7

Automated Remote Microscope for Inspection of Integrated Circuits

by

Somsak Kittipiyakul

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Electrical Engineering

and Master of Engineering in Electrical Engineering and Computer Science

at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 6, 1996

Author ____________
Department of Electrical Engineering and Computer Science
September 6, 1996

Certified by ____________
Donald E. Troxel
Professor of Electrical Engineering
Thesis Supervisor

Certified by ____________
Frederic R. Morgenthaler
Chairman, Department Committee on Graduate Students

Automated Remote Microscope for Inspection of Integrated Circuits

by

Somsak Kittipiyakul

Submitted to the
Department of Electrical Engineering and Computer Science

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Electrical Engineering
and Master of Engineering in Electrical Engineering and Computer Science

Abstract

The MIT remote microscope system being developed as part of the Computer Aided Prototyping of Advanced Microsystems project has demonstrated the feasibility of a remote microscope which operates over the internet and its effectiveness to inspect integrated circuits during remote fabrication. It also helps to provide more collaboration and sharing of resources among researchers in the field of semiconductors. However, because the first generation remote microscope was controlled manually by an attendant, an automation of the remote microscope provides a full computer control and a better precision, convenience, and computer-based analysis of images. In addition, the automated remote microscope will be able to support new valuable functionalities that require exact and automatic positioning of the microscope. Thus, the automation of the remote microscope raises the value and effectiveness of the initial remote microscope as a remote inspection tool and collaboration tool during remote or even local semiconductor fabrication.

Thesis Supervisor: Donald E. Troxel
Title: Professor of Electrical Engineering

Acknowledgments

I would like to thank Professor Troxel deeply for his kind support and advice throughout this past year. In addition to support me financially as a research assistant, he has given me good advice for many times when I had problems with my research. Because English is not my first language, I have a lot of trouble of explaining things, but he always tries to listen to my poor English and helps me with my bad writing. I am grateful to have such a generous adviser as Professor Troxel.

I also would like to thank James Kao who is the first person to start this remote microscope project. He patiently taught me what he had done for the microscope project and suggested many valuable ideas. His basic architecture of the remote microscope was nicely done and thus makes the incorporate of the automated microscope which is the important part of my thesis easily. Actually, my thesis includes and mentions a lot of his thesis. I would like to thank William Moyne for his help and his great knowledge about computer. I am grateful for everyone in the CIDM group for suggestion and support.

My family and my friends at MIT have provided inspiration and support for me to progress and go on with my research.

Table of Contents

List of Figures

List of Tables

Chapter 1

Overview

The MIT remote microscope[1] allows a user to remotely control and view a distant microscope over the internet. Although the remote microscope was primarily designed to be used as a remote inspection tool during a remote fabrication of integrated circuits, it is actually a general tool that will allow any user with an access to an X-window UNIX workstation to view any type of specimen under the microscope [Kao95]. The remote automated microscope which is the initial remote microscope incorporated with an automated microscope provides a convenient, precise, and computer-controlled operation of the microscope. It provides a remote, hand-free microscope that needs no attendant other than for initial loading of a wafer.

The automated remote microscope is versatile; it operates over the internet and consists of a graphical user interface running on an X-window UNIX workstation that communicates with a server program running on a PC connected to an automated inspection microscope. The server program also communicates to a microscope controller which controls the automated microscope and a video module that contains a frame grabber board which is capable of capturing a video signal from a CCD camera that is mounted on

---

[1] The initial manual remote microscope version was developed by James Kao in the MIT CIDM group. The automated remote microscope is the modification of the initial manual remote microscope. See [Kao95].

the top of the microscope. To a user though, the remote microscope is simply a graphical user interface that runs on a workstation because the hardware implementation is hidden from the user.

Typically, a user operates the microscope by setting an X-Y-F pan position,[1] selecting a magnification setting of the microscope and a method of focusing (either automatic focus or manual focus) for the new image, and then requesting a new image from the microscope. When the server process receives a request, it relays the request to the microscope controller to position the microscope and do auto focus if requested. After the controller has notified the server that the microscope is ready for an image capture, the server directs the video module to grab a new image of size 640 x 480 pixels at 16 bits per pixel[2] from the video camera. Then, the server transmits the image back to the client to be displayed.

The remote microscope also allows multiple clients to connect to view the microscope at the same time during a conference inspection session. This allows any number of individuals on the internet to simultaneously view the microscope cooperatively, which can be very productive and effective when semiconductor researchers work together during processing steps.

The current automated remote microscope enables inspection functionalities that require an exact coordinate positioning and computer-controlled movement of the microscope. The current automatic functionalities include automatic focusing, microscope positioning, and changing of magnification. Other functionalities which might be added in the future include storing a sequence of images of the wafer during processing steps for later inspection, scanning the wafer to find areas of interest, computer-based analysis of microscope images such as finding defects on the wafer, and matching and comparing between a circuit layout and the actual wafer after processing steps. These possible applications of the automated remote microscope will prove to be useful even for inspection within local facility.

---

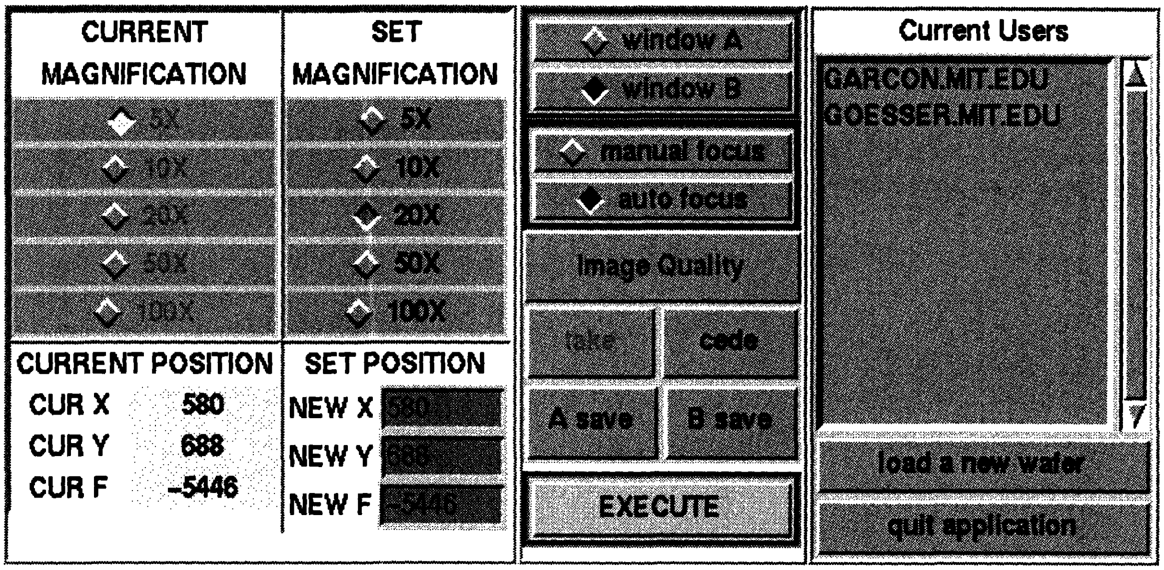
[1] F-axis is the focus axis which is the vertical motion of the microscope stage.

[2] Actually the colors are quantized to 200 colors which are used in the GIF format image before transmission. See appendix A.

Because safety is a major concern in the remote operation of any automated microscopes, the current automated remote microscope has incorporated a safety check ensuring that no microscope objective will crash into the wafer as a user lets the microscope operates automatically without his/her presence. Because the microscope does not have a sensor telling the distance of the objective and the wafer, it relies on the autofocus algorithm to find the optimal focus position which guarantees that the microscope is in a safe focus position.

The hardware needed to implement the automated remote microscope includes a PC running OS/2 Warp operating system, a framegrabber board, a high resolution CCD video camera mounted on an inspection microscope, and an automated microscope that has three computer-controlled stepper motors to control the microscope stage in the X-Y-F axes, a mechanism to select an objective lens, a programmable illuminator, and an RS-232 serial interface. The client, server, and microscope controller software was written at MIT and is easily exportable to other systems.

The user of the automated remote microscope is presented with a very natural interface to the microscope that provides two views of the microscope images. The user is allowed to set the magnification of the microscope, to set a new pan position by either entering coordinates or clicking on a region within the display windows, to set the method of focusing, and finally to initiate a capture where an image from the microscope is displayed into a selected window. The following diagrams are screen dumps of the remote microscope main console and two display windows of microscope images.



