope	R^2
Mouse	1.80	0.979	2.32	0.914	2.28	0.870
Pen Tablet	2.04	0.993	1.52	0.882	2.85	0.931
Standalone Joystick	2.30	0.966	2.99	0.970	3.66	0.971
Transferman NK2	2.41	0.978	2.81	0.985	3.21	0.839

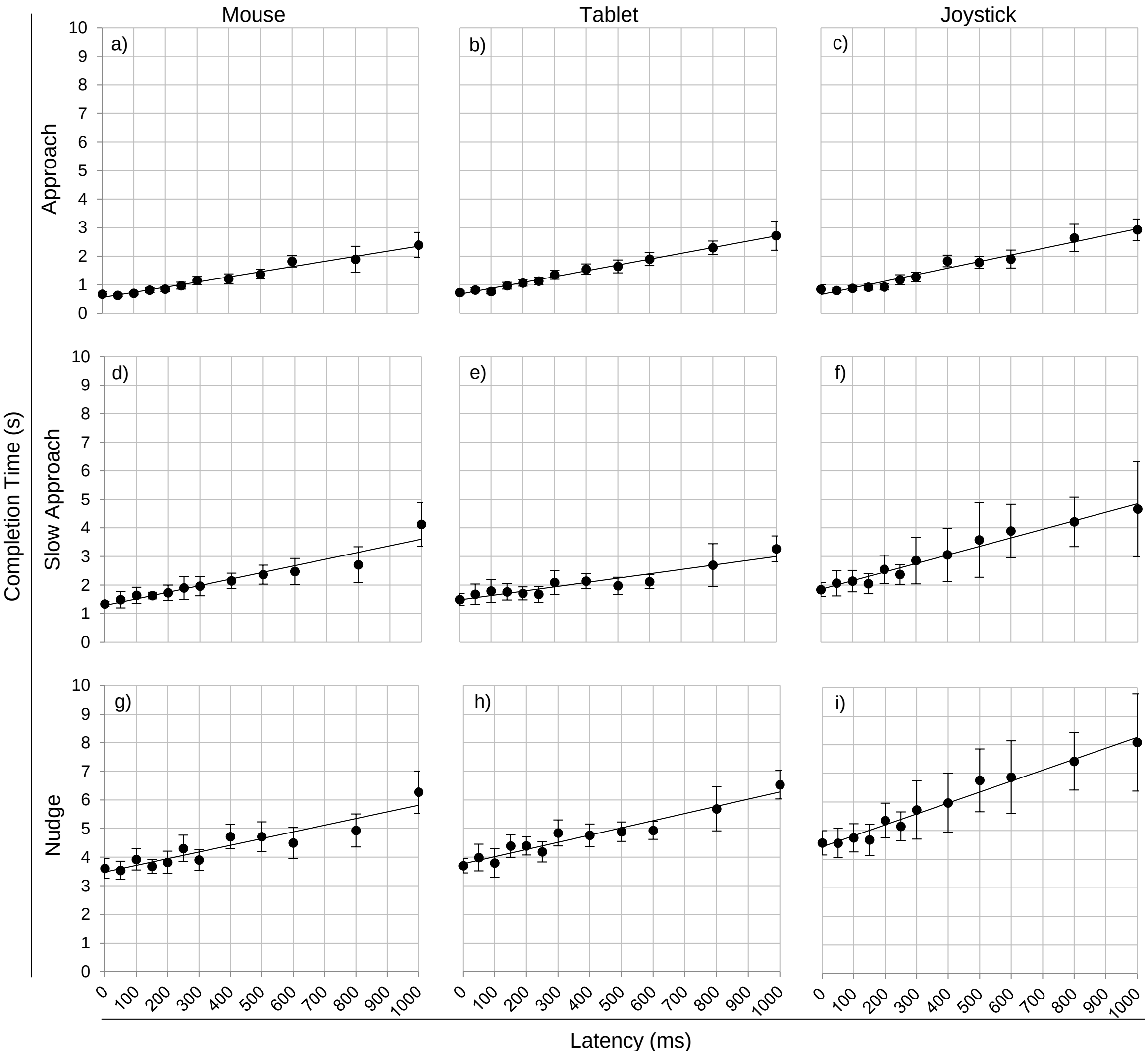


Figure 3.5: Results of each test including 95 % confidence intervals of the means using the mouse (a,d,g), tablet (b,e,h), and joystick (c,f,i).

For the Approach test, the mouse exhibited the lowest increase in completion times in relation to increased latency indicating that the device allowed users to handle latency the best of all the devices. Conversely the position joystick exhibited the most detrimental response to increased latency. Furthermore, the mouse and pen tablet had similar completion times at low latency but diverged at higher latencies.

The lowest detectable latency for visual feedback depends on many factors, including the highest velocity of an object and the visual acuity of the user. The transition from imperceptible latency to perceptible latency typically occurs below 100 ms for a rapid animation like a fast moving object or instantaneous color change [51, 52]. Examining the completion times at low latencies reveals that the 95% confidence intervals often overlap up to 200 ms, as seen in Figure 3.6. Overlapping confidence intervals are indicative of statistically similar task performance. While there is a positive correlation, the overlapping confidence intervals indicate that the performance degradation below 200 ms is minimal. The confidence intervals were considered alongside linear trends in performance when identifying regions of performance degradation.

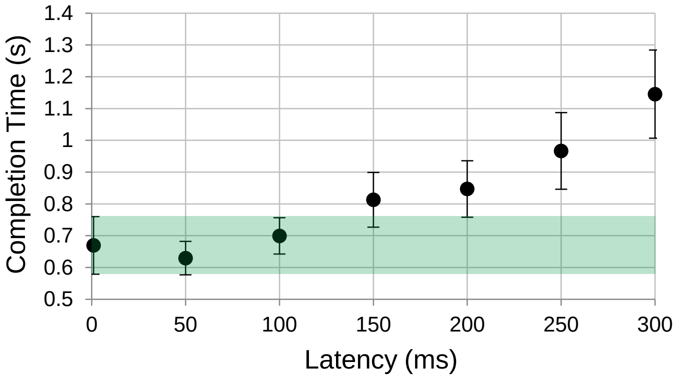


Figure 3.6: Results of Approach test using a mouse at 50 ms latency intervals from 0 ms to 300ms including a band corresponding to the 95 % confidence interval of the mean at 0 ms.

As latency increased, the ability to complete tasks with smooth continuous motions deteriorated for all task types and devices. This deterioration was attributed to latency preventing users from receiving visual feedback during each individual cursor motion. This effect can be quantified by analyzing the trend in the error term as latency increased. Heteroscedasticity (a trend in the standard deviation) indicated that the amount of variation in the data increased with latency. Comparing the linear regression of the confidence intervals in Table 3.3 shows the growing inconsistency of control with a device as latency increased.

The variation increased significantly with latency for the Approach and Slow Approach test with every device except the pen tablet Slow Approach. The heteroscedasticity exhibited in the task completion times were expected because tasks were completed with

Table 3.3: Slope and goodness of fit for the linear regression of the standard deviation between task completion time and latency.

	Approach		Slow Approach		Nudge	
Device	Slope	R^2	Slope	R^2	Slope	R^2
Mouse	1.139	0.878	1.267	0.819	0.905	0.859
Pen Tablet	0.995	0.855	0.587	0.294	0.460	0.220
Standalone Joystick	0.939	0.818	1.964	0.766	1.164	0.828
Transferman NK2	0.914	0.781	0.802	0.542	1.070	0.810

less consistency when users did not have reliable visual feedback. Occasionally the user completed a task at high latency quickly by moving the cursor precisely without waiting for feedback. However, if the user missed the target, latency delayed the following movement while the user waited for the cursor to catch up. It was difficult but possible to complete a task without waiting for any visual feedback if the user was proficient with the device. At very high latencies, users consistently attempted to complete tasks without waiting for visual feedback, indicating that they could not rely on it. Based on observation, users commonly used this strategy beyond 600 ms, revealing that these latencies were unacceptable. Further comparison between devices in Figure 3.3 are limited by the sample size of the study. Additionally, there were no observed trends in the frequency of failures at each latency for the Slow Approach. This lack of trend can be attributed to users reducing their movement speeds to compensate for increased latency. Overlapping confidence intervals indicate no significant performance degradation. Users pausing for visual feedback indicate severe performance degradation accompanied by increased deviation in completion times. Table 3.4 shows a summary of performance degradation over selected latency ranges. The standalone joystick yielded similar performance to the Transferman NK2, demonstrating its viability as a low-cost option with similar hardware for remote workstations. Figure 3.7 highlights the similarities between the task completion times between the standalone joystick and the Eppendorf joystick.

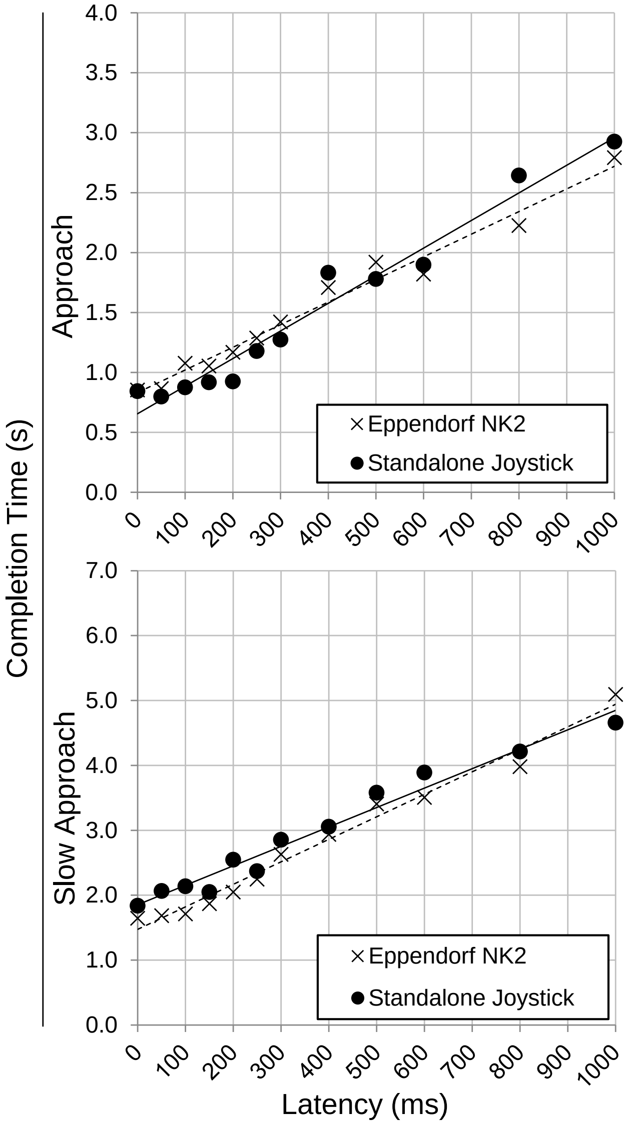


Figure 3.7: Comparison between the standalone joystick and Transferman NK2 for the Approach and Slow Approach tasks.

Table 3.4: Severity of performance degradation.

Latency Range (ms)	Mouse	Tablet	Standalone Joystick	Eppendorf Joystick
100	None	None	None	None
100-400	Moderate	Moderate	Moderate	Moderate
400-600	Moderate	Moderate	Severe	Severe
>600	Severe	Severe	Severe	Severe

3.3.2 User Perception

Six users were polled on whether or not they considered a particular latency ”acceptable for laboratory use” after each imposed latency during randomized testing. Figure 3.8 shows the frequency at which a particular latency was accepted or rejected. Users accepted 100% of latencies below 200 ms where as a majority rejection occurred at latencies of 400 ms and greater. At 600 ms and above nearly all users reported that they would reject the system. This latency threshold corresponded with the observations reported in Table 3.4. Additionally, during the qualitative acceptance tests, a user remarked that they struggled with task completion when moving from a high latency to a low latency simply because they had acclimated to the higher latency. This observation is consistent with research indicating that unpredictable feedback delays result in greater performance degradation than predictable delays [53]. An option to impose a minimum latency was added to the remote networking components of the GUIs outlined in Chapter 4.

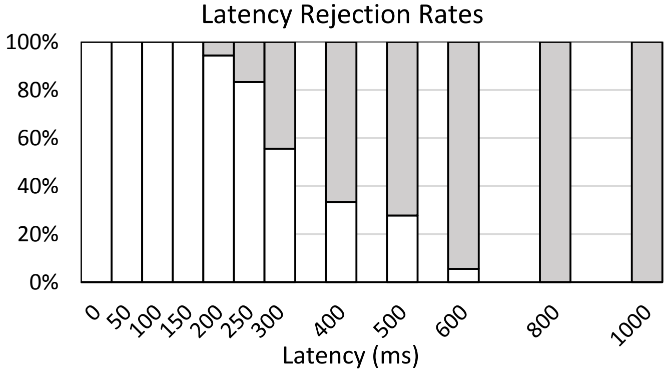


Figure 3.8: Frequency of qualitative acceptance (light) and rejection (dark) at each blindly and randomly sampled latency.

Interestingly, subjects in the study performed by Xu *et. al.* reported a higher tolerance for latencies above 400 ms. While precision during surgery is important, there are a number of reasons why users would be less tolerant of latency in micromanipulation. Quick movement, such as the ones involved in these latency tests, cause latency to be more noticeable and longer tests, such as those used by Xu *et. al.* [38], may allow for a user to acclimate to a higher latency than if they were working at a particular latency for a shorter amount of time. We also expect that medical students, like those in [39], may be more willing to accept a high latency system due to the necessity of the procedures. By extension, if a user is performing research with a manipulator, they may be more likely to tolerate higher latencies than a student using the manipulator in an educational setting because the student is not using the device out of necessity and can more freely reject the system. The observation that users would likely not reject an implementation involving latencies below 300 ms were consistent with [38], indicating that latency acceptance is similar between manipulation and surgery at low latencies only.

Chapter 4

Remote User Interfaces

The use of alternative control hardware, remote access, and customized automation features all require full computerization of a manipulator. A GUI and it's underlying control communication drivers represent the software element of the computerization. A well designed GUI is flexible and expandable to facilitate the aforementioned functionality.

4.1 Graphical User Interfaces

The goal of micro- and nanomanipulator automation is to create a system requiring only minimalistic intervention, optimally none, to carry out a specific desired function. In this way, automation of predefined tasks increases precision and throughput while reducing variability and time. Most automation requires computer vision algorithms to access the position of manipulation targets.