CURRENT MAGNIFICATION

◆ 5X
◇ 10X
◇ 20X
◇ 50X
◇ 100X

SET MAGNIFICATION

◇ 5X
◇ 10X
◆ 20X
◇ 50X
◇ 100X

CURRENT POSITION

CUR X 580
CUR Y 688
CUR F -5446

SET POSITION

NEW X 580
NEW Y 688
NEW F -5446

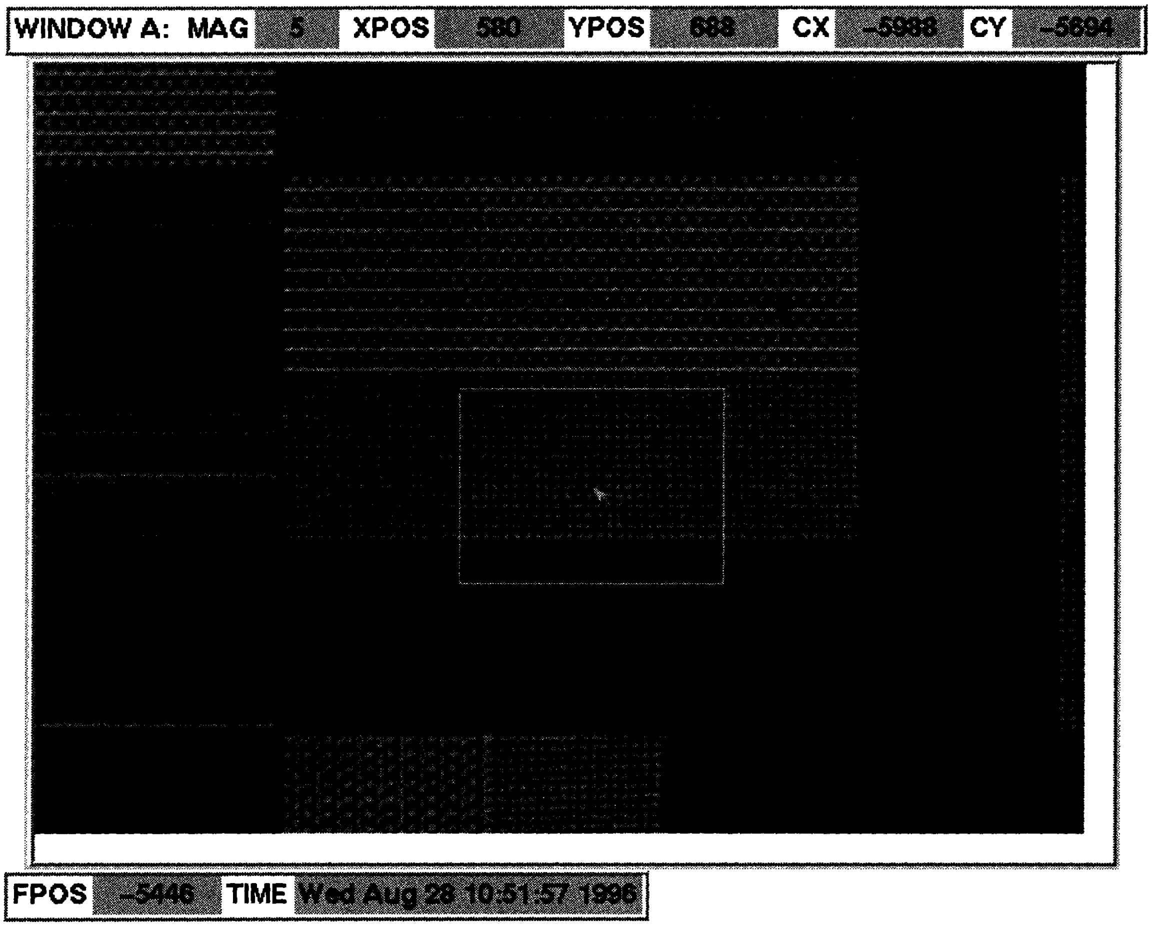
◇ window A
◆ window B

◇ manual focus
◆ auto focus

Current Users

GARCON.MIT.EDU
GOESSER.MIT.EDU

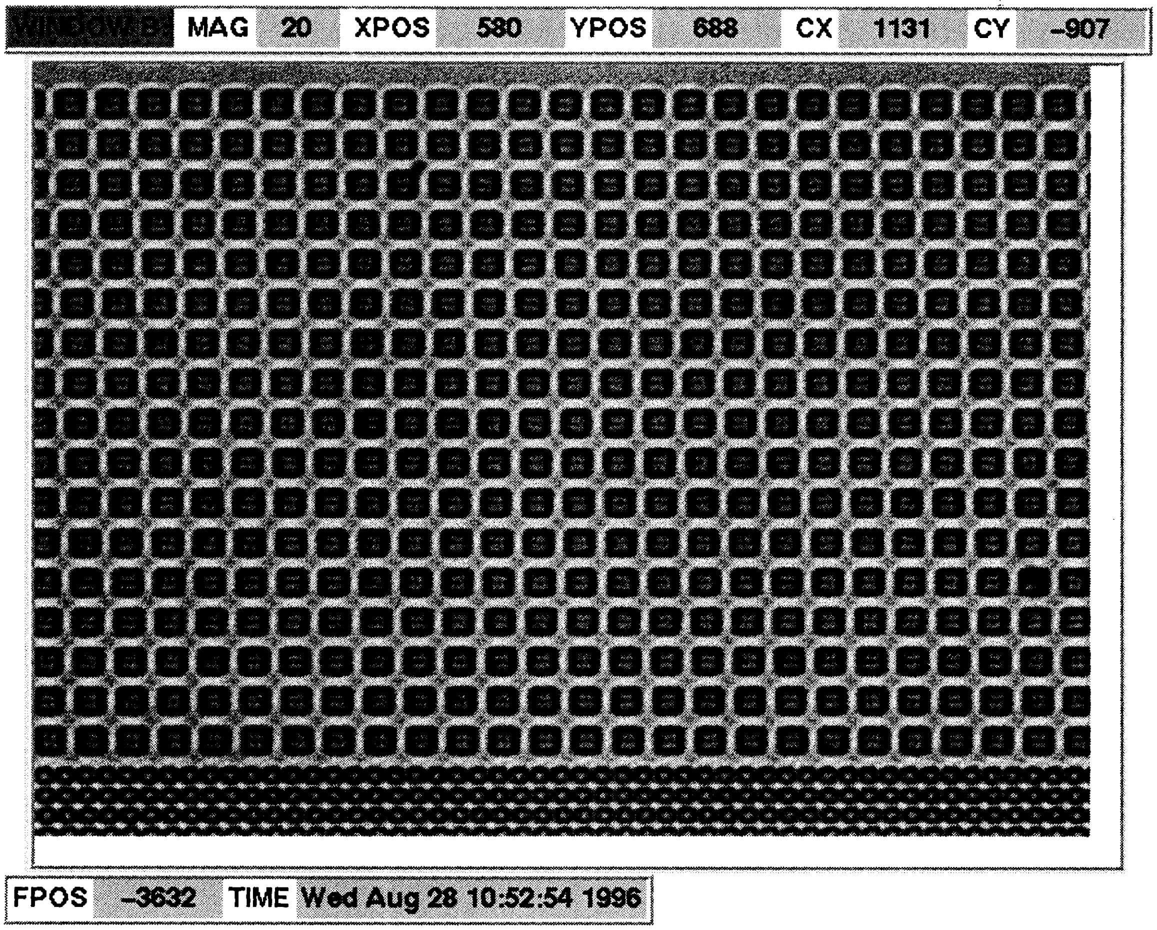
FIGURE 1. Remote microscope main console.



WINDOW A: MAG 5 XPOS 580 YPOS 688 CX -5988 CY -5694

FPOS -5446 TIME Wed Aug 28 10:51:57 1996

FIGURE 2. Remote microscope WINDOW A display. This shows a panoramic view at 5x. The rectangle around the middle shows the region of interest that will be zoomed in. This next image is shown in Figure 3.



WINDOW B MAG 20 XPOS 580 YPOS 688 CX 1131 CY -907

FPOS -3632 TIME Wed Aug 28 10:52:54 1996

**FIGURE 3. Remote microscope WINDOW B display.
This shows area of interest at 20x.**

Chapter 2

Background

Being developed at MIT as part of the Computer Aided Prototyping of Advanced Microsystems project (CAPAM) supported by the Advanced Research Projects Agency, the remote microscope was designed "to provide an inspection tool to aid in the remote fabrication of integrated circuits, which hopefully will improve the efficiency and flexibility of current semiconductor manufacturing technologies." By having an ability to perform different wafer processing steps at different facilities, wafer manufacturing can be more cost effective and robust. For example, expensive machines can be shared, rather than duplicated, among several facilities and wafers can be rerouted to different facilities if a machine goes down in a production line. In addition, the process designers and circuit designers can provide more interactive and collaborative role in maintaining quality by inspecting the wafer after each sensitive processing step at the remote facilities [Kao95].

2.1 The First Generation of the Existing MIT Remote Microscope

The initial remote microscope has demonstrated that it effectively allows a user to operate and view in "real-time," with a small delay, an actual microscope located at a distant facility. Thus, it has exhibited the feasibility of a remote microscope to inspect wafers during remote fabrication. In addition, the remote microscope has proved to be extremely versatile; it operates over the internet and allows a user to run the graphical microscope interface on any standard X-windows UNIX terminal. It also allows multiple users to view the microscope at the same time in a conference inspection. Thus, it provides more collaboration and sharing of resources among researchers in the field of semiconductors.

The MIT remote microscope was the first example of remotely viewing a microscope over the internet. The initial remote microscope was officially demonstrated between Stanford University and MIT, where users at Stanford could see and manipulate in real time a wafer located under a microscope at MIT. This was demonstrated at the August 1995 Stanford TCAD Symposium, "Hierarchical Technology CAD Process, Device, and Circuits," where the MIT-Stanford team described "A New Research Paradigm using Internet Collaboration" that would allow distributed researchers to work together in the semiconductor field [Kao95].

The software and hardware to implement the manual remote microscope uses available parts. The hardware includes a PC running OS/2 Warp operating system, a Targa+ framegrabber board, and a video camera mounted on top of an inspection microscope. The software was developed here at MIT and easily exportable to other systems. Thus, the remote microscope is a highly portable, low cost solution to provide remote inspection capability to researchers.

The initial remote microscope system has proved to be a useful foundation for future development such as automation of the microscope. This prototype system provides a client graphical user interface that is fully interactive, user friendly, and functional. The communication among the client(s) and the server program has been implemented with the CHAT messaging system [Car95] which can easily incorporate new programs into the system by messaging interface between each program.

In addition, there is generally a delay greater than 30 seconds between when a user requests a new microscope image and when he/she finally receives the updated picture. Although tolerable, this delay should be reduced to make the microscope more effective. The delay consists of several factors, including the network delay, and most importantly delay due to converting the image file format from a TGA format provided by the frame grabber board to a GIF format utilized by the Tcl/Tk client. To speed up the conversion, the current automated remote microscope has improved the conversion.

2.2 Incorporation of Automation to the Remote Microscope

Initially, the remote microscope was controlled by an attendant who manually set the magnification, pan, and focus of the microscope whenever the server received a request for a new microscope image from the user. After the attendant had positioned the microscope properly, the server grabbed an image from the video camera mounted on top of an inspection microscope and sent the image back to the clients to be displayed. Because the exact way in which the microscope moves is irrelevant to the remote user, this manual setup would still appear, form the user point of view, to be fully automated [Kao95].

However, there are several difficulties in using this "manual" microscope. Not only this manual microscope needs an attendant to operate the microscope all the time, but also it is difficult to pan to the exact X-Y-F microscope coordinates. Thus, it does not allow the applications that need well known microscope coordinates and the capability of an automated microscope such as automatic focusing, comparing the layout with the actual wafer after processing steps, and keeping record of wafer after each processing step.

The incorporation of the automation of the microscope thus provides a true remotely computer-controlled microscope and improves the usefulness of the remote microscope as an inspection tool for both remote and local inspection of integrated circuits. The automated remote microscope requires presence of an attendant only for putting a wafer properly on the microscope stage. After that, the microscope is able to pan, zoom, and do focusing automatically. Thus, a process designer can view and operate the microscope without suiting up to go into the clean room to operate the microscope. Integrating the automated remote microscope into the clean rooms would make the inspection process much more versatile, providing remote access as well as easy access to images in electronic form.

The automation of the remote microscope will be proved much more useful in the future as new inspection functionalities are developed. We hope that it will enhance the existing capability of providing feedback between process designer and remote facility during collaborative semiconductor fabrication in the future.

However, to be able to automatically control the microscope without the presence of the user, we must build a control algorithm that knows how to move the microscope stage and rotate the objective turret safely and thus not to crash the wafer. This automated control operation will be based on the current microscope position and magnification and the autofocus analysis of the video images of the wafer.

Chapter 3

Functionality of the Automated Remote Microscope

The remote microscope system was initially built for remote fabrication and inspection of integrated circuits. It enables a more collaborative and more distributed approach to fabrication of ICs. A user can have her wafer fabricated in the distant factory and still be able to inspect the wafer from her X-terminal and can do conference inspection with others engineers and designers. [Kao95].

The new functionality and improvement from the initial remote microscope system is mainly incorporation of automated microscope to the existing system. The general operation and set up of the remote microscope system, thus, still remain unchanged. The system still allows a user to view and "virtually control" a microscope over the internet. It also provide a conference inspection for multiple clients. The client/server architecture and most of the client user interface still unchanged as was described in Kao's thesis. However, to be able to fully control the automated microscope, a new version of the client user interface is introduced and the server has been modified to accommodate the automated microscope controller module.

3.1 Automation of the Microscope

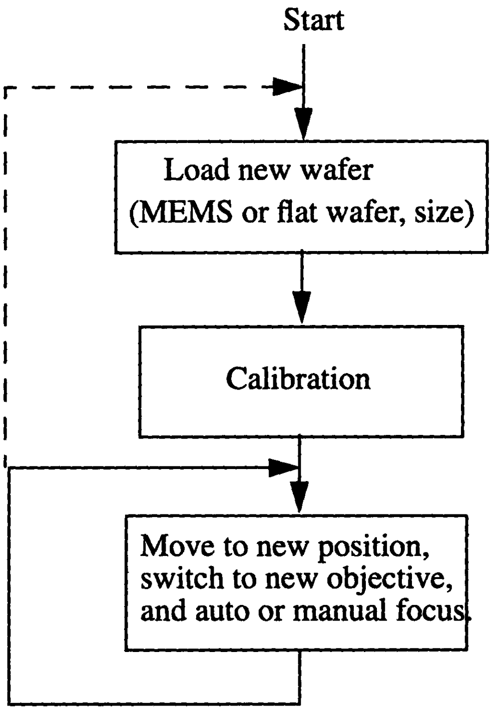
In the initial version of the microscope system, the microscope is controlled manually by an attendant. But the current microscope system is fully automated; an attendant is needed only for loading a wafer on the stage. The current automated microscope system has incorporated computer-controlled command interface to the microscope. Thus, a user can specify the position for the microscope stage to move to and the magnification. At the new position and new magnification before the new image frame is grabbed, the user must choose either he/she wants to use the automatic focus algorithm or manual focus, which the focus axis position for the microscope must be specified.

The true remotely controlled microscope is needed for convenience, precision, computer-based processing of images, and full computer control. However, the incorporation of the automatic microscope into the current system needs more functionality at the client user interface that will enable a user to set the pan and magnification, either manual focus or autofocus of the microscope. More details of how a user uses the automated microscope is discussed section 3.3.

3.2 Overall operation of the automated microscope

The overall operation of the automated microscope shown in Figure 4 starts with the loading of a new wafer onto the stage, wafer calibration, then it goes into a waiting loop for a user's requests to pan, change magnification of the microscope and do auto focusing, or to load a new wafer.

FIGURE 4. Overall Operation of the Automated Remote Microscope



In the current system, during loading a new wafer, the wafer is put at about the middle of the stage but with no accuracy in the location. This unspecification of the wafer position might cause a problem in the future when we needs to access the same position in a continuous inspection of the same wafer for several wafer-processing steps. Thus we will need some kind of position marking such as a perpendicular corner or a mark indicating the previous position of the wafer.

Before the calibration of the new wafer, the user must specify type (MEMS or regular flat wafer), size, approximated thickness of the new wafer so that the system can adjust the system parameters for movement and focusing algorithms. These parameters are important for the safety of both the microscope lenses and the wafer itself since an objective might be able to crash into the wafer if the parameters are incorrectly specified. For example, for a flat wafer, the system can afford to move the stage while under a high magnification lens (i.e. 100x) to a new point within the bounded area around the current position. For a MEMS wafer, which is rougher than a flat wafer, the bounded area is, possibly much, smaller than the area for a flat wafer. If incorrectly specified the type of the new wafer from MEMS to flat, the wafer is possible to be crashed.

During the calibration step, the microscope takes samples of focus positions for all magnification lenses at several locations on the wafer. It pans to several positions of the wafer and tries to get the focus positions under all the magnification lenses at each position. The user should be aware that this step takes some time to finish (usually about 5 minutes). More detailed description of the calibration step is in 7.2.4.

3.3 Client User Interface

To run the automated remote microscope system, a user must first connect to the server program which is running on a PC at the microscope site. The user will run the client's user interface on his workstation and connect to the remote microscope server. Multiple clients can connect to the server, but only one is allowed to control the microscope. All the clients will receive the same image that the client in control requests to the server. When a request for a frame by the client in control is sent to the server, the server relays the request to the microscope controller which moves the microscope to the desired position and magnification. Then the controller notifies the server to grab a new frame from the video camera, converts the image to a GIF format, and transmits the image back to all clients.

Before the microscope can be used, the server program must be already running on the PC and be ready to accept connections via socket from clients. When the client interface is initially started up, the user is prompted with a dialog box to input the server address and the port number[1] [Kao95].

FIGURE 5. Initial connection window. Process is requesting user to enter microscope location.

---

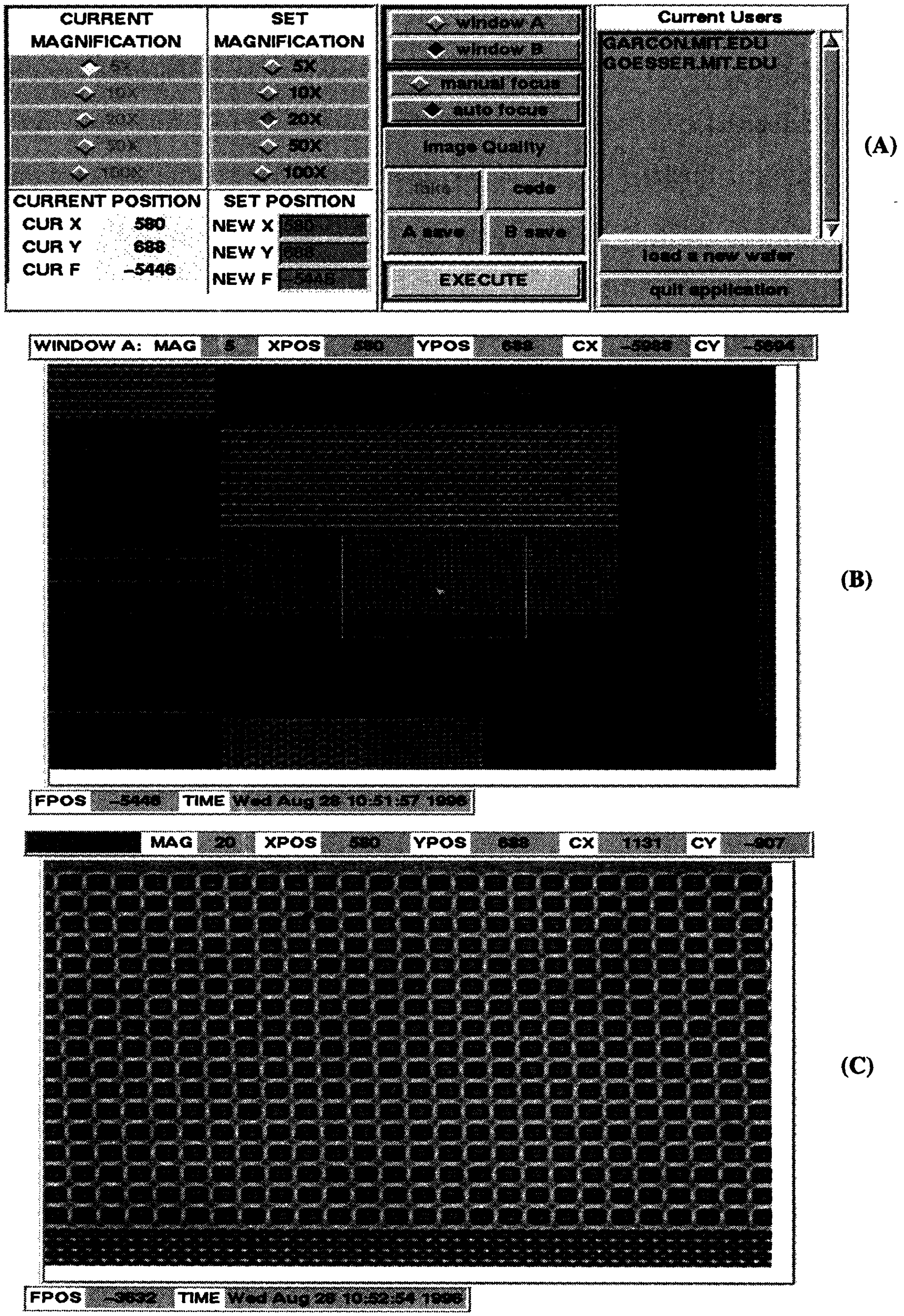
[1] The connection to the server and the dialog box were kept unchanged from the initial version written by James Kao.