However, situations exist, especially in the development and utilization of new tools and techniques, where the automation and control of these manipulators, and other related equipment, needs to be flexible and adaptable. For example, in the NBIL's work towards developing CNT-based probes for single cell analysis [18,19], micro- and nanomanipulators are routinely used to maneuver the functional end of the probes in order to interface with single living cells in a non-routine manner, often requiring on-the-fly repositioning or customized movements based on qualitative visual feedback from the microscope. Here, the tips of CNT-based probes are manually maneuvered in Cartesian space by the manipulators joystick and positioned within the intracellular environments of single living cells under an optical or fluorescence microscope to perform functions or analysis with tertiary instruments.

New probe-based single cell analysis techniques, as well as traditional cell physiology techniques such as patch clamp electrophysiology, involve continuous interactions with multiple instruments simultaneously. The user is required to often switch attention and focus between the microscope, the micro- or nanomanipulator, tertiary instruments (*e.g.* electrophysiology amplifier), and computer screen (often displaying the field of view from a microscope camera and/or GUI of tertiary instruments), making the work difficult and laborious. Although many commercial microscopes and tertiary instruments come equipped with some form of GUI for use on standard computer workstations, no such interface has been provided for micro- or nanomanipulator control. Moreover, no interface exists as an expandable platform for the inclusive control of multiple instruments.

Two GUIs were developed to address this concern. The first is a robust GUI, expanded from the GUI started in the NBIL by Nicholas Hensel [1]. The robust GUI is feature rich however it may be computational expensive for some systems particularly when operated remotely. A minimal GUI was developed using many of the same concepts but including only the features required to operate the manipulator with a particular manipulator and camera configuration and a single movement type. The computational performance of the GUIs were compared.

4.1.1 Robust GUI

System Layout

A typical configuration for performing micro- or nanomanipulator operations was used to develop the manipulator control system. The system, shown in Figure 4.1, consists of four primary components: a manipulator and its control unit (Eppendorf TransferMan NK 2), a microscope (Zeiss Observer.A1m), a camera (Point Grey Chameleon), and a computer (Dell with Intel Core i5-2400 @ 3.1 GHz) to interface with all of the controllable

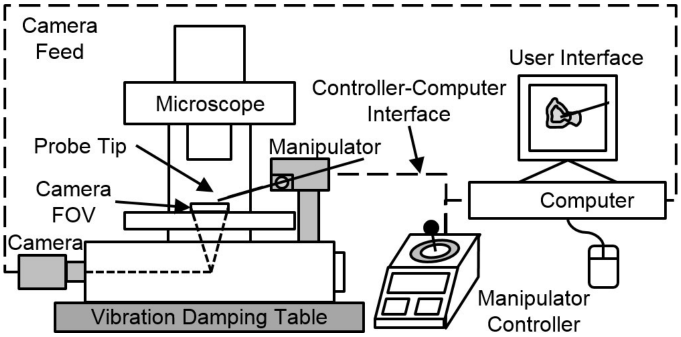


Figure 4.1: System layout illustrating the transfer of information between system components for manipulator GUIs.

components. The software was initially developed in MATLAB R2011a and updated for MATLAB R2014a, and again for MATLAB R2016a. The microscope is mounted on a vibration isolation table in order to minimize detrimental vibrations during manipulation operations. The computer and manipulator controller are located near the microscope but separated from the vibration isolation table. Each system component is a commercially available device with no hardware modifications.

While each of the system components can be reasonably interchanged and the control software adapted to the new equipment, the current implementation assumes a number of things about each of the components. The camera must be a device recognizable by MATLAB, which requires that it provide video information and accept computer commands through the MATLAB Image Acquisition Toolbox. This restriction prevents cameras with proprietary communication protocols from being used with the GUI without the use of additional drivers. The camera utilized in this particular hardware configuration was selected because the manufacturer provides control drivers, which specifically allow for open interfacing. There are however many microscope cameras which use restrictive or proprietary communication and control schema. To access information from more restrictive camera systems, it would be necessary to run a separate executable from the control software or develop device drivers which can allow MATLAB to interface with the camera.

The control scheme of the manipulator controller and its method of digital communication are critical to the design of the GUI. The selected control unit utilizes an absolute positioning system, wherein all movement commands sent to the manipulator are interpreted as a request to place the needle tip at the specified location in three-dimensional space by traveling at a specified velocity along each axis. The control software is designed to generate movement commands according to this control scheme. However, coordinate information is maintained both for the current position of the needle and the movement location, so that it would be possible to extend the software to support a manipulator which utilizes a relative positioning system. Additionally, the means by which the system is calibrated has been managed in such a way that it could be readily adapted to a relative positioning system. The manipulator selected has a range of travel of approximately 20 mm along each axis and can travel at up to 7.50 mm/s. The finest possible resolution of movement is approximately 40 nm. This allows for sufficient movement of the manipulator tip over a wide range of magnifications while also providing fine resolution for accurate manipulations at high optical magnifications. The Eppendorf TransferMan control unit is programmed to receive serial communication. Besides movement commands and coordinate requests, the control unit can receive commands to perform a number of other functions including connecting and disconnecting or toggling between manual and computerized control.

The schematic view of the system, shown in Figure 4.1, also illustrates the flow of information between components. The host computer controls the manipulator and camera using the control program developed in MATLAB. The computer interfaces with the manipulator controller over the serial port and with the camera over a Universal Serial Bus (USB) port. Information from the camera is sent to the computer, which sends and receives commands to the manipulator controller, and visual feedback from the movements is visible through the camera. In this way, a closed loop is created in the system. The microscope used for developing the software does not have any form of computer control, so this aspect of the system is not directly managed by the host software. More sophisticated commercial microscope systems do exist which provide programmatic control of the X, Y, and Z stage position.

Program Structure

Program flow is broken into three primary components: initialization, main loop execution, and termination. Program initialization consists primarily of the creation of the main control window and all of the control mechanisms for each of the graphics objects. After the end of the program initialization step, the graphics object is fully defined and the program enters the main loop.

The main loop of the program, shown in Figure 4.2, acts as a control scheme and continuously queries the graphics object for the current state of information. Given a change in the state of the graphics object, reference functions are called to carry out different actions based on the observed update in the state of the system. Such system changes are generated by user input in a variety of means. An example of this process might be the user pressing an interface button, which results in the execution of a callback function, which acts as an interrupt at the current point of program execution. In general, these callback functions can be executed at any time, but one callback cannot interrupt another callback. Within the callback, some element of the system is updated, such as changing the state of a figure objects value. The main loop then observes this update in the figure object when a check function is called. This check function is contained within the GUI object class and is used to observe the current state of some part of the system, possibly as newly updated by user input.

The hierarchy of program flow, as shown in Figure 4.2, is established to allow for multiple control schemes. The main loop has multiple sub-procedures that are invoked differently depending on which control scheme is currently active. In the loop, the system checks for movement, updates the tip coordinates, updates the graphics displays and restarts. If the user terminates the program, the loop terminates and the shutdown procedure is called. The series of movement checks in the main loop is the aspect of the program that enables control of the manipulator, and is designed in such a way as to allow multiple control schemes.

The first type of control checked in the movement cycle is continuous movement control, which allows for intuitive user control of the manipulator. In this control scheme, the program continually monitors the position of the mouse in the control window and sends commands to the manipulator to move to the queried position. This control scheme does not use any image processing feedback in order to provide real-time control with minimal movement and command latency. It does not check for completion of movements, so that the move command always corresponds to the exact desired position, without requiring completion of a possibly outdated command.

The second type of movement control checked in the main loop is driven by a series of user-specified point movements, which can be generated in a number of ways as described in the User Interface Layout section. This type of movement checks that the previous movement has been completed before starting the next movement. Furthermore, if feedback is enabled, the software adjusts the end effector position before the next move is loaded such that its observed position falls within a pre-defined distance from the commanded position. This is accomplished through loading the currently defined movement again using a new coordinate transform, which utilizes the calculated position data generated by tip detection in a proportional feedback control scheme.

Capabilities

The Movement Control sub-panel of the Manipulator Control Panel contains six different movement control schemes:

1. *Manual Calibrate XY Center*: Prompts the user to indicate the location of the tip in the field of view. This information is used to define the image-manipulator transformation, using a purely translational model. After obtaining the transformation, the tip

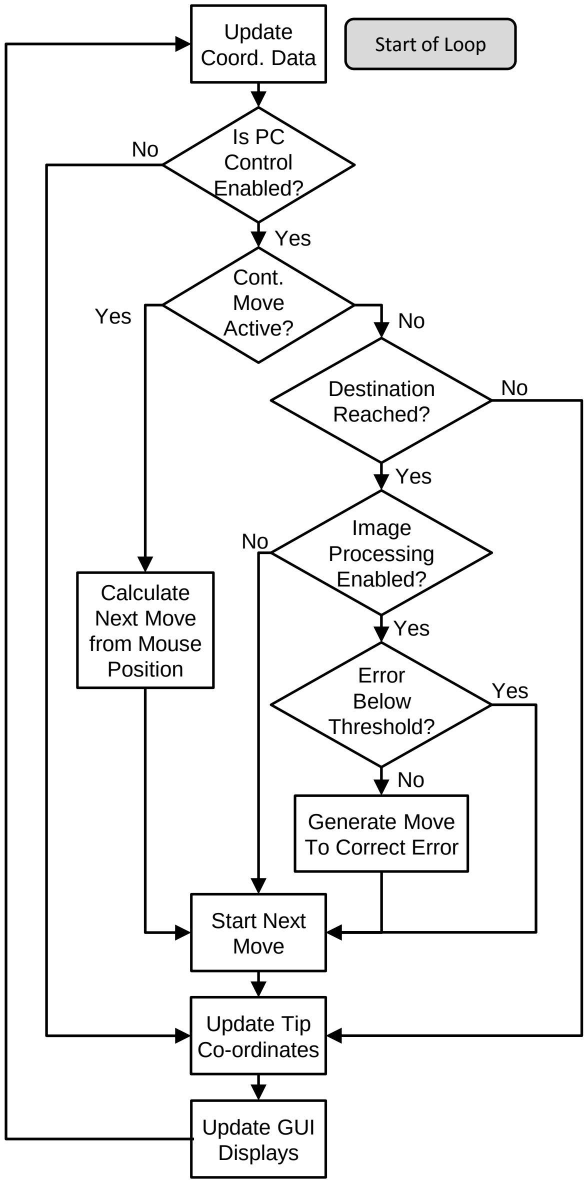


Figure 4.2: Software flow of the main loop that processes movement commands and updates coordinate information. The progression of the main loop is from top to bottom.

is moved to the center of the image. This process can be carried out whenever the operator desires to calibrate the GUI and the hardware systems, and should only need to be carried out once each time the system is initialized.

2. *Preconfigured Movement*: Prompts the user to select an Excel spreadsheet file containing a series of Cartesian coordinates. These coordinates are loaded into the program as a list of movement commands to be immediately executed by the manipulator.
3. *Single Move*: Prompts the user to select a point in the camera field of view. The tip is moved to that point.
4. *Return to Zero*: Returns the tip to the center of the image.
5. *Multi-Move*: Prompts the user to specify a series of points using mouse clicks. When the user is finished and presses the Enter key, the manipulator moves to each of these points in series.
6. *Continuous Move*: This is a toggle-able control scheme for manipulator control. While active, the tip of the manipulator is continuously driven to the currently detected mouse position within the field of view. This is done by repeatedly polling the current position of the mouse and sending the detected position as movement command to the manipulator. The system does not check that the last movement is completed, so that it is possible to smoothly control the device. In this scheme, movement is bounded by the field of view of the image display panel. This scheme does not use feedback, even if enabled, in order to minimize response delay and provide the user the same feel as if using the manipulator joystick.

Manual XY calibration is necessary because some manipulators including the TransferMan NK 2 allow manual adjustments to the probe that are not measured by the device. For example, setting the probe to a different angle will change the tips location relative to the manipulators actuators. Once the XY position is calibrated, no other calibration is required because the movement scale factor is automatically calculated by the software using information such as camera resolution, image display panel size and zoom level. The user only needs to ensure that the GUI is set to the correct zoom level and scaling calibration is handled automatically to further maintain ease-of-use.

The interface allows for other simple commands for ease-of-use purposes. All movements can be immediately stopped by pressing the Esc key. Another movement cannot be started while a previous movement is still being executed. The movement speed affecting all movement types can be changed at any time between movement commands. This movement speed, adjustable in the GUI, represents the maximum movement speed in micrometers/second that the end effector will be moved. When Continuous Move is enabled, the tip can be moved more slowly if the user moves their mouse slower than the specified speed and if the mouse is moved abruptly the end effector will follow at the set speed. This limit is useful if the user wishes to limit the speed of objects being manipulated. Multi-Move can be used to carry out predefined movements using static instructions, such as in Figure 4.4, or instructions can be generated by other means to create dynamic instructions.

4.1.2 Minimal GUI

A computationally expensive GUI on a low end PC can cause significant processing delay. Chapter 3 outlines guidelines for acceptable delays and combining round trip network latency with significant processing time increases the likelihood of operating in a detrimental latency range. This issue motivates the creation of a minimal GUI that exchanges features for performance. A GUI was created using many of the same concepts outlined in Section 4.1.1 and contains the following major differences from the Robust GUI:

1. A single movement type is utilized.
2. Microscope and camera information is written into the software rather than identified dynamically.
3. The viewing area cannot be resized during operation.
4. The Windows cursor and keyboard are the only source of inputs to the system.

The single movement is a combination of *Single Move* and *Continuous Move.* When the left mouse button is pressed the manipulator is driven to the cursor location. When the left mouse button is held down every iteration of the Minimal GUI's main loop recognizes the input the same way as a click. The end result is a continuous series of single moves updating as fast as the GUI iterates mimicking continuous movement. This method combines the advantages of *Single Move* and *Continuous Move* while minimizing computational complexity.

The Robust GUI is easily extensible to other microscopes and cameras. Camera information is populated automatically so as long as the camera is natively compatible with MATLAB it will work with the GUI. The Robost GUI can be used with different microscopes and objectives but providing information about magnification. This information is hardcoded in the Minimal GUI which saves on startup time and allows a simplified co-ordinate system to be used resulting in less computations for each loop of the GUI. The Minimal GUI can be used with other hardware, but it must be changed in the main MATLAB m-file. If a compiled version of the GUI is being used it must be recompiled.

To use a simplified co-ordinate system that maps cursor position to manipulator position on a 1-to-1 basis, the viewing area cannot be resized without reinitializing the GUI. Furthermore this approach does not rely on windows resize callbacks. Finally, multiple hardware devices have been interfaced with the Robust GUI which requires a number of additional functions to handle unique elements of the control devices. For example the pen tablet must be interfaced directly with MATLAB so pressure sensitivity can be used for Z-axis control. The minimal GUI uses the windows cursor directly. Alternate hardware devices can be used if they are controlling the Windows cursor.

4.2 Performance Evaluation

Figure 4.3 demonstrates the use of Continuous Move and Multi-Move. The Continuous Move shown on the left column took 8 seconds to complete and was performed by lowering the tip with the mouse wheel and moving the mouse in a W pattern. The Multi-Move command was completed by lowering the tip and specifying the 5 points of an M shape. The manipulator completed the movement in 5 seconds.

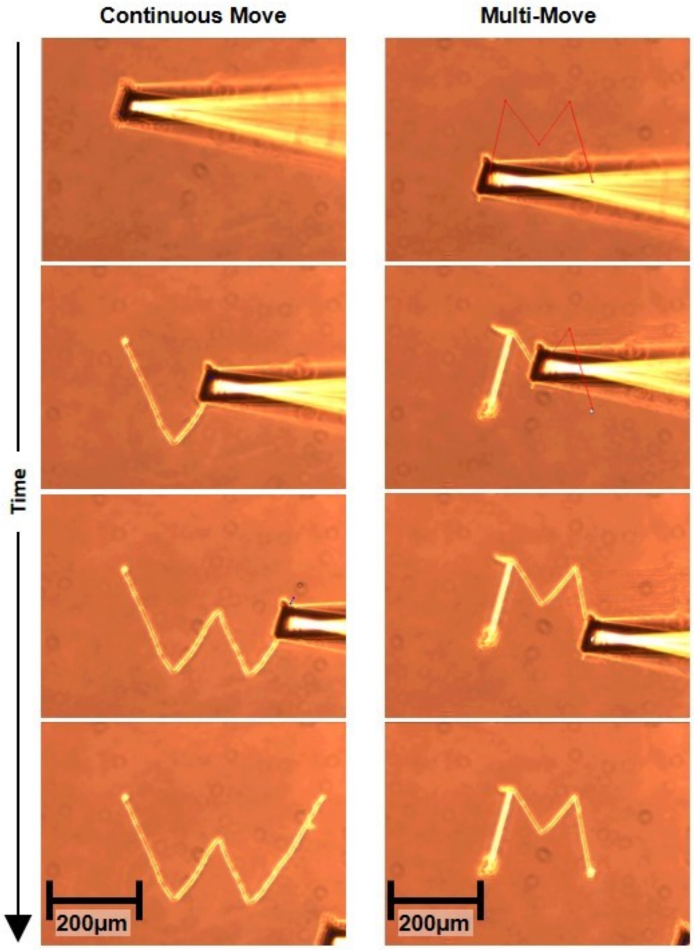


Figure 4.3: Demonstration of Continuous Move and Multi-Move functionality at 10x magnification. The medium is negative photoresist baked on a glass slide at 300 °C for 2 minutes.

Figure 4.4 shows the result of an example movement command carried out by the system using the Preconfigured Movement feature. To demonstrate the capability of this command, the acronym of our research laboratory, NBIL, has been indented into a film of positive photoresist deposited on a slide using a glass pipette tip. The use of the prerecorded movement requires a file with XYZ coordinates for each point. The file used

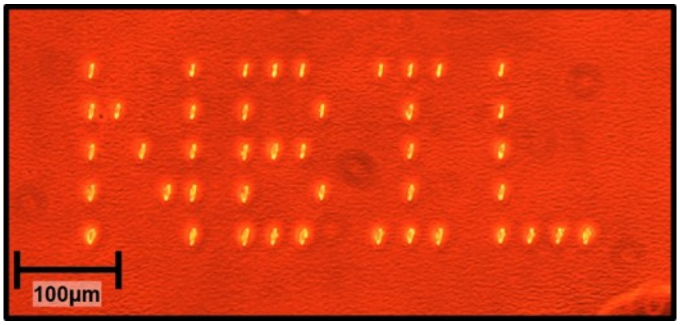


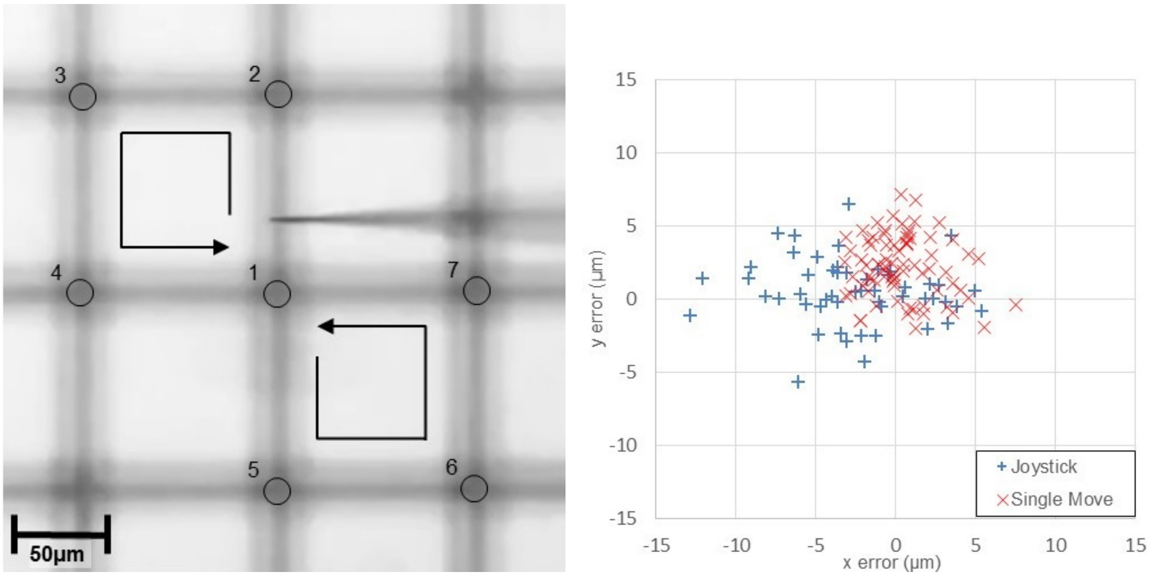
Figure 4.4: The letters NBIL etched automatically in negative photoresist using the NBIL GUI with an Eppendorf Transferman NK2 micromanipulator.

to create Figure 4.4 was generated with a MATLAB script that records mouse-clicks and converts them to a coordinate file. For each mouse click, the resulting file contains a point that moves the tip to the specified position, then two more to raise and lower the tip. The process required the manipulator to make 43 indentations, which was completed in 35 s, representing a rate of 1.23 Hz. The patterns in Figure 4.3 and Figure 4.4 were completed while the manipulator speed was set to 1000 μm/s.

To evaluate the accuracy and speed of GUI versus manual operation, a user without prior experience with the GUI was asked to repeat a simple movement task with the traditional manipulator joystick as well as with GUI movement. The three movement types compared were Single Move and Continuous Move with the GUI and joystick movement without the GUI. Low magnification (10x) was used. The user was tasked with maneuvering the manipulator tip to seven predetermined targets on a 0.1 mm grid, as shown in Figure 4.5a, over several cycles. Each cycle consists of eight target visits because the first target is visited twice per cycle. The recorded movement was then analyzed using open-source software[1]to obtain the location of the tip at each target.

Each time the user visited a target, the tips position was recorded and the absolute deviation from the target was calculated. The absolute deviation of every target visit for Single Move and joystick control is shown in Figure 4.5b. The deviations from all targets and all cycles for each movement type were averaged to calculate the mean error, $\overline{E}$. The

[1] Tracker, https://www.cabrillo.edu/~dbrown/tracker/



(a) Figure 8 pattern used for performance evaluation.

(b) XY positional error from Figure 8 task for Single Move and NK2 Joystick control.

Figure 4.5: Comparison of accuracy between Robust GUI and traditional manipulator control.

Table 4.1: Average error $\overline{E}$, average move duration $\overline{\Delta t}$, and clustering values C for control with and without the GUI.

	$\overline{E}$	$\overline{\Delta t}$	C
GUI - Single Move	3.43	1.93	1.544
GUI - Continuous Move	3.80	4.19	1.133
Joystick	4.58	3.17	0.735

duration of each movement was also recorded and averaged to obtain the mean movement duration, ($\overline{\Delta t}$). To determine if the error about any of the seven targets was consistent, the clustering value, C, was calculated according to the formula $C = log_{10}(1 - \frac{E_c}{\overline{E}})$, where E_c is the error of the average tip position about a particular target. A high clustering value (> 1) indicates that the direction and magnitude of the errors about a particular target are consistent. The mean error, mean movement duration, and highest clustering value for each movement type is displayed in Table 4.1.

The mean errors for GUI Single Move and GUI Continuous Move of 3.43 µm and 3.80 µm respectively are very similar but the mean move time for Continuous Move was 4.19 s, over twice as long as the mean move time of 1.93 s for Single Move. In general, a user may sacrifice accuracy with any control method by operating the manipulator faster. A much higher mean movement time and a similar mean accuracy indicates that it takes the user longer with Continuous Move than Single Move to achieve similar accuracy.

The mean error for the traditional joystick control was 4.58 µm, greater than both GUI control methods. The traditional joystick mean movement time was 3.17 s, which was faster than Continuous Move but slower than Single Move. Traditional joystick control was 21 % less accurate but 24 % faster than Continuous Move indicating that these control methods are similarly effective. Continuous Move was meant to allow mouse control imitate traditional joystick control and these results indicate that they methods are indeed similarly effective. Single Move was designed to be the most effective method of manual control through the GUI and its effectiveness was demonstrated by improved accuracy and faster movement times. When asked their opinion of which control method was most comfortable and efficient, the user reported that the Single Move felt the most effective while the Joystick was the most comfortable. Comfort was not a major emphasis of this research and may largely depend on the PC's mouse; the GUI is compatible with any commercial mouse and keyboard combinations.

4.3 Remote Access Implementations

The three major software subsystem components of the nanomanipulator are the computer input controls, GUI, and the remote control access. Various flexible methods of enabling remote access were explored. Each implementation was designed so that they could be applied to existing GUIs, such as the Robust and Minimal GUI, by handling user input and visual output and transferring over the Internet. Both the Robust and Minimal GUIs have been augmented to support this enhanced functionality. The software support for the network-based capability has been implemented using Java code. The Java components of the software subsystem are instantiated within the MATLAB-based GUI through the MATLAB Java Runtime Environment. This allows for a simple integration between the GUI and the network architecture, within minimal software deployment installation requirements. The network support provided is based upon the standard TCP/IPv4[2] protocol suite and the User Datagram Protocol[3], allowing for connectivity over the public Internet. Controls are available to the user to allow for modification of the quality of the real-time visual feedback, allowing for a performance adjustment at run-time depending on actual overall network connection speeds. Compression is performed by the network software to improve performance and decrease required bandwidth. A visual refresh rate of 30 frames per second is desired, and the networking components seek to achieve this ideal rate with an image resolution of 720-by-480 pixels or greater.

Two standalone software GUI applications were developed and are required for use of the manipulator: the server and the client, separately. The server application is launched on a computer connected to the nanomanipulator and to the microscope camera (the local computer.) The server program provides a basic GUI showing the status of the manipulator system and also Internet connection information. The client application provides a connection dialog into which the end-user enters the connection information provided by the server GUI (the remote workstation.) Upon successfully connecting the client application provides a real-time video feed from the microscope camera and input control is accepted. The software GUI network components created with MATLAB and Java are built into distributable executables using the MATLAB compiler. The requirements to use the distributed server and client application executables are only the Microsoft Windows operating system (OS) and the royalty-free MATLAB Compiler Runtime. A standalone MATLAB networking packaged was also developed without JAVA packages. The MATLAB seeks to maximize performance but lacks stability and the ability to re-establish a connection automatically.

A number of commercial remote access software implementations are available and were investigated to form a basis of comparison. Secure Shell (SSH) is a network protocol

[2] Transmission Control Protocol/Internet Protocol version 4, suitable for control.

[3] User Datagram Protocol, suitable for video.

commonly used to enable remote logins to a network. XQuarts (X11) is a graphical framework that can be used in conjunction with SSH to access graphical applications remotely. Windows Remote Desktop Protocol (RDP) is proprietary protocol available natively on most Windows operating systems and facilitates remote logins from local user accounts. TeamViewer is a third-party application for commercial operating systems allowing for simultaneous remote and local access. Each of these implementations have drawbacks. SSH with X11 forwarding is not available from Windows based servers meaning that the Local Model of the remote system would require a Linux based OS. RDP does not allow a local and remote user to operate the system simultaneously. RDP and TeamViewer both broadcast the entire desktop rather than only the MATLAB GUI causing a significant drop in performance. Table 4.2 outlines the performance difference between these implementations. Both workstations were wired by Ethernet on RITs Local Area Network (LAN) to minimize network latency as much as possible. Initialization time was the amount of time required to establish the connection and was measured from initial control to successful connection. Frame-rate was measured by transmitting a gradually changing gradient with MATLAB while recording the rate at which changes were reflected at the remote client. Latency (round-trip) was measured as the average time required for an input to be reflected at the remote client. The custom implementation was outperformed by SSH and RDP, with RDP performing the best. TeamViewer had the fastest initialization time though this metric is less important than framerate and latency.

Table 4.2: Comparison of JAVA, SSH, RPD and TeamViewer remote access methods including initialization time ($n = 5$), frame-rate and latency ($n = 100$)

	TeamViewer	MATLAB	JAVA	SSH	RDP	
Initialization Time	6	31	34	12	16	s
Framerate	2	21	8	19	32	fps
Latency	341	37	189	72	55	ms

Chapter 5

CNT Array Characterization

To test the new latency metrics, an automation process that would benefit the NBIL was implemented and timed to understand if it could run in real time. This chapter outlines an image processing based technique to characterize CNT array devices for enhanced cell transfection. Investigating how manufacturing parameters affects CNT array geometry, and how geometry affects transaction, requires the arrays to be measured. Obtaining a statistically sufficient number of measurements by hand is tedious and subject to human error. An automated system to characterize the arrays facilitates data collection of numerous pore properties. Scanning electron microscopy (SEM) images are pre-processed to identify the location of CNTs which are measured individual to obtain their characteristics. The data from single pores is aggregated to generate a numerical summary of the array parameters. Stereomicroscopy techniques are used to measure the heights of the CNTs using pairs of tilted images. The overall technique accurately measures the parameters relevant to cell transfection significantly faster than manual measurements while eliminating human error and bias.

5.1 Introduction

Cell transfection is an important tool for numerous biological research applications including disease study. Carbon nanotube (CNT) arrays are used to transfect thousands of cells simultaneously. The NBIL has demonstrated effective manufacturing of CNT arrays by depositing carbon in anodized aluminum oxide (AAO) templates and etching away the surface to expose the tips of the CNTs [54]. The process yields a porous surface through which cargo can be transfected into cells.