After a connection is made, the client is presented with three windows which constitute the microscope interface: control window, window A and window B[1]. These three windows have been modified to accommodate the control of the automated microscope. The control window displays the current position and magnification of the microscope. It allows the user to set the new position and magnification, to choose method of focusing, and to select either window A or B in which the new captured frame will be displayed. Then the user can initiate a capture. The two windows are identical in all functionality. In the current system, we have only two windows to display images. Although having more than two windows is more effectively, the user can utilize the two windows effectively by keeping a large reference view in one window, and in another window, examining in more details a magnified view of an individual feature. The three windows of the remote microscope client user interface are shown in Figure 6.

The three windows which constitute the microscope interface was well implemented by James Kao especially the communication and the control of multiple clients. But the client user interface did not have enough functionality for controlling an automated microscope. Thus, only new interface for the automated microscope has been added to the initial implementation. The description of the initial version of the interface is well written by James Kao in his thesis [Kao95], and will not be repeated here. Only the new client user interface for the automated microscope will be described.

---

[1] Window A and B are called Global and Zoom window, respectively, in the initial version.



CURRENT POSITION
CUR X 580
CUR Y 688
CUR F –5446

SET POSITION
NEW X 580
NEW Y 688
NEW F –5446

Current Users
GARCON.MIT.EDU
GOESSER.MIT.EDU

WINDOW A: MAG 5 XPOS 580 YPOS 688 CX –5988 CY –5694

FPOS –5446 TIME Wed Aug 28 10:51:57 1996

MAG 20 XPOS 580 YPOS 688 CX 1131 CY –907

FPOS –3632 TIME Wed Aug 28 10:52:54 1996

FIGURE 6. “Console,” WINDOW A,” and “WINDOW B” windows that makes up client interface

3.3.1 The Client's Control Window

The client's control window is shown in Figure 6(A). The left most column displays to the user the current X-Y-F[1] position of the microscope stage[2] and the current magnification of the microscope. The directions of X-Y-F coordinate for the microscope stage position are shown in Figure 7. A global coordinate system is used in the remote microscope. The unit of the coordinates is in 0.1 microns (10^{-7} meters). The reason for this choice of unit is that a step of the focus axis stepper motor of the current automated microscope moves the stage 0.1 microns, and 0.4 microns for a step of the X-axis or Y-axis stepper motor. Thus, one increment of the F direction is equal to one step of the focus axis motor and four increments of the X or Y directions is equal to one step of the x-axis or y-axis motor. The calibration of the actual microscope position and the display position, which is derived in pixels, is critical for a correct transformation of the pixels to meters, and thus for a correct match between the display and the actual microscope position. The calibration and transformation will be described in section 7.2.5.

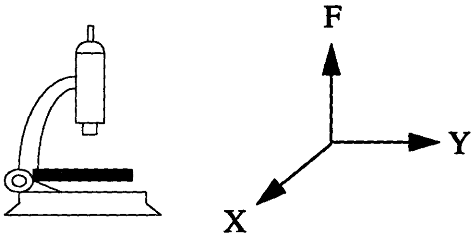


FIGURE 7. X, Y, and F coordinate system for the microscope stage position.

The next column consists of radio buttons that allow the user to select a new magnification setting for the microscope. Below these buttons are the entries for the X, Y, and F coordinates, NEW X, NEW Y, and NEW F, that the user would like to pan towards. If the user wants to increase the magnification, the new F position is ignored. Also, the user can set "NEW X" and "NEW Y" coordinates by clicking the left most button mouse on the display windows. The microscope is set so that the F coordinate is equal to 0 at the

---

[1] F-axis is the focus axis which is the Z-axis in the regular X, Y, Z coordinates.

[2] I will use the position of the microscope and the position of the microscope stage interchangeably.

mechanical stop. Thus, the possible working area of the microscope stage is in the negative F coordinate.

For safety reason, there are limits for each new X, Y, and F coordinates the microscope can pan to.The X and Y limits depend on the size of the microscope stage since there is nothing beyond the stage. Because the stage can move up as far as the bottom tip of the objective, the distance between the objective and the stage when the microscope is in focus determines the limit on F direction. For example, at low magnification lenses (5x and 10x), the stage can move up to the mechanical stop without crashing the wafer. Thus the limit on F is about the mechanical stop position. At other magnification lenses (20x, 50x, and 100x), the stage can move only to a new F position which is within a bounded distance from the current F position. However, the user need not to know about how big these limits are since the microscope will take care of the checking and adjustment of the new F position so that it will not violate the limit.

The "window A" and "window B" buttons in the next column allow the user to choose which window the new image will be displayed in once it is sent back from the server. Both windows are identical in functionality. The user can choose the method of focusing for the new image by pressing either "manual focus" or "auto focus" buttons. If the "manual focus" button was chosen, the new image is grabbed from the video camera immediately after the microscope stage has changed to the new position and magnification. In contrast, if the "auto focus" button was chosen, the microscope will attempt to do the automatic focus algorithm before capturing the image. Although the focus algorithm may not be successful, the image will be captured anyway and returned to the clients.

After setting the new position and magnification, the window to display the new image, and the method of focus, the user can execute the image capture by pressing the "EXECUTE" button at the bottom of the column. This button sends request containing all the settings to the server. The request will first cause the microscope stage to move to the correct coordinates and magnification (do auto focus if requested) and secondly initiate a video capture by the PC and transmit the image back to the client. The "Busy" window is popped up to every clients to let them know that the new request for an image capture is being processed.

The "TAKE" and "CEDE" control buttons are important for orchestrating the control of multiple clients for the remote microscope. The control of who is in charge of the microscope and the display of the current users in the next column are unchanged from the initial version and described in [Kao95] p. 32-35. The user can save the images in the two windows by choosing "A save" or "B save". The image will be saved in the GIF format. This functionality is useful for keeping a record during the fabrication in an electronic image which can be easily viewed by a program such as XV and incorporated into documents [Kao95}.

To load a new wafer onto the stage, the user must first press the "Load A New Wafer" button in the right most column. This button send a request to the microscope controller to prepare for a new wafer by changing magnification to the lowest (5x), lowering the stage, and moving it out so that the attendant who has been called from the user can safely and properly put a new wafer on the stage and then inform the user in control that the wafer is loaded and ready for wafer calibration. The user, who after clicking the "Load A New Wafer" button, is now being presented with a window asking the user to select OK if the wafer is ready for calibration, can now click the OK button to start the calibration.

Finally, the "Image Quality" menubutton contains several functions that allow the user to adjust the quality of the next image capture. The reason for the user to improve the image quality is that during the conversion of the captured image from TGA format to GIF format, the colors are quantized to less than 200 represented colors. The quantization takes a while to find the represented colors. To speed it up, the color table which maps every possible colors to the quantized color is generated and can be used for later conversion so that the quantization needs just only mapping the colors. The color table should be generated for each new wafer loaded. However, sometimes the represented colors do not do the job well. For example, if the color table is found from the mostly blue image, the represented colors are mostly in the blue tone. If a new, red-tone image is captured, the quantized image using the previous color table will have very few levels of colors since the previous, represented colors of red are few. Thus the user might try to grab the image again but this time with a color table generated specially for this image. The colors of the new image will be better represented.

The other function provided in the "Image Quality" menubutton is the type of color: gray or color. Although the gray image option helps to speed up the quantization, it is not much useful in reducing the quantization time when we can use the previous color table. But still this option is provided.

3.3.2 Client's Display Windows: Window A and Window B

Window A and window B (Figure 6(B) and 6(C)) display the microscope images. Any particular microscope image in the display has an associated magnification, X-Y-F position corresponding to the microscope stage, and the time the image is taken. In addition, each window also keeps track of the current cursor location whenever it is positioned within the microscope image (CX and CY values). These coordinates reflect the global coordinate system of the wafer, and as a result scale appropriately with changing magnifications.

The cursor position in each window plays a primary function during the operation of the microscope because one usually pans the microscope stage by positioning the cursor at an appropriate location and then clicking the first mouse button in order to set the NEWX and NEWY value (in the Control Window) that will be sent to the microscope during the next Execute command. The NEWX and NEWY (which will become the XPOS and YPOS values in the display after the grab is complete) correspond to the middle of the new microscope image. When the first mouse button is clicked and the NEWX and NEWY values are set, a pointer will also appear on the image, indicating to where the microscope will pan. In fact, every client connected to the server will see the pointer in the image, so that in effect, the pointer can be used to communicate between users to bring everyone's attention a certain feature in the microscope image.

The new pan position for the next frame grab can also be explicitly set by manually entering a value into NEWX and NEWY, and press return. Then the pointer will be placed according to the position specified in NEWX and NEWY. If however, the point to which one would finally zoom lies outside the borders of the global display, the easiest and quickest way to set the pan location is to manually enter the coordinates. Obviously, if one first views the microscope at the lowest magnification and makes incremental adjustments, then it will be easy to quickly locate the desired area of interest by a series button clicks on the mouse.