The performance of a CNT array is largely dependent on the geometric surface properties of the array [55,56]. The dimensions and shape of the tubes as well as their spatial density contribute to the effectiveness of enhanced transfection [57,58]. Bulk characterization of the arrays is extremely tedious if done manually and requires multiple measurements for every CNT. The accuracy of manual measurement is subject to human error and bias. Automatic characterization allows entire arrays to be analyzed accurately and quickly with added confidence due to the elimination of human bias, eliminating a common bottleneck of CNT array experimentation.

CNT arrays are imagined using electron or scanning probe microscopy (SPM) because the resolution of the geometric features exceeds what can be imaged by traditional optical microscopy [59]. Scanning electron microscopy (SEM) is common for this application [60–62]. An SEM detecting secondary electrons generates contrast corresponding to geometric features. An SEM detecting backscattered electrons generates contrast corresponding to material properties as well. The primary concern of CNT array characterization is surface geometry so secondary electron detection is used.

There are a number of other nano-scale surface characterization methods relying on microscopy or spectroscopy techniques. SPM including atomic force microscopes (AFM) are particularly useful for measuring small geometric features. AFM is capable of producing three-dimensional data facilitating convenient geometric measurements. However, some experimentation requires capabilities that AFM does not allow, such as tilt. X-ray microtomograpy (micro-CT) has been used to characterize CNT scaffold porosity [63], but lacks the resolution for individual CNT measurement. Nano-CT is a promising technique for characterizing CNT arrays [64], but is not widely commercially available. Spectroscopy techniques including Raman spectroscopy and X-ray photoelectron spectroscopy

(XPS) are more suited towards measuring non-geometric material properties. By utilizing a software-based approach to characterize the CNT arrays, researchers can determine their geometric properties without relying on additional expensive and often complicated hardware.

Surface characterization is beneficial at multiple steps in the template based CNT array fabrication process, notably to measure the geometry of template and the finalized CNT array. The membranes used in our procedure, a Whatman Anodisc 13 (Cat. No. 6809-7023), have a surface region of variable depth in which pores are irregularly shaped and spaced. This region is refereed to as the lattice layer. In order to obtain desired surface characteristics this layer must be removed up to a point where the geometry is sufficient. Experiments relating to the removal of the lattice layer require surface characterization for each sample. Finally, as desirable CNT array geometry is identified, fabrication techniques can be evaluated based on surface characterization without having to test cell transfection and viability with every device.

Because SEM is 2-dimensional, a single image with contrast corresponding to spatial geometry is not sufficient to accurately determine surface height. It is possible to obtain accurate measurements in the 3$^{\text{rd}}$ dimension by rotating or tilting the microscope stage and comparing the resulting images. Li *et. al.* demonstrated that existing methods of 3D projections obtained through tilting can be utilized on SEM micrographs to obtain nanoscale height measurements [65, 66]. Tafti *et. al.* has created a successful point-cloud based surface reconstruction system for SEM imagery [67, 68]. At minimum two images of the same sample separated by angular tilt in the plane of the sample can be used to obtain measurements along the direction of the electron beam.

This chapter outlines the pre-processing techniques and numerical analysis used to complete a method of characterizing CNTs utilizing SEM micrographs. A number of established image processing techniques are utilized including morphological operations, and region identification. Once identified, each tube is analyzed independently then aggregated to generate statistics about the CNT array geometry. The system was validated through a comparison to manual measurements and observation of false-positives and false-negatives.

5.2 System Overview

The overall software flow of the system is shown in Figure 5.1. The system utilizes an SEM micrograph as an input. Scaling information can also be provided if it is not present in the image. If no scaling information is provided the output unit will be. To obtain height measurements, a set of two SEM micrographs of the same viewing area must be provided. The first image should be oriented flat, such that the surface is normal to the electron beam. The second image should be tilted by approximately 5°. The tilt, θ_s must also be specified as an input. A Tophat filter is utilized to obtain uniform regional intensity in the case that one region of an image is darker than the other edge. Dynamic contrast thresholding is applied to separate the image into three layers, each representing a geometric feature including the CNT walls, pores, and membrane surface. Pairs of pores with their associated walls are treated as independent regions which are processed separately. Shape analysis identifies which regions are tubes, rather than other geometric features, and calculates several geometric properties. Each region corresponding with a tube is superimposed with the optional tilted image and features are matched to calculate their height based on lateral displacement. The resulting information for each region is aggregated for numerical analysis and the image is reconstructed.

5.3 Pre-Processing

Before structural analysis takes place regions must be identified that are potentially pores. The pre-processing makes intensity adjustments to the image and converts the image to black-and-white based on contrast thresholding to identify candidate regions representing paired pores and CNT walls.

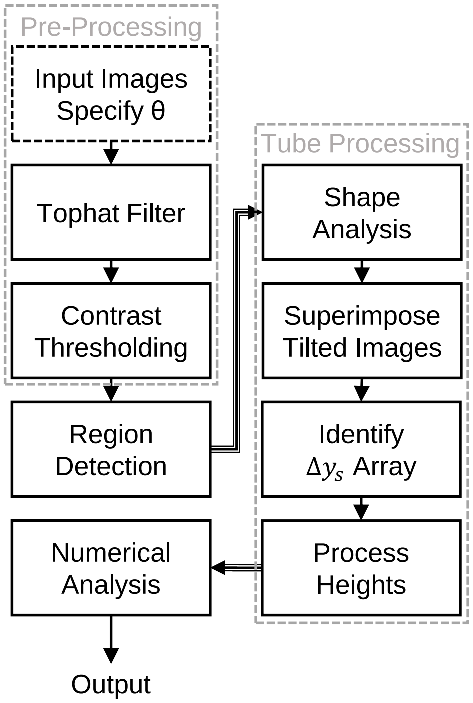


Figure 5.1: Software flow of CNT-array characterization system.

5.3.1 Tophat Intensity Filtering

Because intensity is the source of geometric information it is important for the intensity of the images to be uniform in order for the same intensity related parameters are used throughout the whole image. Because a tilted SEM stage will cause one edge of the sample to be closer to the electron beam a intensity gradient is normal. Variable overall intensity can also occur in an SEM micrograph if a region is shadowed by a larger object. A Tophat filter adjusts the average intensity of large regions such that the regions have matching average intensities. The intensity of large regions change while maintaining contrast of fine geometric features. It is important to specify a structuring element[1]larger than the geometric features. A disk-shaped structuring element with a diameter, d_{st}, equal to 20 % of the image's smallest dimension is used by default but d_{st} can be manually provided if pore diameters exceed d_{st}.

[1] A structuring element is a shape used to probe an image. Many image processing techniques function pixel-by-pixel considering all neighboring pixels that fall within the structuring element [69].

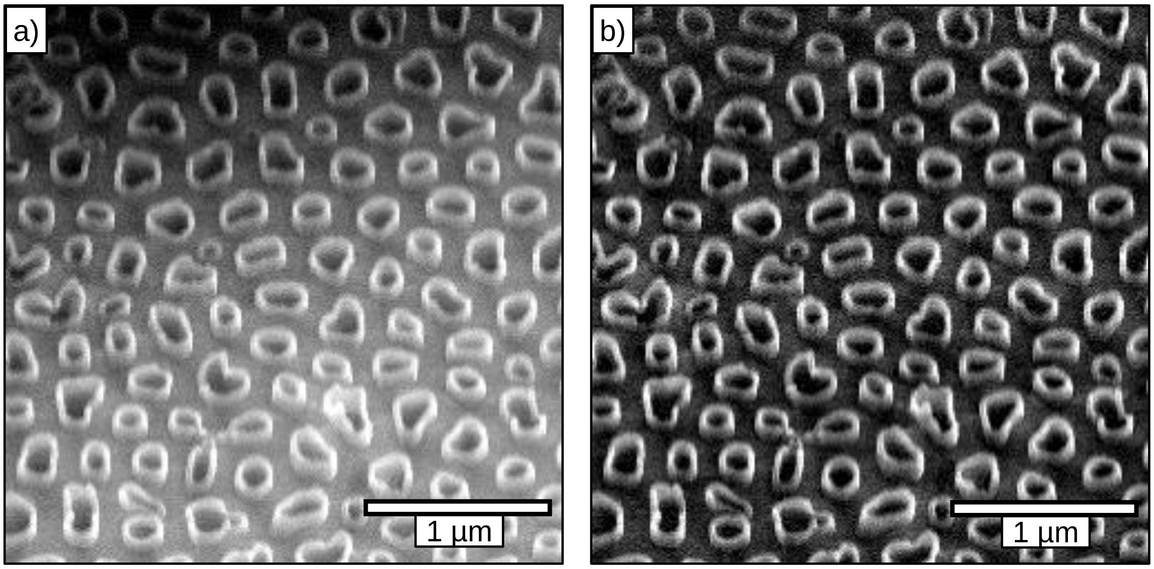


Figure 5.2: Comparison of contrast adjusted CNT array SEM micrograph of a region shadowed by another device before (a) and after (b) Tophat filtering.

The filtering process is constrained such that it does not make an adjustment if it would over-saturate a region of the image as to not lose any geometric information. It is important that d_{st} exceed the size of the individual CNTs by a wide margin otherwise intensity adjustments will change the contrast of geometric features within a single tube. If downstream shape analysis identifies a CNT with $d_{CNT} > d_{st}$ the process starts over and the Tophat filtering is carried out with a structuring element 50% larger than the previous d_{st}. Figure 5.2 illustrates an application of the Tophat filter.

5.3.2 Contrast Thresholding

CNT Arrays have three major defining geometric features: tube walls, pores, and the membrane surface. These features, shown in Figure 5.3 must be discreetly identified. The features are identifiable based on their relative intensity. In general the pores have the lowest intensity while the CNT walls have the highest intensity. The following assumptions are used to perform contrast thresholding:

1. The membrane surface takes up the largest area.

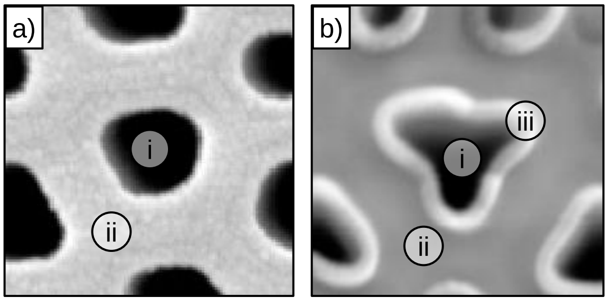


Figure 5.3: Identifiable geometric features of membrane before carbon coating (a) and after coated carbon tubes are exposed (b). The pore (i), the membrane surface (ii) and the CNT walls (iii) are visible and distinct.

2. Pores consist of the lowest intensity.
3. CNT walls consist of the highest intensity.
4. The intensity ranges do not significantly overlap.

Based on these assumptions the histogram can be used to identify geometric features by assigning intensity ranges to them. An intensity range for the pores, CNT walls, and membrane surface are established using the shape of the histogram after Tophat filtering. Figure 5.4 shows a typical histogram of an SEM micrograph of an etched CNT array. The value of h_{max} is identified as the first local minimum after the global maximum. h_{min} is the corresponding value that has the same frequency in the direction of the global maximum from h_{max}. The corresponding ranges are listed in Table 3.4.

For porous membranes before carbon coating, only two ranges are required: pore and membrane surface. The effectiveness of the process is dependent on the assumptions listed above. These assumptions are true for the samples imaged for the NBIL, but adjustments can be made if an assumption is not valid. If Assumption 1 is invalid the global maximum may not correspond to the membrane surface and other means of identifying the membrane region must be used. If Assumption 2 or 3 are invalid the regions outlined in Table 5.1 must be reassigned. If the intensity ranges severely overlap, the regions will not be distinguishable to the system or to visual inspection resulting in false-negatives.

Figure 5.5 illustrates the thresholding process on a well imaged CNT array. Note

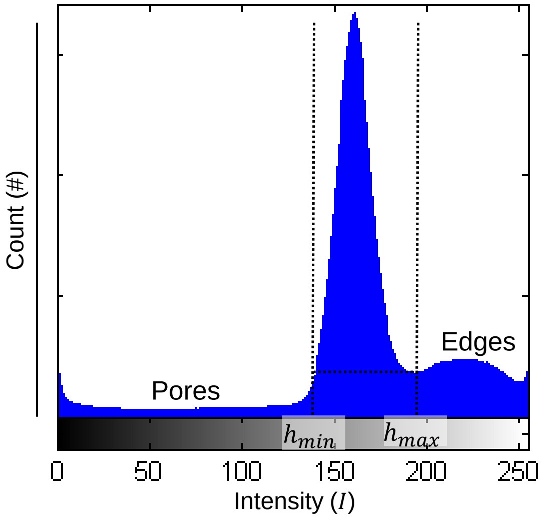


Figure 5.4: Histogram of typical CNT array with h_{min} and h_{max} labeled along with corresponding geometric features represented by each intensity range.

that because CNT walls have a high intensity and pores have a low intensity there is a transition from edge to pore that falls within the threshold range of the membrane surface. This region is accounted for during the shape analysis of individual pores. Morphological closing is used to remove very small regions. Finally, pairs of CNT walls and pores are identified based on their proximity as the final step of pre-processing. These regions potentially contain a CNT and are referred to as candidate regions. They are defined by a bounding box extended by 50 % in each dimension. When analyzing a membrane rather than a CNT array, the bounding boxes only contain a single feature (representing the pore) as there is no CNT.