In addition to a pointer appearing in the image after a button click, a rectangle detailing the region that will be magnified will also appear. However, this rectangle is only created when the new magnification setting in the console window is higher than the current magnification of the reference image, because otherwise the effective rectangle would stretch beyond the boundaries of the display. Also, the dimensions of the rectangle are directly proportional to the ratio of reference magnification to new magnification. Effectively, the rectangular region will show exactly what fraction of the microscope image will be blown up during the next view. As the user changes the magnification in the control window, the size of the rectangle change according to the new magnification.[Kao95]

The user can also generate arbitrary size rectangles by dragging the mouse while clicking the middle mouse button. Although this rectangle will no longer have the meaning of showing what the next microscope image will be, it does serve a purpose (like the pointer) to direct people's attention to various features or regions within the microscope. The third mouse button is used to clear the rectangle. [Kao95]

Since the user might toggle between the two windows, the time stamp at each image capture can tell the user which window is the most recent image taken and thus the current view from the microscope. In addition, the green background at "WINDOW A" or "WINDOW B" at the upper left corner of the windows shows the most recent image capture.

3.3.3 Auto Focusing[1]

Usually, the autofocus algorithm gets the microscope to the correct position so that the microscope is in focus. However, sometimes the returned image looks blurred or totally empty because the auto focus algorithm fails to find the optimal focus position. If the features in the returned image looks blurred but can still be seen, the user should try another autofocus. The possible causes of the failure of the auto focus algorithm is discussed in section 6.1. If it still is not in focus, the user should try manual focus by setting the NEWF

---

[1] The algorithm for auto focus is described in 6.1.

variable manually.

However, if the returned image looks almost empty (cannot see any features), there are two possible reasons. First, the autofocus algorithm fails because the initial position of the microscope stage is too far from the optimal focus position and thus the sharpness information of the image is too small to start with. Secondly, the image looks empty because there is really no features in that region. To separate which reasons, the user should change the microscope to lower magnification and try to do another autofocus. The suggested lower magnifications if the microscope is already in high magnification are 20x, 10x, and 5x because at these magnifications the autofocus algorithm is more successful than at high magnifications like 50x and 100x. Supposedly, the autofocus algorithm should give a better result, but if not, change to lower magnification and repeat the auto focus.

Sometimes, the autofocus algorithm finds the focus position of dust on the wafer in stead. Since dust is on top of the wafer, it means that the F-position of the microscope stage is too far from the objective. To get to the desired features on the wafer, the stage must move up. The user can manually increment the NEWF a little and use manual or auto focus. The drawback of using autofocus in this case is that if the NEWF is not close enough to the real optimal focus position of the features, the autofocus algorithm will get to the same focus position of the dust. Thus it is suggested that for the dust case the user should increment the NEWF and use manual focus until the features in the image can be seen, and then can use autofocus.

However, for all of the above failures of the autofocus algorithm, the user can ignore the above suggestion and play with the manual focus. The user does not need to worry about the possibility of crashing the wafer because the limit on F-direction which protects the wafer already will adjust the NEWF value to correspond to the safety rule.

Chapter 4

Hardware Implementation

The current automated remote microscope system adds the automated microscope to the initial remote microscope system. The automated microscope includes the capability for the PC to automatically position, change magnification, and focus the microscope. The complete set up for the automated remote microscope is shown in Figure 8.

Although the cost of the automation of the microscope is the main cost, the cost of the entire system (which excludes the cost of the microscope) is in the order of $55,800. The cost of each component of the automated remote microscope is shown in Table 1. Because the automated microscope is only the prototype made specially for this automated remote microscope system, its cost is higher than it would be in the future. Other more economical microscopes can be used as well as long as they provide the same serial interface and commands as the current automated microscope because the communication between the automated microscope and the microscope controller is well defined in the current system.

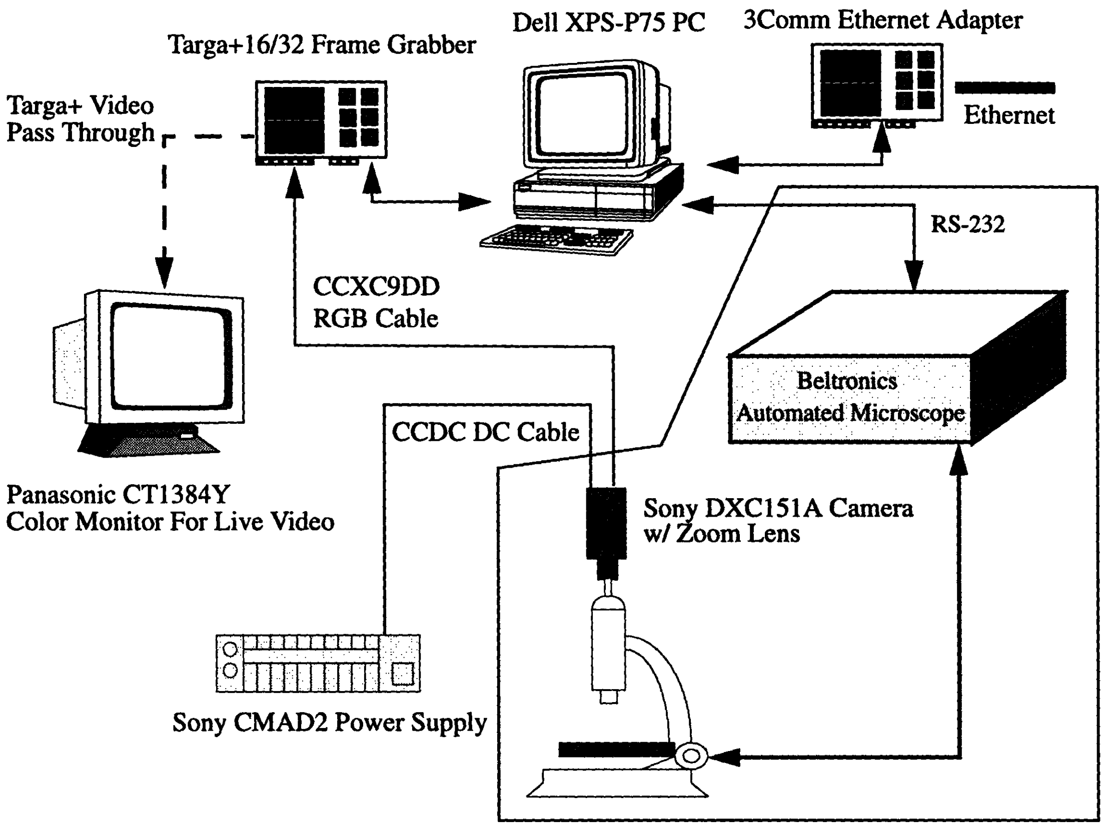


FIGURE 8. Component setup for remote microscope. The components in the shaded area are added to the initial manual remote microscope.

TABLE 1. Approximated cost of each component of the automated remote microscope hardware. The costs of all components except the automation of the microscope were for 1995.

Component	**Cost**
75 MHz Pentium PC	~$2100
Targa+ 16/32 Image Capture	~$1800
Video Camera Equipment	~$1900
Automation of the Microscope	~$50000
Total (excluding the Zeiss Axiotron microscope)	~$55800
Zeiss Axiotron Microscope	~$30000
Total (including the Zeiss microscope)	~$85800

4.1 Automated Microscope System

We need an automated microscope system that has a computer interface so that the commands for the microscope can be ordered by a PC. The PC receives the user's request from the server and processes the request by sending commands to the microscope and receiving replies back when the microscope has finished the order.

For the current system, we have Beltronics build an automated microscope system. The Beltronics Automated Microscope System consists of a Zeiss Axiotron microscope equipped with computer controlled: X-Y stage, auto focus system, an illuminator, and an objective lens selector. One RS-232 interface is required to control the system, which contains the following hardware modules: power supply, X-Y-F axis motor controllers and drivers, an I/O card used to select and monitor position of microscope objective, and a programmable illuminator. The automated microscope provides commands for controlling the movement of the microscope stage, changing magnification, setting light intensity, and automatically focusing. The list of some useful commands provided by the automated microscope system is in Table 2. The resolution in X and Y directions is 0.4 microns, and in F direction is 0.1 microns. The current setting for RS-232 interface is at baud rate of 9,600, no parity check, and 2 stop bits [Bel96].

TABLE 2. Commands for the Beltronics Automated Microscope System

Stage Commands	Functions
calib s	calibrate stage
halt	stop all the motors
here	define the current position
move	move to specified absolute position
movrel	move relative to current position
speed	define final motor speed
status	checks if motors are running
where	returns position of axis
Focusing Commands	**Functions**
focus l	focus image using parameters optimized for 5x objective.
focus m	focus image using parameters optimized for 10x and 20x objectives.
focus h	focus image using parameters optimized for 50x and 100x objective.
fspeed	defines focus axis motor speed when focusing
ldprdist	transfers memory array data into focus hardware module[a]
rdprdist	reads vertical distance parameters directly from focus module hardware

TABLE 2. Commands for the Beltronics Automated Microscope System

Stage Commands	**Functions**
sig f	computes and returns value proportional to focus quality
write focus array	defines optimal focus parameters for each objective, specifically the vertical distance between each focus position in focus profile, and stores parameters in a memory array.
Turret Relating Commands	**Functions**
read I0	check if 5x objective is in position
write o2=0 and then write o2=1	rotates turret to the next objective in direction of decreasing magnification[b]
write o3=0 and then write o3=1	rotates turret to the next objective in direction of increasing magnification
Light Setting	**Functions**
volt i	read and set the intensity of the light

---

[a] Ldprdist command requires a short time to update the memory in the focus module hardware. Thus, we cannot do read this memory immediately right after write into it.

[b] To rotate multiple objectives, one must wait at least 1.2 seconds before issuing a next command to rotate turret to the next objective.

Although the automatic focus algorithm in the current microscope is implemented in hardware, it can be implemented easily in software running on the PC and might help to reduce the cost. However, the advantage of having in hardware is speed and simplicity. The future more economical automated microscope system might not need the auto focus hardware module.

Chapter 5

Software System Overview

The software architecture of the automated remote microscope is based on the architecture of the initial remote microscope. The important additional part is the automated microscope controller module. Moreover, to be able to control the microscope a few more messages are needed between the clients, the server and the microscope controller.

5.1 Architecture of the Automated Remote Microscope System

The architecture of the automated remote microscope (Figure 5) includes multiple client units which run on UNIX workstations, a server process, a microscope controller, and a video module which grabs video frames. The last three run on a PC at the microscope site. More than one client can connect to the running server. Each client unit[1] consists of a Tcl/Tk user interface and a client daemon for processing binary data. The communication between the client user interface, the client daemon, the server, and the microscope controller is implemented by the CHAT message passing tools [Car95]. The communication of the video module and the server is a socket interface (like a pipeline).

---

[1] In the future the client unit might be implemented in JAVA.

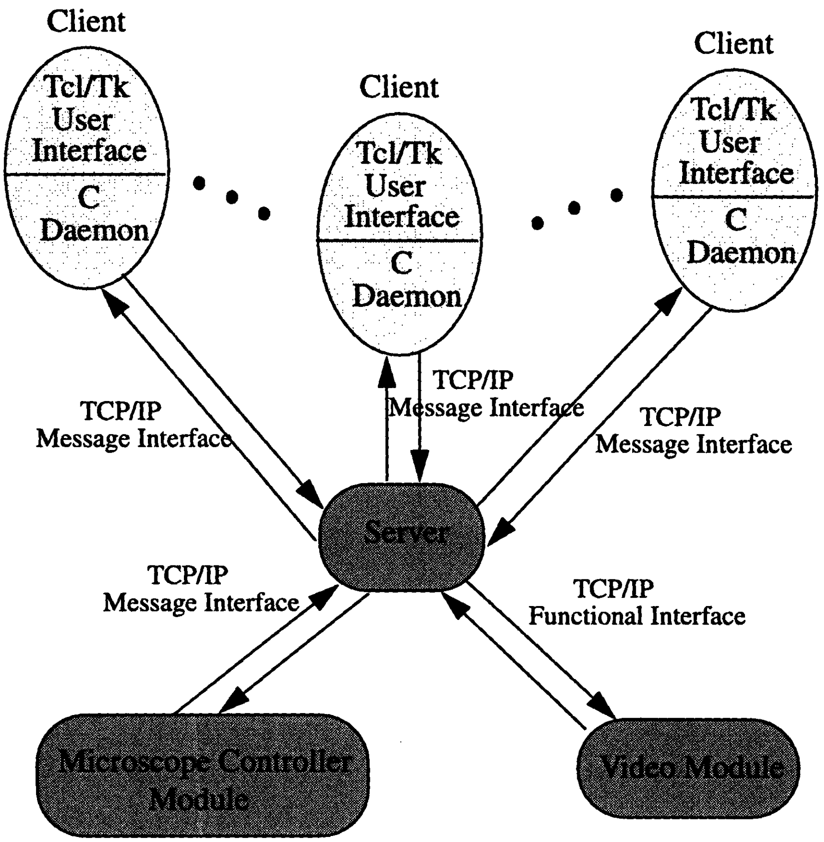


FIGURE 9. Architecture of the Automated Remote Microscope System. The remote client units run on X terminals. The server, video module and the microscope controller module run on a PC at the microscope site.

The reason for this architecture is simplicity because we already have the initial remote microscope working. Thus, we add the automated microscope controller which replaces the attendant who manually controls and adjusts the microscope for the initial version. The video module is attached to the server since after the attendant manually pan the stage and change magnification, the server is notified and then asks the video module to grab a frame from the video camera. Therefore, the easiest and most effective way to add the automated function is to make a microscope controller module connected to the server by the CHAT messaging interface. The server's main responsibility here is to keep records of clients and of who is in control of the microscope. It can relay messages from a client to other clients as well.

This architecture is not what I had in mind at the beginning. At first, I thought that the microscope controller module would include the video module as its submodule. This was because at first I thought that the controller had to implement the autofocus algorithm in itself and thus had to be able to grab multiple frames of the microscope easily and quickly. For an autofocusing, the controller moves the stage to different F-position, calculate a measure of focus quality of the image at that position, then repeats moving and calculating the focus quality until it gets to the optimal focus point. However, since the autofocus algorithm is already in the automated microscope hardware, we can use this one instead and simplify the programming of the microscope controller. Including the video module in the controller module would not be much harder because we just need to move the interface already in the server into the controller instead. If in the future we want to develop our own autofocus algorithm, the video module might need to be packed into the controller module.

Actually, it should be noted that the server need not be located on the PC at the microscope site because all the interface between the microscope controller and the video module is implemented by TCP/IP socket interface. Although the interface between the server and the video module is not CHAT message passing, it is connected by TCP/IP socket. Thus, the server can be located anywhere in the internet network. In some cases, it might be desired that the server be separated to be located elsewhere. For example, when the microscope is located at MIT, while most clients are in California. Having the server located in California, near most of the clients, will save network delay time to send images from the server to all the clients.

5.2 Microscope Controller Module

When a request for a frame capture at new position and magnification from the user of the remote microscope arrives at the server, the request will be relayed to the microscope controller. Then the controller interprets the request and calculates the sequence of commands which will be sent to the automated microscope to move the microscope stage to the requested position and to rotate the turret to the requested magnification. The microscope also does the auto focusing if requested. After the microscope has reached the new position and magnification, the microscope controller sends message to the server to notify that the server can initiate a frame grab from the video camera now. Then the server will notify the video module to grab a microscope image, convert the image to GIF format, and return the image to the server. The server then returns the GIF image to the user. At this point, one operation of frame capture is thus completed. The structure of the microscope controller module and the relation of the controller, the server, and the video module are shown in Figure 10.

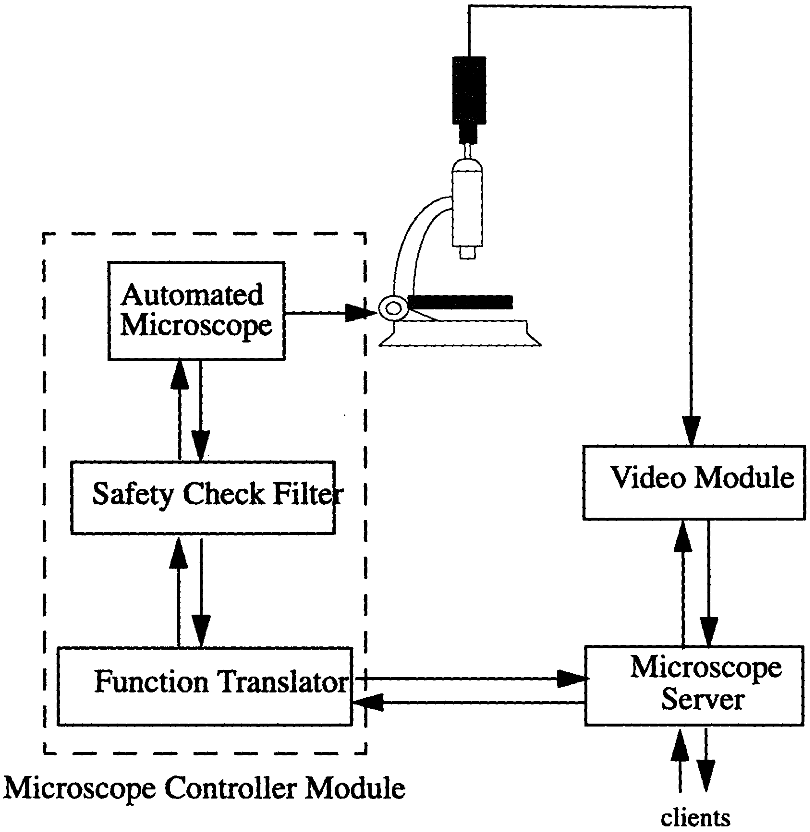


FIGURE 10. Microscope controller module in relation with the server and the video module.

The microscope controller module can be represented as three components: Function Translator, Safety Check Filter, and the Automated Microscope. The function translator translates the received request for the new position, magnification, and method of focus from the microscope server into a series of microscope commands such as move the stage, change magnification, and do auto focus. Before these series of command be carried out by the Beltronics automated microscope, they must be checked by the safety-check filter to avoid possibility of crashing the objective to the wafer. The limit on the F-coordinate of the new position is also checked by the safety-check filter. The commands are sent from the safety-check filter to the automate microscope via the RS-232 interface.

5.3 Additional Messaging for Automated Remote Microscope

The backbone of the communication between the client, the server, and the microscope controller programs is the message passing built into the CHAT communication tools. It is very easy to use the messaging interface to integrate the different programs in the remote microscope system [Car95].

The current automated microscope system only modified and added a few more messaging to the "manual" remote microscope version. The modified and additional messages (Table 3) are relating to request to grab a frame into either windows ("GRABFRAME" and "ZOOMGRAB") and to request to load and calibrate a new wafer ("LOADNEWWAFER" and "CALIB"). For details of the initial messaging that are still in use in the current automated version (see [Kao95] section 6.6).