Table 5.1: Intensity ranges corresponding to geometric features

$I < h_{min}$	Pore
$h_{min} < I < h_{max}$	Membrane Surface
$I > h_{max}$	CNT Edge

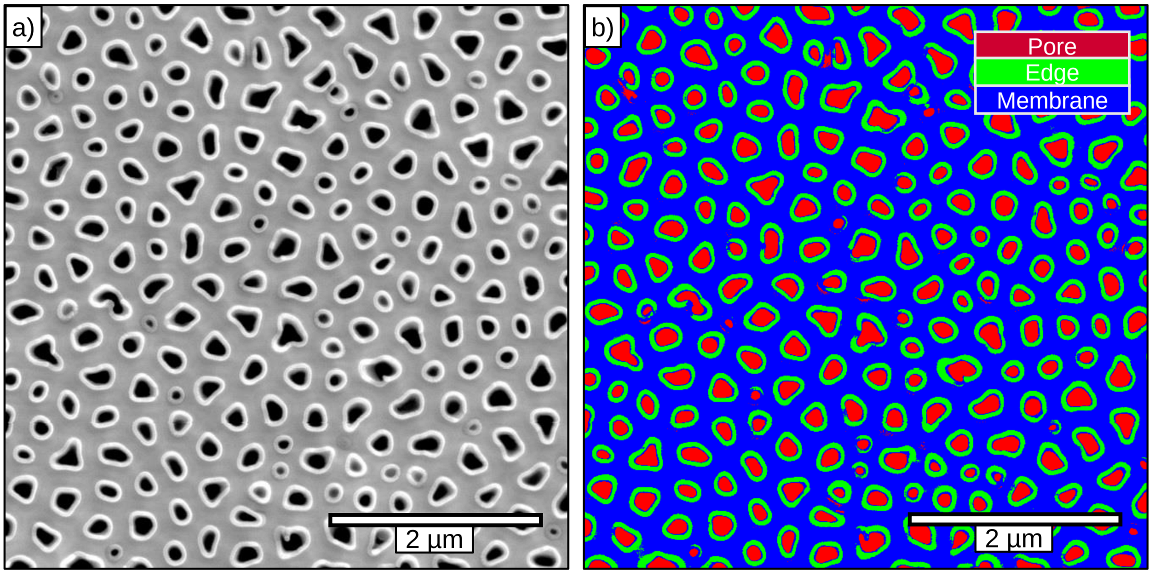


Figure 5.5: Typical SEM micrograph of a CNT array before (a) and after (b) thresholding. Pores are shown in red, CNT walls are shown in green and the membrane surface is shown as blue.

5.4 Processing of Individual Pores

After pre-processing, an array of candidate regions are processed independently. Note that the analysis procedure is similar for a membrane and a CNT array, except all processes on the CNT wall are disregarded when analyzing a bare membrane. Table 5.2 lists the parameters obtained for each pore. Eccentricity is a measure of how elongated the shape is relative to a circle. A best-fitting ellipse is found for the region defined as having the same second moments as the region. The eccentricity is calculated as the ratio of the distance between foci and the major axis length. Most other geometric properties of a shape are trivial to calculate from the data obtained in this process.

5.4.1 Shape Analysis

The determination of whether or not the region represents a pore is based on a solidity threshold (S_{min}), an eccentricity threshold (E_{max}), and an area threshold (A_{min}). If the pore parameters exceed any of these thresholds ($S < S_{min}$, $E > E_{max}$, or $A < A_{min}$) the region is not considered a pore and its parameters are not aggregated. These thresholds can be overwritten if the system is being applied to an abnormally deformed membrane or array. A_{min} is defined dynamically as 5 % of the largest pore region. S_{min} is specified as 0.7 while E_{max} is specified as 0.9. S_{min} and E_{max} in particular eliminate abnormal pores from the data aggregation. If a pore is blocked it will have a very low S and if two pores are connected as a single region it will have a very high E. If the measurement of abnormal pores is desired these thresholds may be adjusted or ignored.

Table 5.2: Pore parameters obtained during individual processing

A	Area	-
P	Perimeter	-
d_h	Hydraulic Diameter	$4A/P$
d_{eq}	Equivalent Diameter	$\sqrt{4A\pi}$
A_c	Convex Area	Area of Convex Hull
S	Solidity	A/A_c
E	Eccentricity	-
t	Wall Thickness	-
h	Wall Height	-

The shape of the CNT wall is also analyzed to determine the CNT thickness (t). The centerline of the CNT wall is found by removing branches from the morphological skeleton of the wall region to form a polyline[2]. t along the skeleton is defined as the width of the CNT region normal to the direction of the skeleton. The direction of the skeleton is calculated as being parallel to the line formed by the points 5 px in each direction for computationally efficient smoothing. The minimum, maximum and average t is recorded for each pore and CNT wall pair. Figure 5.6 shows a pore superimposed with the detected centerline of the CNT edge. If a CNT wall is not fully formed the end-points of the skeleton are connected with a straight line with $t = 0$.

[2] A connected sequence of line segments defined by a series of points.

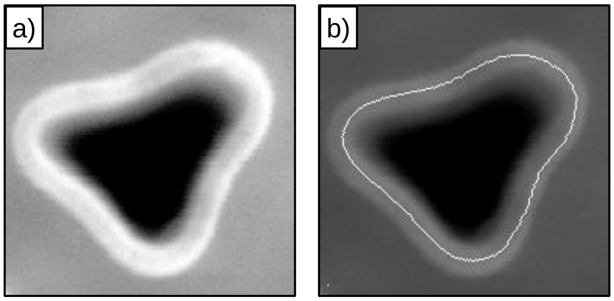


Figure 5.6: An example of CNT wall detection showing the original SEM micrograph (a) and the centerline of the CNT wall (b).

5.4.2 Height Analysis

When observing a sample using microscopy the geometric features are observed in the microscopes xy-plane making measurements in x and y trivial while measurements in z are problematic. The underlying geometric principle governing the height analysis is that when the sample is tilted by θ about y, the displacement of a point along the x-axis can approximate the original z position (z_0). Equation. 5.1 is accurate if the point is directly above the center of rotation. A point's displacement along x from the center of rotation (L) contributes to it's total displacement along x during rotation (Δx_t). If the center of rotation, O, is unknown then L also unknown. The displacement of the membrane surface along x (Δx_e) can be measured and subtracted from Δx_t to obtain Δx as in Equation 5.2. By overlaying the original image and the tilted image, pixels from the CNT wall and membrane surface can be matched to obtain values for Δx_t and Δx_e respectively. The scale-invariant feature transform (SIFT, [70]) algorithm is used to detect local features in both images which are then matched, measuring their displacements relative to the membrane surface along the direction of tilt. Multiple values of z_0 are calculated along the CNT wall and averaged to determine h. The variability and range of z_0 allows irregularly shaped CNTs with varying heights to be characterized if necessary.

$$z_0 = \Delta x * csc(\theta) \tag{5.1}$$

$$z_0 = (\Delta x_t - \Delta x_e) * csc(\theta) \tag{5.2}$$

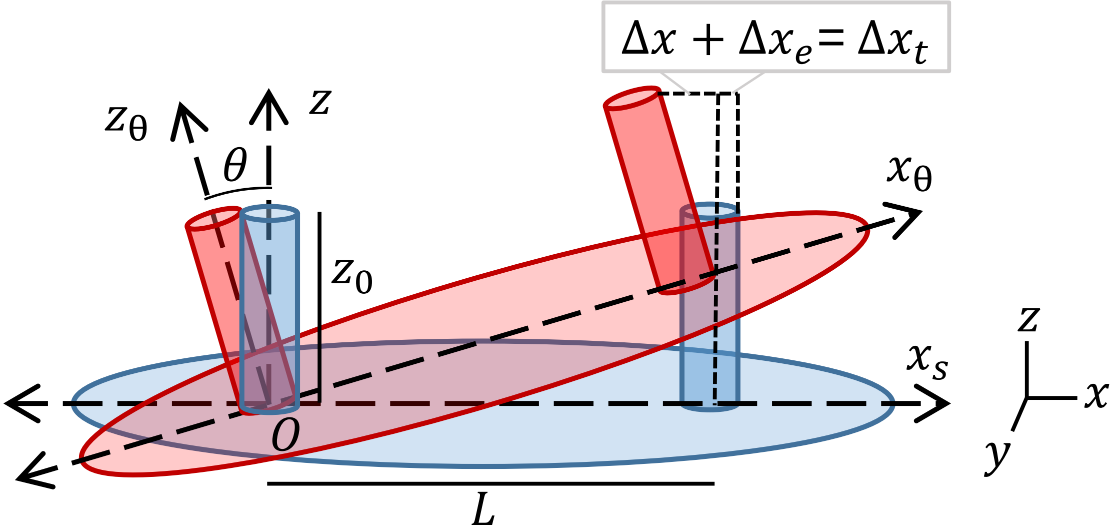


Figure 5.7: Schematic of parameters used in height calculation based on tilt.

5.4.3 Pore Aggregation

After analyzing each region and discarding regions that are not pores the parameters from Table 5.2 are aggregated to obtain histogram data including minimums, maximums, averages, and various percentile ranges and standard deviations. Additional metrics are calculated for the entire visible portion of the CNT array outlined in Table 5.3. Data is encapsulated by pore so multivariate numerical analysis is convenient.

Centroid-to-centroid links are generated by connecting each pore and eliminating intersecting links, resulting in a mesh of neighboring pores. The outermost links are discarded because they would intersect with links to pores that are outside of the visible range. Figure 5.8 shows an example of a mesh produced by this process.

Table 5.3: Aggregated parameters of the visible array.

$\overline{C_s}$	Spacing	Average distance to neighboring pores
ϕ	Porosity	% of surface covered by pores
D	Pore Density	Number of pores per μm^2

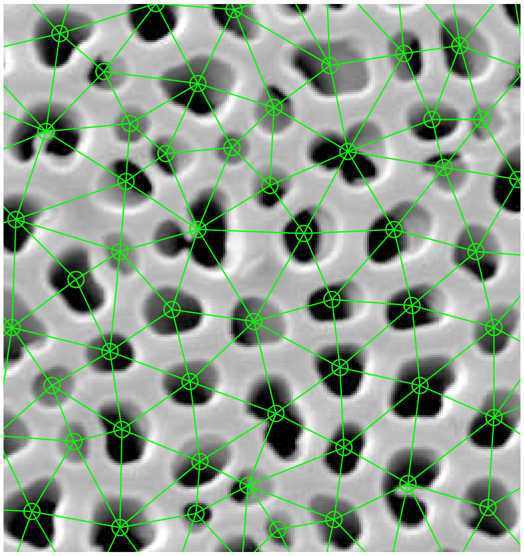


Figure 5.8: Procedurally generated centroid-to-centroid links of a porous AAO membrane.

5.5 Results and Discussion

To validate the system an AAO membrane and a CNT array were measured manually and with our system. The Whatman Anodisc 13 AAO membrane was polished with a 1 µᴍ diamond suspension at 15 N and 150 RPM for 3 min and etched with a 1 ᴍ NaOH solution for 5 min. The CNT array was produced by depositing carbon (CVD) on an Anodisc membrane. After the CVD, the carbon layer on one side of the membrane was removed using oxygen plasma (LAM 490) at 300 mTorr, RF 250 W and oxygen flow rate of 100 sccm for 3.75 min. CNTs were partially exposed by selectively etching the AAO template using a RIE (PlasmaTherm720/740 Etcher). First, the AAO templates were etched with 80 %/20 % mixture of boron trichloride (BCl_3) and argon (Ar) gas, respectively, at 15 mTorr, RF 250 W and 100 sccm of total gas flow rate for 6 hours. The exposed CNTs were trimmed using oxygen plasma at 300 mTorr, RF 250 W and oxygen flow rate of 100 sccm for 3.5 min. Then, the CNTs were exposed again using RIE with 80 %/20 % mixture of BCl_3 and Ar gas, respectively, at 15 mTorr, RF 250 W and 100 sccm of total gas flow for 4 hours. Additionally, renderings of the automated process were generated and checked for false-positives and false-negatives. A region that was identified as pore that should not have been was considered a false-positive. A pore that was not recognized was considered a false-negative. For the automatic and manual measurements if an open region of a membrane appeared to be two connected pores, the region was still considered a single pore.

5.5.1 Validation

Table 5.4: Aggregated comparison of manual measurement and automated measurement of an AAO membrane. Units in px and px^2.

	Automated	Manual	e
A_{mean}	827.2	864.4	4.3 %
A_{max}	1433	1382	3.7 %
σ	252.8	205.4	23.1 %

The results of manual measurements were compared to the results from automated analysis in Table 5.4 for the bare membrane. The hand measurements for area were estimated by measuring the major and minor diameter of the pore and modeling the pore as an ellipse. This method was particularly inaccurate for pores with low solidity. The effectiveness of the analysis was ensured qualitatively by monitoring for false-positives and false-negatives on a sample CNT array. For a field of 355 pores the automated procedure and the careful manual measurements yielded very similar means. However, the σ for each were considerably different. This difference is likely due to the tendency of manual measurements to be biased towards avoiding deformed portions of pores.

Visual inspection showed no false positives and a single false negative. Figure 5.10 shows the false-negative which occurred when two pores were connected by an intended segment of the membrane. The false positive was incorrectly discarded from the aggregated data but only represented 0.28 % of the pores. Figure 5.9 shows a section of the AAO membrane with detected edges superimposed. Visual inspection indicates that the edges are correctly identified within a pixel indicating that the measurement error

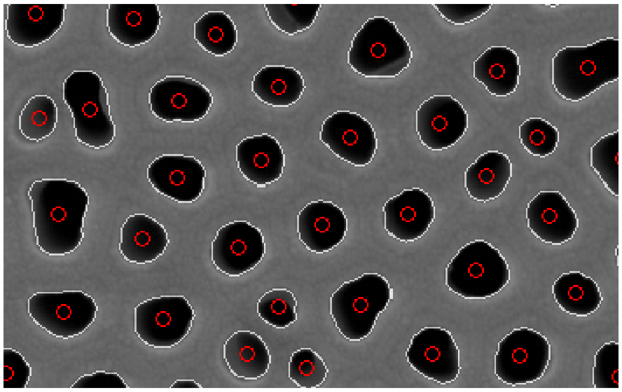


Figure 5.9: Portion of membrane with identified pores outlined in white and centroids labeled in red.

dependent on the resolution of the SEM micrograph.

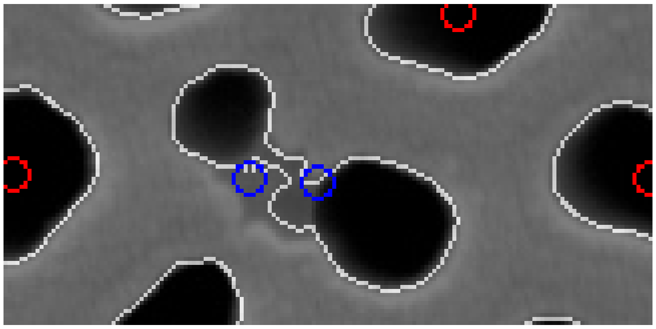


Figure 5.10: False negative occurring when two pores are connected by a shaded area of the membrane.

Analyzing a CNT Array rather than a bare AAO membrane introduces CNT thickness, t, and CNT height, h. Table 5.5 shows these characteristics of the CNT array collected manually and automatically. The measurements for edge thickness are in good agreement, as the edge is very clearly defined. There is less variability for automated measurements of $\bar{t}$ because the t for each pore is an average of every thickness around the edge of the pore rather than a single measurement for each pore.

5.5.2 Lattice Layer Removal

Visual inspection of the Anodisc 13 membranes revealed irregularities in pore shape and distribution, referred to herein as the lattice layer. CNT array transfection studies

Table 5.5: Aggregated comparison of manual measurement ($n = 35$) and automated measurement ($n = 70$) of the CNT Array. Lengths are reported in nm.

	Automated	Manual	e
$\bar{h}$	442.1±104.6	487.4±70.2	9.4%
$\bar{t}$	39.2±5.5	38.6±5.8	1.5%

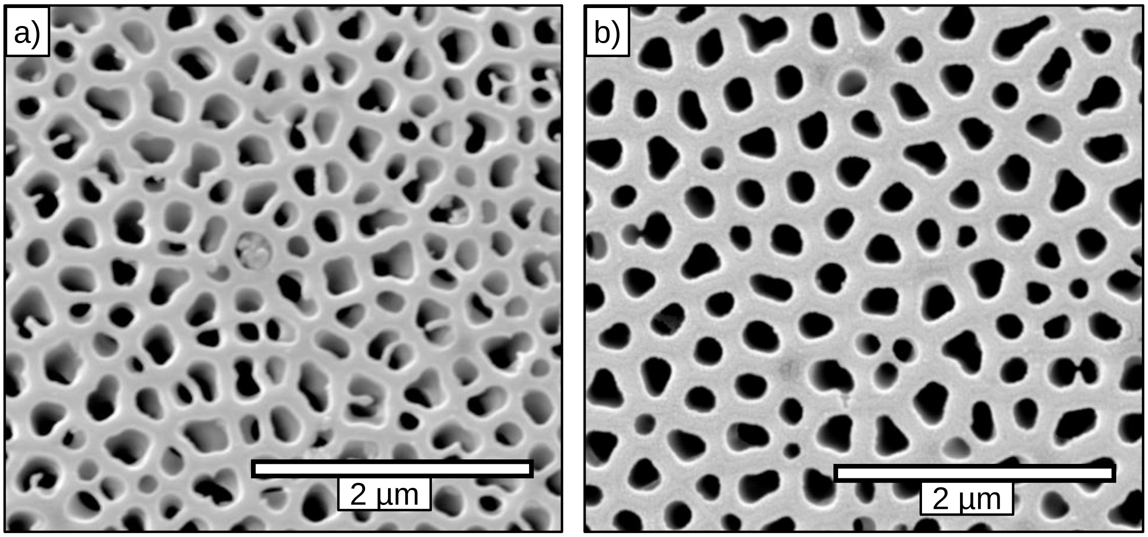


Figure 5.11: Segments of two AAO membranes polished with diamond suspension at 150 RPM. The membranes were polished for 12 min at 15 N (a), and 18 min at 30 N (b).

benefit from uniformly shaped and arranged tubes and therefore the lattice layer is detrimental for our purposes. To remove the lattice layer from the sacrificial AAO membrane, the surface is polished. The polishing force and time will determine if a sufficient amount of the lattice layer is removed resulting in desirable surface geometry. Two membranes were polished at 150 RPM with 1 µM diamond suspension. The first sample, M_A, was polished for 12 min at 15 N and 150 RPM. The second sample, M_B, was polished for 18 min at 12 N and 150 RPM. Both samples were then etched with a 1 M NaOH solution for 5 min. Figure 5.11 shows both polished membranes. Figure 5.11a shows the noticeable lattice layer of membrane S_B though the geometric difference between M_A and M_B is difficult to evaluate without measurement. Both membranes were characterized and the results are shown in Table 5.6 alongside properties of the uniform region of the membrane outlined in [71].

Table 5.6: Surface characteristics of M_A ($n = 406$), M_B ($n = 357$), p of the ANOVA test between M_A and M_B for each measurement, and desired characteristics.

		M_A	M_B	p	Desired
$\overline{A}$	µm^2	0.038 ± 0.013	0.041 ± 0.015	2.82×10^{-4}	72 600
$\overline{d_{eq}}$	nm	214.7 ± 38.5	226.3 ± 43.9	1.26×10^{-4}	304
$\overline{C_s}$	nm	397.9 ± 69.8	359.2 ± 83.7	4.54×10^{-8}	-
$\overline{E}$	-	0.640 ± 0.148	0.607 ± 0.148	1.90×10^{-3}	0
$\overline{S}$	-	0.952 ± 0.028	0.964 ± 0.037	3.29×10^{-7}	1

M_B had much more desirable surface characteristics than M_A. For each ANOVA test, $p < 0.01$ indicating confidence that the means were unique. Figure 5.12 shows the probability density function (pdf) of d_{eq} for both membranes. Figure 5.13 shows similar histograms for E and S. Furthermore, the standard deviation of S was 0.038 for M_A and 0.028 for M_B indicating that the pores were more consistently spaced on M_B.

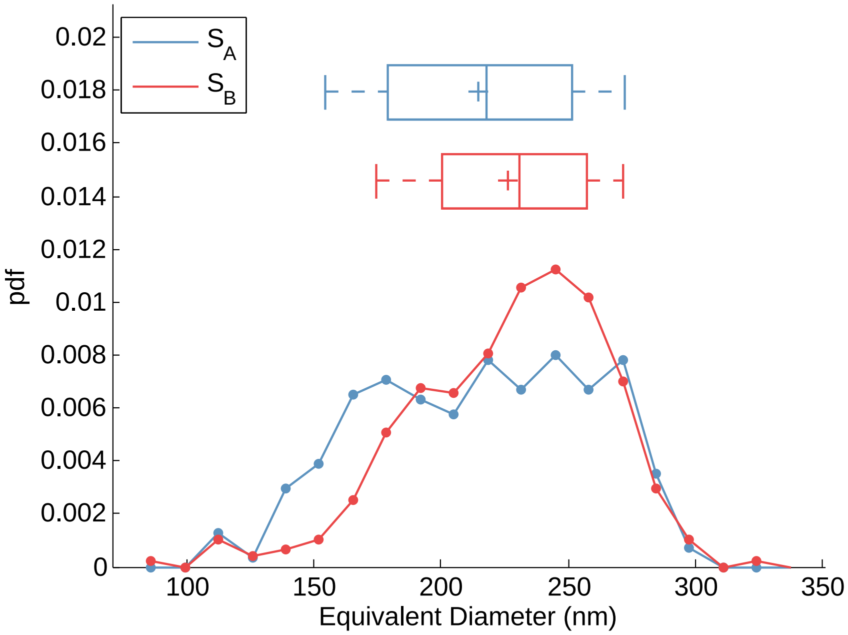
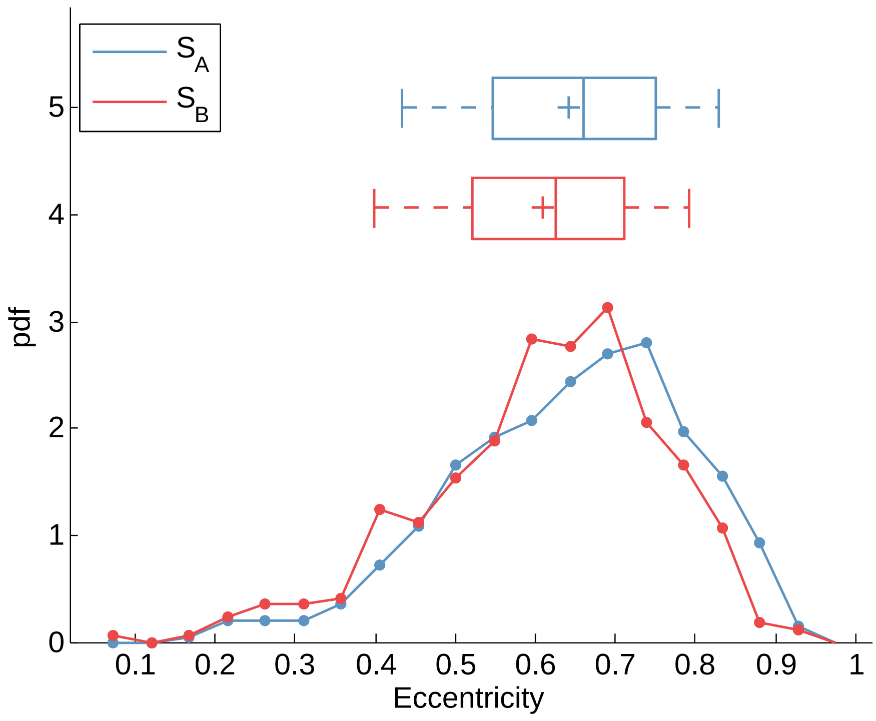
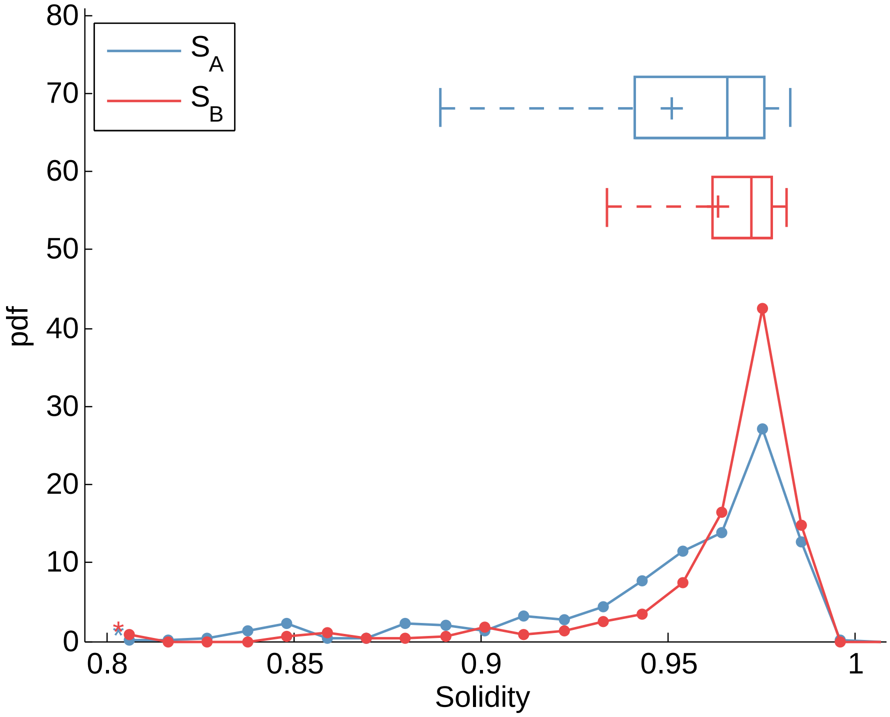


Figure 5.12: Histogram of d_{eq} for M_A and M_B including a boxplot showing the mean (+), median (central dash), 25 % and 75 % quartiles and the box boundaries, and percentiles of 9 % and 91 % represented by whiskers



(a) Histogram of E



(b) Histogram of S

Figure 5.13: Histogram of S for M_A and M_B including a boxplot showing the mean (+), median (central dash), 25 % and 75 % quartiles and the box boundaries, and percentiles of 9 % and 91 % represented by whiskers

Chapter 6

Conclusions

This research culminated in a detailed understanding of long-range remote micromanipulation including alternate hardware implementations, user performance as a function of hardware device and latency, user interface considerations, and computational cost of automation procedures.

User Performance

Proprietary manipulator control devices, including the Eppendorf NK2, are expensive and their effectiveness over alternative hardware devices was not well established. This research demonstrated that for at least single-probe manipulation, a simple computer mouse facilitates more effective performance than the Eppendorf NK2 manipulator. The mouse outperformed the other candidate devices and was the most well received by users qualitatively. The tablet also performed very well and was identified by users as an enjoyable device to use. The velocity joystick performed poorly and was not well received and should not be considered for nano- or micromanipulation. These results indicate that manipulation systems, both remote and local, benefit highly from digitization allowing for the use of additional hardware control devices.

When designing a system for remote manipulation it is critical to understand how the system latency will affect performance. The results of this research indicates that at latencies below 100 ms performance will not be appreciably impacted when controlling a single probe manipulator with traditional hardware control devices. Latencies above

200 ms impede performance so they should be avoided when possible but are still acceptable for research and education. If these moderate latency ranges cannot be avoided, then the use of alternate hardware should be considered, particularly a mouse due to its widespread familiarity and better performance at high latencies. Severe latencies significantly impact task performance and should prompt the implementation of a remote system to be reexamined regardless of hardware device or user proficiency.

When performing tests involving latency it is important to minimize biases from fatigue and developed familiarity with a task or control device. Imposing latencies in a pre-defined order is insufficient in this regard, making it difficult to extract meaningful trends. Latency tests should be designed such that latencies are imposed in a random order and that the same latency is imposed multiple times. Furthermore, unpredictable latencies can cause discomfort and performance degradation relative to similar consistent latency. To address this concern, remote manipulation systems can be implemented with an imposed minimum latency equal to the expected average latency of the system, resulting in less fluctuation in latency perceived by the user.

User Interfaces

Given that the limiting factor in many manipulation tasks is the skill of the user, the development of user-friendly software is an important step in facilitating more efficient manipulation. The GUI and functions described herein allows for users to perform manipulation procedures without needing to develop dexterity with a less intuitive operating method. The GUI corrects for error by detecting the manipulator's end effector using image processing to ensure the end-effector reaches the desired destination with more precision. Further development of the image processing protocols and device interfacing can allow for additional types of procedures, including cell injection and micro-assembly, to be carried out through the GUI.

Currently, users are limited to controlling manipulators using devices supplied by the manufacturers. Computerization of the manipulator control processes facilitates the use of a wide array of alternative controllers. The combination of a keyboard and mouse is familiar to most users and facilitates improved user performance. Additionally, computerization of micro- and nanomanipulators provides the ability to enhance commercial instruments by accommodating third-party devices for users with special needs.

The developed user interfaces are accessible remotely through a number of implementations including two custom packages, Secure Shell, and RDP. The two custom packages were MATLAB based with one relying on JAVA libraries within MATLAB. These implementations were compared and only the MATLAB standalone package and RDP performed at an acceptable framerate.

Automation Processing Time

The GUIs developed in the NBIL facilitate the implementation of automated procedures and analysis based on computer vision. Thanks to our new understanding of the effects of latency, we can identify how computationally expensive these automation tasks can be while still allowing for effective real-time manipulation. Characterization of CNT arrays was used as an example because it had direct application in the NBIL and can be implemented for real-time characterization of SEM based nanomanipulation systems.

A successful method was developed to quickly analyze an AAO membrane or CNT array while aggregating information about the shape and density of it's pores. The results contained acceptably few false-positives and false-negatives and the detected pores closely match the correct regions determined by careful manual analysis. Comparison to hand measurements yielded excellent results and the runtime of the system is fast enough to analyze arrays on the order of seconds. The system also provided a reliable comparison of the geometry between two membranes with different removed lattice layer. While this procedure was designed based on CNT arrays, it could also be applied to any flat porous surface provided that parameters are adjusted when the listed assumptions are not satisfied.