TCL CLIENT - TO -DAEMON-TO-SERVER-TO-MICROSCOPE	**DAEMON - TO - TCL CLIENT**
COMMAND: "GRABFRAME" *XPOS:* X pan position *YPOS:* Y pan position *FPOS*: F position *MAG:* microscope magnification setting *FOCUS*: method of focus (auto or manual)	*COMMAND:* "GRABFRAMEREPLY" *XPOS:* X pan position *YPOS:* Y pan position *FPOS*: F position *MAG:* microscope magnification setting *FOCUS*: method of focus (auto or manual)
COMMAND: "ZOOMGRAB" *XPOS:* X pan position *YPOS:* Y pan position *FPOS*: F position *MAG:* microscope magnification setting *FOCUS*: method of focus (auto or manual)	*COMMAND:* "ZOOMGRABREPLY" *XPOS:* X pan position *YPOS:* Y pan position *FPOS*: F position *MAG:* microscope magnification setting *FOCUS*: method of focus (auto or manual)
COMMAND: "LOADNEWWAFER" *COMMAND:* "CALIB" DOIT: do calibration or not.	*COMMAND:* "LOADNEWWAFERREPLY" *XPOS:* X pan position *YPOS:* Y pan position *FPOS*: F position *MAG:* microscope magnification setting *FOCUS*: method of focus (auto or manual)

TABLE 3. Additional and modified Message formats used between Tcl/Tk Client and Daemon.

The “LOADNEWWAFER” request prepares the microscope stage for manually loading a new wafer. Then the “CALIB” request initiates the calibration of the new wafer. After calibration is done, the image of the new wafer at the middle of the stage and at the lowest magnification is returned with the “LOADNEWWAFERREPLY” to the user’s window A.

When the user sends a request for frame capture or loading a new wafer to the server program, the server relays it to the microscope controller program. After the controller has finished preparing the microscope for next frame capture, it sends a reply back to the server which then uses the video module to grab a frame and converses it into a GIF format image. Then, the server packs the image into the reply sent from the controller. This reply will be sent to the client C daemon who saves the image into a file for the TCL client to read and display. The rest of the reply is sent to the TCL client. An example of the messaging between each programs for the “GRABFRAME” request is shown in Figure 11.

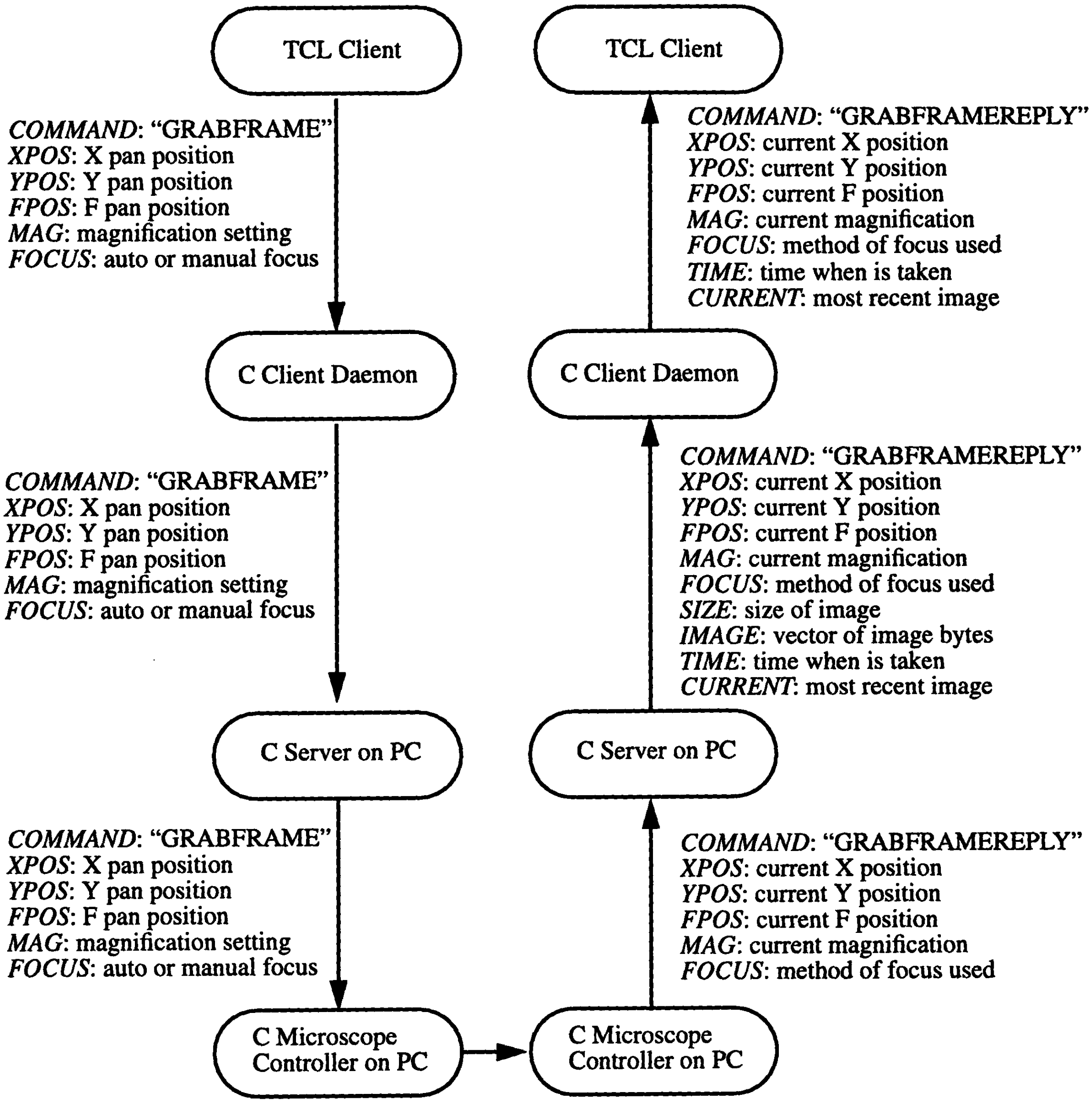


FIGURE 11. Example of messaging between each programs for the "GRABFRAME" command which requests to grab a new image at the new pan position and magnification with automatic focus or manual focus.

Chapter 6

Important Features of the Current Automated Microscope

This chapter will describe some important features of the current automated microscope that constrains how and under what conditions the microscope can move and rotate the turret safely and fast. The microscope controller which includes mainly of the movement algorithm for the microscope will be described in the next chapter. Thus, this chapter describes the features of the Beltronics Automated Microscope System that are the constraints or limitations to the movement algorithm of the microscope. These features are the built-in automatic focus algorithm, offset between a pair of objectives in the focus positions, depths of focus of each objectives, and the built-in safety mechanism.

6.1 Built-In Automatic Focus Algorithm

Because all algorithms relating to safety implicitly trust the autofocus algorithm to move the microscope stage to the correct optimal focus position, the autofocus algorithm is very important for the safety of the microscope. Although at 5x and 10x there is a mechanical stop that protects the wafer, at other objectives only the correct focus position ensures that the objective does not crash the wafer.

The autofocus algorithm depends on how the focus quality of an image is measured. When the microscope is at the "optimal" focus point, the image looks the sharpest. As the stage moves upward or downward from this position, the image becomes blurred. Thus, we can invent a measure of focus quality as the contrast of the image. The plot of focus quality as a function of focus axis position will have a maximum at the optimal focus position. If the measure of the focus quality is accurate, the image at the optimal focus position must looks in focus to our eyes as well.

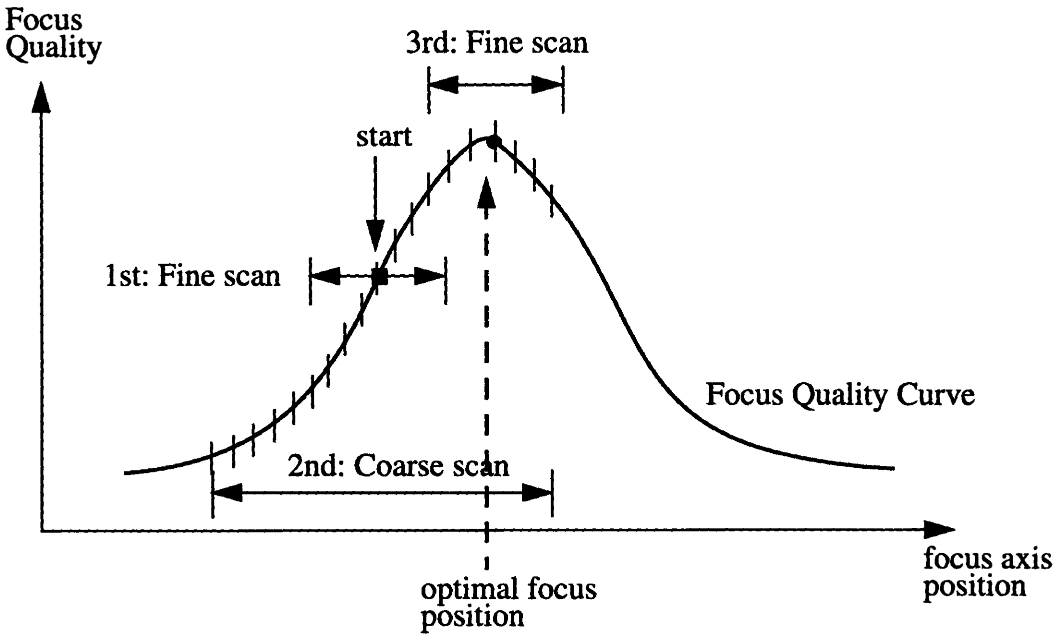
In the current implementation, the autofocus algorithm is implemented in the hardware of the automated microscope. The focus quality at a given vertical stage position is calculated by high pass filtering the video signal to detect edges and integrating the output of the high-pass filter within a window which has the size of two third of the video image for 100 video frames [Bel96].