A large portion of the technique is computationally inexpensive enough to run in real time for hundreds of pores, up to and including the aggregation of pore geometry. Height analysis is not possible in real time in systems with a single detector because it relies on parallax. Inter-pore distance requires each centroid to be compared to numerous other centroids and therefore is exponentially more sensitive to the number of pores in the field of view. Additionally, this system will continue to facilitate CNT array development by eliminating one of the bottlenecks of our experiments in the NBIL.

Bibliography

[1] N. C. Hensel, R. M. Dunn, and M. G. Schrlau, “Computerized control system and interface for flexible micromanipulator control,” *Advances in Engineering Software*, vol. 86, pp. 107–114, 2015.

[2] D. J. Cappelleri, M. Fatovic, and U. Shah, “Caging micromanipulation for automated microassembly,” in *Proceedings - IEEE International Conference on Robotics and Automation*, (Shanghai), pp. 3145–3150, IEEE, 2011.

[3] D. J. Cappelleri, M. Fatovic, and Z. Fu, “Caging grasps for micromanipulation & microassembly,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 925–930, IEEE, 2011.

[4] W. Wang, Y. Sun, M. Zhang, R. Anderson, L. Langille, and W. Chan, “A system for high-speed microinjection of adherent cells,” *Review of Scientific Instruments*, vol. 79, no. 10, p. 104302 (6pp), 2008.

[5] K. Kim, X. Liu, Y. Zhang, J. Cheng, X. Y. Wu, and Y. Sun, “Manipulation at the nanonewton level: Micrograpsing for mechanical characterization of biomaterials,” in *Proceedings - IEEE International Conference on Robotics and Automation*, (Kobe), pp. 902–907, IEEE, 2009.

[6] W. Wang, X. Liu, D. Gelinas, B. Ciruna, and Y. Sun, “A fully automated robotic system for microinjection of zebrafish embryos,” *PLoS ONE*, vol. 2, no. 9, p. e862, 2007.

[7] W. H. Wang, X. Y. Liu, and Y. Sun, “Autonomous Zebrafish Embryo Injection Using a Microrobotic System,” in *Control*, pp. 363–368, IEEE, 2007.

[8] W. H. Wang, D. Hewett, C. E. Hann, and X. Q. Chen, “Machine Vision and Image Processing for Automated Cell Injection,” in *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, (Canterbury), pp. 309–314, IEEE, 2008.

[9] W. H. Wang, X. Y. Liu, and Y. Sun, “High-throughput automated injection of individual biological cells,” *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 2, pp. 209–219, 2009.

[10] L. S. Mattos, E. Grant, R. Thresher, and K. Kluckman, “Blastocyst microinjection automation,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 822–831, 2009.

[11] L. Mattos, E. Grant, and R. Thresher, “Speeding up video processing for blastocyst microinjection,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5825–5830, 2006.

[12] G. Becattini, L. S. Mattos, and D. G. Caldwell, “A novel framework for automated targeting of unstained living cells in bright field microscopy,” in *Proceedings - International Symposium on Biomedical Imaging*, pp. 195–198, 2011.

[13] L. S. Mattos and D. G. Caldwell, "A fast and precise micropipette positioning system based on continuous camera-robot recalibration and visual servoing," in *2009 IEEE International Conference on Automation Science and Engineering, CASE 2009*, pp. 609–614, 2009.

[14] G. Becattini, L. S. Mattos, and D. G. Caldwell, "A fully automated system for adherent cells microinjection," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 83–93, 2014.

[15] M. G. Schrlau, E. Brailoiu, S. Patel, Y. Gogotsi, N. J. Dun, and H. H. Bau, "Carbon nanopipettes characterize calcium release pathways in breast cancer cells," *Nanotechnology*, vol. 19, p. 325102, aug 2008.

[16] M. G. Schrlau, E. M. Falls, B. L. Ziober, and H. H. Bau, "Carbon nanopipettes for cell probes and intracellular injection.," *Nanotechnology*, vol. 19, no. 1, p. 015101, 2008.

[17] M. G. Schrlau, N. J. Dun, and H. H. Bau, "Cell electrophysiology with carbon nanopipettes," *ACS Nano*, vol. 3, no. 3, pp. 563–568, 2009.

[18] M. G. Schrlau and H. H. Bau, "Carbon-based nanoprobes for cell biology," *Microfluidics and Nanofluidics*, vol. 7, no. 4, pp. 439–450, 2009.

[19] R. Singhal, Z. Orynbayeva, R. V. Kalyana Sundaram, J. J. Niu, S. Bhattacharyya, E. a. Vitol, M. G. Schrlau, E. S. Papazoglou, G. Friedman, and Y. Gogotsi, "Multifunctional carbon-nanotube cellular endoscopes.," *Nature nanotechnology*, vol. 6, no. 1, pp. 57–64, 2011.

[20] J. J. Niu, M. G. Schrlau, G. Friedman, and Y. Gogotsi, "Carbon nanotube-tipped endoscope for in situ intracellular surface-enhanced raman spectroscopy," *Small*, vol. 7, no. 4, pp. 540–545, 2011.

[21] Z. Orynbayeva, R. Singhal, E. A. Vitol, M. G. Schrlau, E. Papazoglou, G. Friedman, and Y. Gogotsi, "Physiological validation of cell health upon probing with carbon nanotube endoscope and its benefit for single-cell interrogation," *Nanomedicine: Nanotechnology, Biology, and Medicine*, vol. 8, no. 5, pp. 590–598, 2012.

[22] W. Ahmed, M. J. Jackson, and I. Ul Hassan, "Nanotechnology to Nanomanufacturing," in *Emerging Nanotechnologies for Manufacturing*, pp. 1–15, Elsevier, 2010.

[23] E. Du, H. Cui, and Z. Zhu, "Review of nanomanipulators for nanomanufacturing," *International Journal of Nanomanufacturing*, vol. 1, pp. 83–104, nov 2006.

[24] A. Bolopion, C. Dahmen, C. Stolle, S. Haliyo, S. Régnier, and S. Fatikow, "Vision based Haptic Feedback for Remote Micromanipulation in-SEM Environment," *International Journal of Optomechatronics*, vol. 6, no. 3, pp. 236–252, 2012.

[25] A. Bolopion, C. Stolle, R. Tunnell, S. Haliyo, S. Régnier, and S. Fatikow, "Remote microscale teleoperation through virtual reality and haptic feedback," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 894–900, IEEE, sep 2011.

[26] R. M. Dunn and M. G. Schrlau, "Towards an Intuitive and Remotely Accessible Control System for Commercial Nanomanipulators: American Society for Engineering Education," in *2015 ASEE Annual Conference and Exposition*, (Seattle, WA), ASEE, 2015.

[27] A. Bolopion and S. Régnier, “A review of haptic feedback teleoperation systems for micromanipulation and microassembly,” *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 496–502, 2013.

[28] K. Go, K. Kashiwagi, Y. Ito, Y. Nakazawa, and J. Arata, “Eye, robot: A network control system for ophthalmologic examination,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5068 LNCS of *Lecture Notes in Computer Science*, pp. 48–57, Springer Berlin Heidelberg, 2008.

[29] A. J. Szameitat, J. Rummel, D. P. Szameitat, and A. Sterr, “Behavioral and emotional consequences of brief delays in human-computer interaction,” *International Journal of Human Computer Studies*, vol. 67, no. 7, pp. 561–570, 2009.

[30] A. Kawamura, K. Tahara, R. Kurazume, and T. Hasegawa, “Robust visual servoing for object manipulation with large time-delays of visual information,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 4797–4803, 2012.

[31] J. Bohren, C. Papazov, D. Burschka, K. Krieger, S. Parusel, S. Haddadin, W. L. Shepherdson, G. D. Hager, and L. L. Whitcomb, “A pilot study in vision-based augmented telemanipulation for remote assembly over high-latency networks,” in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3631–3638, IEEE, may 2013.

[32] E. Hiemstra, E. M. Terveer, M. K. Chmarra, J. Dankelman, and F. W. Jansen, “Virtual reality in laparoscopic skills training: is haptic feedback replaceable?,” *Minimally invasive therapy & allied technologies : MITAT : official journal of the Society for Minimally Invasive Therapy*, vol. 20, no. 3, pp. 179–184, 2011.

[33] V. Vitiello, S. L. Lee, T. P. Cundy, and G. Z. Yang, “Emerging robotic platforms for minimally invasive surgery,” *IEEE Reviews in Biomedical Engineering*, vol. 6, no. 1, pp. 111–126, 2013.

[34] A. Gudeloglu, J. V. Brahmbhatt, and S. J. Parekattil, “Robotic-assisted microsurgery for an elective microsurgical practice,” in *Seminars in Plastic Surgery*, vol. 28, pp. 11–19, 2014.

[35] R. Wirz, L. G. Torres, P. J. Swaney, H. Gilbert, R. Alterovitz, R. J. Webster, K. D. Weaver, and P. T. Russell, “An experimental feasibility study on robotic endonasal telesurgery,” *Neurosurgery*, vol. 76, no. 4, pp. 479–484, 2015.

[36] A. Degani, H. Choset, A. Wolf, and M. A. Zenati, “Highly articulated robotic probe for minimally invasive surgery,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, pp. 4167–4172, 2006.

[37] G. H. Ballantyne, “Robotic surgery, telerobotic surgery, telepresence, and telementoring: Review of early clinical results,” *Surgical Endoscopy and Other Interventional Techniques*, vol. 16, pp. 1389–1402, oct 2002.

[38] S. Xu, M. Perez, K. Yang, C. Perrenot, J. Felblinger, and J. Hubert, “Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-Trainer simulator,” *Surgical Endoscopy and Other Interventional Techniques*, vol. 28, no. 9, pp. 2569–2576, 2014.

[39] S. Xu, M. Perez, K. Yang, C. Perrenot, J. Felblinger, and J. Hubert, “Effect of latency training on surgical performance in simulated robotic telesurgery procedures,” *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 11, no. 3, pp. 290–295, 2015.

[40] T. Nakamura, K. Kita, R. Kato, K. Matsushita, and Y. Hiroshi, “Control Strategy for a myoelectric hand: Measuring acceptable time delay in human intention discrimination,” *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, vol. 2009, pp. 5044–5047, 2009.

[41] E. D. Zamani, G. M. Giaglis, and A. Pouloudi, “Tactile Experiences: User Interpretations and Meaning with Tablets,” in *2014 47th Hawaii International Conference on System Sciences*, vol. 47, (Waikoloa, HI), pp. 3337–3346, IEEE, jan 2014.

[42] C. A. Romney, “Tablet PC use in freshman mathematics classes promotes STEM retention,” in *Proceedings - Frontiers in Education Conference, FIE*, (Rapid City, SD), pp. 1–7, IEEE, 2011.

[43] S. Oviatt, A. Arthur, and J. Cohen, “Quiet interfaces that help students think,” in *Proceedings of the 19th annual ACM symposium on User interface software and technology UIST 06*, (N. Y., N.Y.), p. 191, ACM, 2006.

[44] K. Kotani and K. Horii, “An Analysis of Muscular Load and Performance in Using a Pen-tablet System,” *J Physiol Anthropol*, vol. 22, no. 2, pp. 89–95, 2003.

[45] M. K. O’Malley, O. Celik, J. C. Huegel, M. D. Byrne, J. Bismuth, B. J. Dunkin, A. C. Goh, and B. J. Miles, *Training and performance analysis: Robotics as a tool for training and assessment of surgical skill*. Online: Springer New York, 2014.

[46] A. Komlodi, E. Jozsa, K. Hercegfi, S. Kucsora, and D. Borics, “Empirical usability evaluation of the Wii controller as an input device for the VirCA immersive virtual space,” in *2011 2nd International Conference on Cognitive Infocommunications (CogInfoCom)*, (Budapest), pp. 1–6, 2011.

[47] G. Dagnino, L. S. Mattos, G. Becattini, M. Dellepiane, and D. G. Caldwell, “Comparative evaluation of user interfaces for robot-assisted laser phonomicrosurgery,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 7376–7379, IEEE, aug 2011.

[48] J. J. Shaughnessy, E. B. Zechmeister, and J. S. Zechmeister, *Research Methods in Psychology*. 2012.

[49] X. Ma and S. Guo, “Dynamic performance analysis of a robotic catheter manipulating system,” in *2013 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2013* (IEEE, ed.), (Takamatsu, Japan), pp. 779–784, IEEE, aug 2013.

[50] D. N. Kim, K. Kim, K. Y. Kim, and S. M. Cha, “Dexterous teleoperation for micro parts handling based on haptic/Nisual interface,” in *MHS 2001 - Proceedings of 2001 International Symposium on Micromechatronics and Human Science*, pp. 211–217, IEEE, 2001.

[51] A. Adamoli, M. Jovic, and M. Hauswirth, “LagAlyzer: A latency profile analysis and visualization tool,” in *2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS)*, pp. 13–22, 2010.

[52] Y. Endo, Z. Wang, J. B. Chen, and M. I. Seltzer, "Using Latency to Evaluate Interactive System Performance.," in *OSDI '96 Proceedings of the second USENIX symposium on Operating systems design and implementation*, vol. 30, pp. 185–199, ACM, 1996.

[53] P. Cardoso-Leite and A. Gorea, "Comparison of perceptual and motor decisions via confidence judgments and saccade curvature.," *Journal of neurophysiology*, vol. 101, pp. 2822–36, jan 2009.

[54] M. Golshadi and M. G. Schrlau, "Template-based synthesis of aligned carbon nanotube arrays for microfluidic and nanofluidic applications," *ECS Transactions*, vol. 50, pp. 1–14, apr 2013.

[55] T. H. Nam, K. Goto, Y. Yamaguchi, E. Premalal, Y. Shimamura, Y. Inoue, K. Naito, and S. Ogihara, "Effects of CNT diameter on mechanical properties of aligned CNT sheets and composites," *Composites Part A: Applied Science and Manufacturing*, vol. 76, no. January, pp. 289–298, 2015.

[56] Q. Cao, S.-j. Han, G. S. Tulevski, Y. Zhu, D. D. Lu, and W. Haensch, "Arrays of single-walled carbon nanotubes with full surface coverage for high-performance electronics," *Nature Nanotechnology*, vol. 8, no. 3, pp. 180–186, 2013.

[57] S. Park, Y. S. Kim, W. B. Kim, and S. Jon, "Carbon nanosyringe array as a platform for intracellular delivery," *Nano Letters*, vol. 9, no. 4, pp. 1325–1329, 2009.

[58] A. K. Shalek, J. T. Robinson, E. S. Karp, J. S. Lee, D.-R. Ahn, M.-H. Yoon, A. Sutton, M. Jorgolli, R. S. Gertner, T. S. Gujral, G. MacBeath, E. G. Yang, and H. Park, "Vertical silicon nanowires as a universal platform for delivering biomolecules into living cells.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 5, pp. 1870–5, 2010.

[59] K. Miyoshi, "Surface Characterization Techniques: An Overview," *Mechanical tribology: materials, characterization, and applications. Part*, vol. 2, pp. 1–46, 2002.

[60] J.-h. Han, S. Hong, T. Young, J.-b. Yoo, C.-y. Park, and H. Jin, "Effects of growth parameters on the selective area growth of carbon nanotubes," *Thin Solid Films*, vol. 409, no. 1, pp. 126–132, 2002.

[61] W. Cho, M. Schulz, and V. Shanov, "Growth and characterization of vertically aligned centimeter long CNT arrays," *Carbon*, vol. 72, pp. 264–273, 2014.

[62] Y. Man, Z. Chen, Y. Zhang, and P. Guo, "Patterned growth of vertically aligned carbon nanotube arrays using colloidal lithography and plasma enhanced chemical vapor deposition," *Journal of Alloys and Compounds*, vol. 650, pp. 86–91, 2015.

[63] G. Lalwani, A. T. Kwaczala, S. Kanakia, S. C. Patel, S. Judex, and B. Sitharaman, "Fabrication and characterization of three-dimensional macroscopic all-carbon scaffolds," *Carbon*, vol. 53, pp. 90–100, 2013.

[64] K. E. Mochalov, A. E. Efimov, V. A. Oleinikov, and I. Nabiev, "High-resolution Scanning Near-field Optical Nanotomography: A Technique for 3D Multimodal Nanoscale Characterization of Nano-biophotonic Materials," *Physics Procedia*, vol. 73, pp. 168–172, 2015.

[65] C. Li, Z. Liu, H. Xie, and D. Wu, “Novel 3D SEM Moire method for micro height measurement.,” *Optics express*, vol. 21, no. 13, pp. 15734–15746, 2013.

[66] C. Li, Z. Liu, and H. Xie, “Novel scanning electron microscope bulge test technique integrated with loading function,” *Review of Scientific Instruments*, vol. 85, no. 10, p. 103709, 2014.

[67] A. P. Tafti, A. B. Kirkpatrick, Z. Alavi, H. A. Owen, and Z. Yu, “Recent advances in 3D SEM surface reconstruction,” *Micron*, vol. 78, pp. 54–66, 2015.

[68] A. P. Tafti, J. D. Holz, A. Baghaie, H. A. Owen, M. M. He, and Z. Yu, “3DSEM++: adaptive and intelligent 3D SEM surface reconstruction,” *Micron*, 2016.

[69] E. R. Dougherty, *An introduction to morphological image processing*, vol. 9. SPIE Optical Engineering Press, 1992.

[70] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, IEEE, 1999.

[71] M. Golshadi, J. Maita, D. Lanza, M. Zeiger, V. Presser, and M. G. Schrlau, “Effects of synthesis parameters on carbon nanotubes manufactured by template-based chemical vapor deposition,” *Carbon*, vol. 80, no. 1, pp. 28–39, 2014.

Appendix A

Selected MATLAB Functions

A.1 Main CNT Array Analysis Function

All MATLAB functions and scripts used in this research are available in supplemental material. The MATLAB functions included in this appendix are high-level functions from which the other functions are invoked.

```
% Image processing and measurements for CNT Arrays and AAO membranes
%
% 'params' input has defaults when fields are not specified:
%                 image_name: 'sample.jpg'
%                area_thresh: 100
%         comprehensive_flag: false
%           bw_thresh_offset: 0.0
%          clogged_threshold: 0.2
%                tilted_name: 'sample.jpg'
%             area_threshold: 100
%         solidity_threshold: 0.7
%     eccentricity_threshold: 0.9
%             clogged_thresh: 0.2
%
% 'data' stucture contains fields:
%   PercentOpen
%   pore_flag
%   Area
%   Perimeter
%   HydraulicDia
%   Eccentricity
%   Solidity
%   xCentroid
%   yCentroid
%   EqDia
%   EdgeWidth (optional)
%
% 'image' structure contains:
%   original
%   w
%   h
%   open_region
%   gray
%   tophat
%   bw
%   edges
%
% 'cell_array' structure contains:
%   data        measurements in column form
%   headers     corresponding headers for data
%
% Ryan Dunn
% Nano Bio Interface Lab
% rmd8337@rit.edu

function [output,params] = poreAnalysis(params)

%% Handling Input Cases

% Warn user if no inputs are provided
if nargin == 0
    warning('No input was provided, default values will be used')
    params=[];
end

% Inputs and default values
input_defaults = {
    'image_name',               'newtest.tif'   ;...
    'tilted_name',              'none'          ;...
    'area_threshold',           100     ;...
    'solidity_threshold',       .7      ;...
    'eccentricity_threshold',   .9      ;...
    'bw_thresh_offset',         -.1     ;...
    'clogged_thresh'            .2      ;...
    'tilt_deg'                  -1      ;...
    'clogged_flag',             false   ;...
    'edge_flag'                 false   ;...
    'height_flag'               false   ;...
    'keep_non_pores'            false   ;
    };

% Populate all default values
for j = 1:length(input_defaults)
    if ~isfield(params,input_defaults{j,1})
        fieldname = input_defaults{j,1};
        params.(fieldname) = input_defaults{j,2};
    end
end

% Just some warnings
if ~strcmp(params.tilted_name,'none') && params.tilt_deg == -1
    warning('No tilt angle was specified, set input.tilt_deg to the angle of tilt in the tilted image')
end

% Plot options for testing only
plot_flag = false;

%% Image handling

% Read Image
image.original = imread(params.image_name);

% Image Properties
dimensions = size(image.original);
image.w = dimensions(1);
image.h = dimensions(2);

% Data structures
box_extension = [-20 -20 40 40];
data = []; % Initialize as double
clogged_thresh = .2; % BW threshold for pore clogging measurements
image.open_region = zeros(image.w,image.h);


%% Pre-processing

% Checks for transparency layers and converts to grayscale
if ~ismatrix(image.original)
    image.gray = image.original(:,:,1);
else
    image.gray = image.original;
end

% Do the same for the tilted image
```

```
if ~strcmp(params.tilted_name,'none')
    image.tilted = imread(params.tilted_name);
    if ~ismatrix(image.tilted);
        image.tilted = image.tilted(:,:,1);
    end
end

% Top Hat filter and contrast adjustment
image.tophat = imadjust(imtophat(image.gray,strel('disk',12)));
hist_data = imhist(image.gray);
[~,hist_max_index] = max(hist_data(50:250));

% Generate Black and White Image
image.bw = im2bw(image.gray,hist_max_index/255-params.bw_thresh_offset);

% Optional morphological transformations to account for noise or abnormalities
% image.bw = imdilate(image.bw,strel('disk',2));
image.bw = imclose(image.bw,strel('disk',2));
% image.bw = imclearborder(i);
% image.bw = imopen(image.bw,strel('disk',1));
% imshow(image.bw);

% Identify Morphological edges, these are only used for plotting
image.edges = edge(image.bw);

% Region detection
props  = regionPropertyAnalysis(~image.bw, 'all');
centroids = cat(1, props.Centroid);

for i = 1:length(centroids)
    pore_flag = true;

    if min(props(i).BoundingBox)<1
        pore_flag = false;
    end

    % Check if region borders an edge of the image
    if max(props(i).BoundingBox(1)+props(i).BoundingBox(3))>image.h-2 || max(
       props(i).BoundingBox(2)+props(i).BoundingBox(4))>image.w-2
        pore_flag = false;
        EdgeContactFlag = true;
    else
        EdgeContactFlag = false;
    end

    % Analyze region based on thresholds
    if props(i).Solidity < params.solidity_threshold
        pore_flag = false;
    end
    if props(i).Eccentricity > params.eccentricity_threshold
        pore_flag = false;
    end
    % Check if pore is too small to consider
    if props(i).Area< params.area_threshold
        pore_flag = false;
    end

    %% Data processing
    props(i).HydraulicDia = 4*props(i).Area/props(i).Perimeter;

    % Edge Analysis
    if (params.edge_flag == true || params.height_flag == true) && pore_flag ==
        true
        % Extend the bounding box of the pore for individual analysis
        bb = props(i).BoundingBox;
        bbe = [bb(1)-bb(3), bb(2)-bb(4), bb(1)+bb(3)*2, bb(2)+bb(4)*2];
        bbe = int16(bbe);
        % check bounds
        if bbe(4)<image.w && bbe(3)<image.h && min(bbe) > 1
            if params.edge_flag == true
                img = image.gray(bbe(2):bbe(4),bbe(1):bbe(3));
                [data(i).EdgeWidth,~] = edgeMeasure(img,150/255,hist_max_index
                    /255-params.bw_thresh_offset);
            end
            if params.height_flag == true
                img1 = image.gray(bbe(2):bbe(4),bbe(1):bbe(3));
                img2 = image.tilted(bbe(2):bbe(4),bbe(1):bbe(3));
                [data(i).EdgeHeight] = heightMeasure_bw(img1,img2,params.
                    tilt_deg);
            end
        end
    end

    % Percent Clogged Analysis
    if params.clogged_flag == true
        % Call function to measure how clogged each pore is
        [props(i).PercentOpen,props(i).clogged] = pore_percent_open(image.gray,
            props(i).ConvexHull,params.clogged_thresh);
        if pore_flag == true
            image.open_region = image.open_region+props(i).clogged;
        end
        data(i).PercentOpen = props(i).PercentOpen;
    end



    %% Aggregate data
    data(i).pore_flag = pore_flag;
    data(i).Area = props(i).Area;
    data(i).Perimeter = props(i).Perimeter;
    data(i).HydraulicDia = props(i).HydraulicDia;
    data(i).Eccentricity = props(i).Eccentricity;
    data(i).Solidity = props(i).Solidity;
    data(i).xCentroid = props(i).Centroid(1);
    data(i).yCentroid = props(i).Centroid(2);
    data(i).EqDia = props(i).EquivDiameter;
    data(i).EdgeContact = EdgeContactFlag;
end

% Plotting (for testing)
if plot_flag == true()
    c = imfuse(image.original,image.edges,'blend');
    imshow(c)
    hold on
    plot(centroids(:,1),centroids(:,2), 'ro');

    for i = 1:length(centroids)
        if data(i,1) == true()
            %props(i).intensity = imcrop(image.gray,props(i).BoundingBox+
                box_boundaries);
            %props(i).tilted = imcrop(image.gray_tilt,props(i).BoundingBox+
                box_boundaries);
            plot(props(i).Centroid(1),props(i).Centroid(2), 'go')
            %props(i).heights = rectify(props(i).intensity,props(i).tilted,10,1)
                ;
            text(props(i).Centroid(1),props(i).Centroid(2),num2str(props(i).Area
                ),'Color','w');
        end
    end
end

% Consolidate data
output.data = data;
output.image = image;

% Discard non-pore regions for "cleaner" output
if params.keep_non_pores == false
    output.data([output.data.pore_flag] == 0) = [];
```

```
end

% Optional Plotting, this is usually invoked outside of this function
if false
    label = 'Area';
    plotPores(output,label)
end
end
```

A.2 Minimal GUI Driving Functions

```
function minimalGUI()

    fps = 30;

    % Figure Parameters
    z = 360;    % Figure Size
    l = 100;    % Figure Location
    b = 20;     % Figure Border

    % Co-ordinate systems
    cursor = [0 0]; % Cursor Location
    org = [0 0 0];  % Eppendorf Origin location on figure

    eppScale = [1 1];
    eppVel = 1000;
    scrollMag = 10;


    %% Connect to camera
    imaqreset;
    objects = imaqfind;
    delete(objects);
    vid = videoinput('pointgrey', 1, 'F7_Mono16_648x482_Mode1');
    triggerconfig(vid, 'manual');
    start(vid);
    sample = getsnapshot(vid);
    [yres,xres] = size(sample);
    %z = yres;
    % Screen Resolution
    res = get(0,'screensize');

    %% Figure Setup
    f = figure('Visible','on','Position',[res(3)/2-xres/2,res(4)/2-yres/2,xres
        +2*b,yres+2*b],...
        'name','Path Test','NumberTitle','off');
    a = axes('Units','Pixels','Position',[b,b,xres,yres], ...
        'XLim',[0,xres],'YLim',[0,yres]);
    hold on;
    imgHandle = imshow(sample,'Parent',a);

    set(a,'XTick',[]);
    set(a,'YTick',[]);

    % Plot Cursor
    cursor_marker = plot(0,0,'blacko');


    %% Initialize manipulator connection
    if ~isempty(instrfindall)
        fclose(instrfindall);
    end
    s = SerialManip('COM1','Eppendorf');
    connected = s.connect();
    if connected == true
        set(f,'CloseRequestFcn',@window_shutdown_callback); % set callbacks to
            release manipulator control
        set(imgHandle,'ButtonDownFcn',@click_callback);
        set(f,'WindowScrollWheelFcn',@scroll_callback);
        s.getControl();
    else
        error('Not Connected')
    end

    %% Callbacks and Functions

    function left_click()
        loc = get(a, 'CurrentPoint');
        cursor = loc(1,1:2);
        move_manipulator();
    end

    function move_manipulator()
        offset.x= (cursor(1) - org(1))*eppScale(1);
        offset.y= (cursor(2) - org(2))*eppScale(2);
        pos = [0,-offset.y,-offset.x];
        vel = [eppVel,eppVel,eppVel];
        s.moveStepsCommand(pos,vel);
    end

    function calibrate()
        s.resetCoords();
        org = get(a, 'CurrentPoint');
        loc = get(a, 'CurrentPoint');
        cursor = loc(1,1:2);
    end

    function update_image()
```

```
        newImage = getsnapshot(vid);
        set(imgHandle,'CData',newImage);
        set(cursor_marker,'XData',cursor(1),'YData',cursor(2))
    end

    function click_callback(obj,event)
        click_type = get(f,'SelectionType');
        if strcmp(click_type,'normal')
            left_click();
        elseif strcmp(click_type,'alt')
            calibrate();
        end
    end

    function scroll_callback(obj,event)
        mag = event.VerticalScrollCount*scrollMag;
        c = s.getCoords();
        s.moveCommand(c+[mag 0 0],[eppVel eppVel eppVel])
    end

    function window_shutdown_callback(obj,event)
        s.releaseControl();
        delete(f)
    end

    %% Loop
    pause(.5);
    while 1
        update_image();
    end
end
```