Basically, the autofocus algorithm is as following: from the starting F-position, the stage lowers by a half of the search range and then takes incremental steps up for the specified number of steps. At each step, the focus quality is measured and a focus quality profile is generated. The profile is examined to determine if a local maxima exists. If it does, the corresponding F-position is used as the optimal focus position. If a maxima does not exist the scan range is increased to encompass a larger distance. The smaller scan is called "fine scan" and the wider is "coarse scan." If there exists a local maxima in the coarse scan, a fine scan is performed between the two coarse positions surrounding the maxima. The maxima located in this second fine scan is the optimal focus position. If no local maxima is detected in the coarse scan, the algorithm is aborted.

The number of steps in a scan and the ranges of the fine and coarse scan must be optimized and must be optimized for each lens of the microscope because low magnification lenses, such as 5x objective, have a much larger depth of focus than high magnification lenses, such as the 100x objective.

An example of a search for optimal focus position is shown in Figure 12.

FIGURE 12. Plot of Ideal Focus Quality as a function of focus axis position



When the current automated microscope is asked to do an autofocus, it does not return the result whether the autofocus attempt is successful. It only moves the stage back to the starting position if the autofocus cannot find the local maxima within the coarse scan. Hence I implemented the way to detect the result of the autofocus by after the autofocus is done[1], the focus quality is calculated by the "sig f" command at the current position, a lower F-position, and a upper F-position. If the focus quality at the middle point is the maximum among those at the three points, the autofocus execution is assumed to be successful. This is in a way finding a local maximum by using three samples on the focus quality curve. Because it takes time to move the stage and wait before measuring the focus quality, I use only three samples which should be sufficient. It would be faster and easier if the autofocus algorithm in the hardware returned the result of autofocusing because it has already known the focus quality profile. The positions of the lower and higher samples depend on the depth of focus for each magnification.

For the low magnification such as 5x, 10x, and 20x, there is enough room between the

---

[1] It is checked by detecting if the stage stays in the same F-position for at lest 60 milliseconds.

objective and the wafer for scanning the microscope stage during an autofocus search. Right now an autofocus search scans around the current F-position in F-direction 490 microns under 5x objective and 280 microns under 10x and 20x objectives. Thus there is enough space to move the stage safely upward. For the current implementation, if the first autofocus attempt at 5x, 10x, or 20x objective fails, the stage moves to another position and does another autofocus attempt. For 5x, 10x, and 20x, three tries of autofocus can be done, but for 50x and 100x only one try is allowed.

Examples of real focus quality curve as a function of focus axis position (Figure 13) shows that the focus quality curve actually has a optimal focus position which actually corresponds to the sharpest image. There is a big lobe near around the optimal focus point, and the curve stays almost constant outside the lobe. In some cases, there exist more than one local maximum. Actually, the images at these local maximums do look in focus. These maximums corresponds to the optimal focus positions of different features on different levels of the wafer.

The real focus quality curves are sometimes not smooth. Thus searching for local maximum by scanning over a wide enough range to find the local maximum is not a bad idea. If using this particular measure of focus quality, a hill climbing method to find local maximum might reach those small local maximums but not the greatest maximum. However, the possible advantage of a hill climbing method is the smaller number of samples it needs to reach the local maximum.

From the focus quality curves, the range of coarse scan should at least covers the width of the lobe because if the starting F-position for the autofocus is in the lobe, the autofocus will certainly find the optimal focus point if the point is in the coarse scan range. For the focus quality curves I have collected, I estimate that the coarse scan range for 5x objective should be at least 2500 steps of focus axis motor; for 10x, 1000; for 20x, 800; for 50x, 200; for 100x, 180. However, if the scan range is too big, it takes more time to scan, and also it becomes dangerous at high magnification as the space between the objective and the wafer is very small, thus, not much room left for the scan.

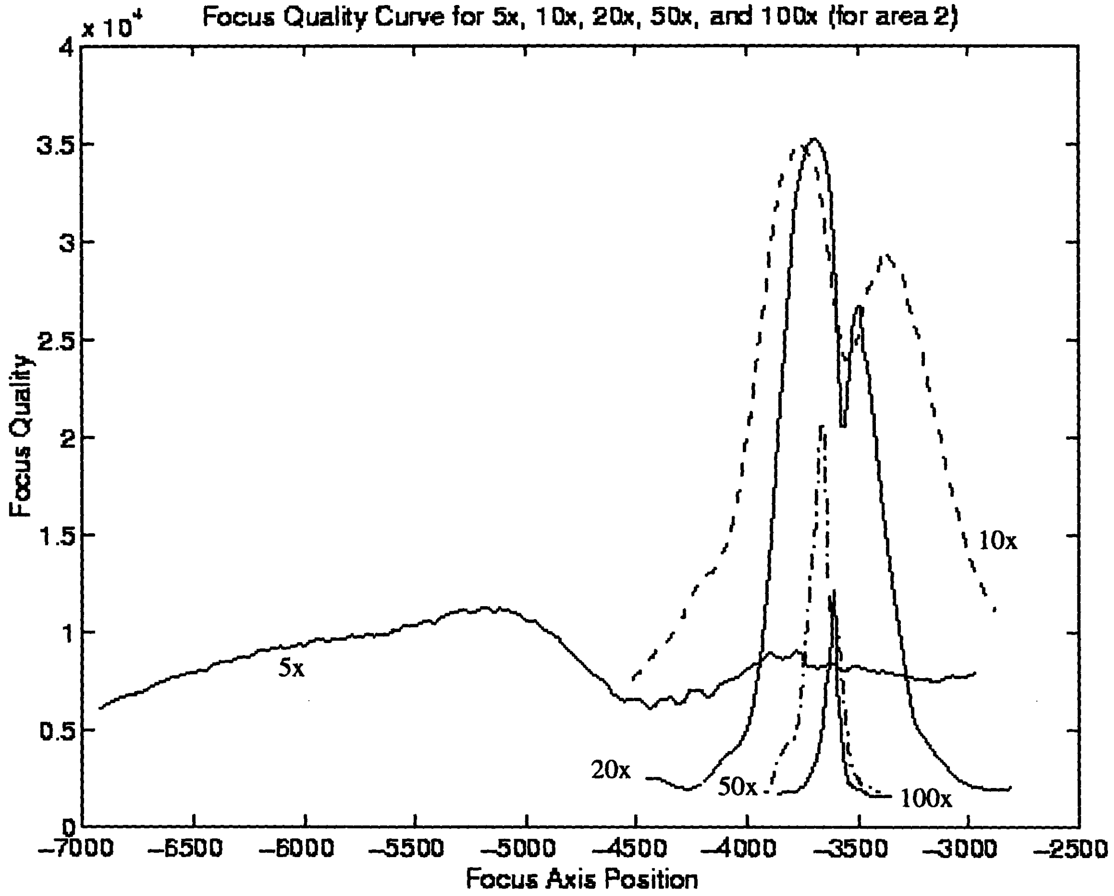


FIGURE 13. Example of the actual focus quality curves of every magnification lenses as a function of focus axis position. These curves are particular for this measurement of focus quality implemented in the hardware of the Beltronics Automated Microscope.

The size of the incremental step in fine scan and coarse scan also need to be optimized for each magnification. The step size of the fine scan determines the resolution of the auto focus. It depends on depth of focus (Table 6). For example, because depth of focus of the 100x objective is about 10-20 steps and thus moving the stage one step changes the image quality substantially, the fine step size should be one or two steps, but for 5x objective, depth of focus is around a thousand steps, the fine step size can be big as 50 steps and the algorithm still gets a good in focus image.

For low magnification objectives (5x, 10x and 20x) at which there is enough room for the stage to move safely, if the first auto focus attempt fails, the stage can move to nearby

F-position and do another auto focus attempt. The microscope can afford to do a few more moves and auto focus attempts safely. The new position for a new attempt is in the direction of higher focus quality (closer to the optimal focus point).

One of the useful features of the current microscope is that if the microscope is in focus at high magnification objective, it is still close to in focus at lower magnification objectives partly because of greater depth of focus at lower magnification. However, because there is a problem with the 100x objective, the microscope does not stay in focus from 100x to 50x objectives. But with a fixed offset in F-position, it stays in focus.

One important note about the autofocus algorithm in the Beltronics automate microscope is that if the range of either fine or coarse scans covers the position of the mechanical stop, the autofocus algorithm will stop at the mechanical position and it mistakes this position as the optimal focus position. I believe this happens because the autofocus scan which takes samples of focus quality along a range of focus axis positions does not know that the stage cannot move upward past the mechanical stop and thus it mistakes that the positions that above the mechanical stop can be reached. Moreover, after the stage has reached the mechanical step, it must move downward at least 1,000 steps before it can move up again because of the nature of the mechanical stop. Therefore, we should make sure that the autofocus search does not scan the stage above the mechanical stop position.

6.2 Offsets for Objectives

Another feature of this current microscope and possibly other microscopes as well is that the optimal focus positions for different magnifications are different. This might be because of different focus lengths in different objectives. Thus it might depend only on the microscope, and the offsets should be closely fixed. However, experiments show that the offsets for the same pair of objectives vary depending on wafer characteristics.

Three examples of the offsets between pairs of consecutive objectives are shown in Table 4. We can see that the offsets depend on magnifications and wafer characteristics. Even for the same wafer the offsets between the same consecutive magnifications can vary. Especially for 5x to 10x, the offset can vary substantially, for example, from 1717 in one area to 1280 steps in another area in the same wafer.

TABLE 4. Examples of the offsets between pairs of consecutive objectives. Example (a) and (b) are taken from different wafer. (b) and (c) are taken from different areas in same wafer. For example, the offset of 5x to 10x in (a) is -409 steps.

objective	infocus f-position	offset (.1 microns)
5x	-3685	-
10x	-4094	-409
20x	-4073	21
50x	-4110	-37
100x	-4065	45

(a)

objective	infocus f-position	offset (.1 microns)
5x	-5485	-
10x	-3768	1717
20x	-3616	152
50x	-3634	-18
100x	-3593	41

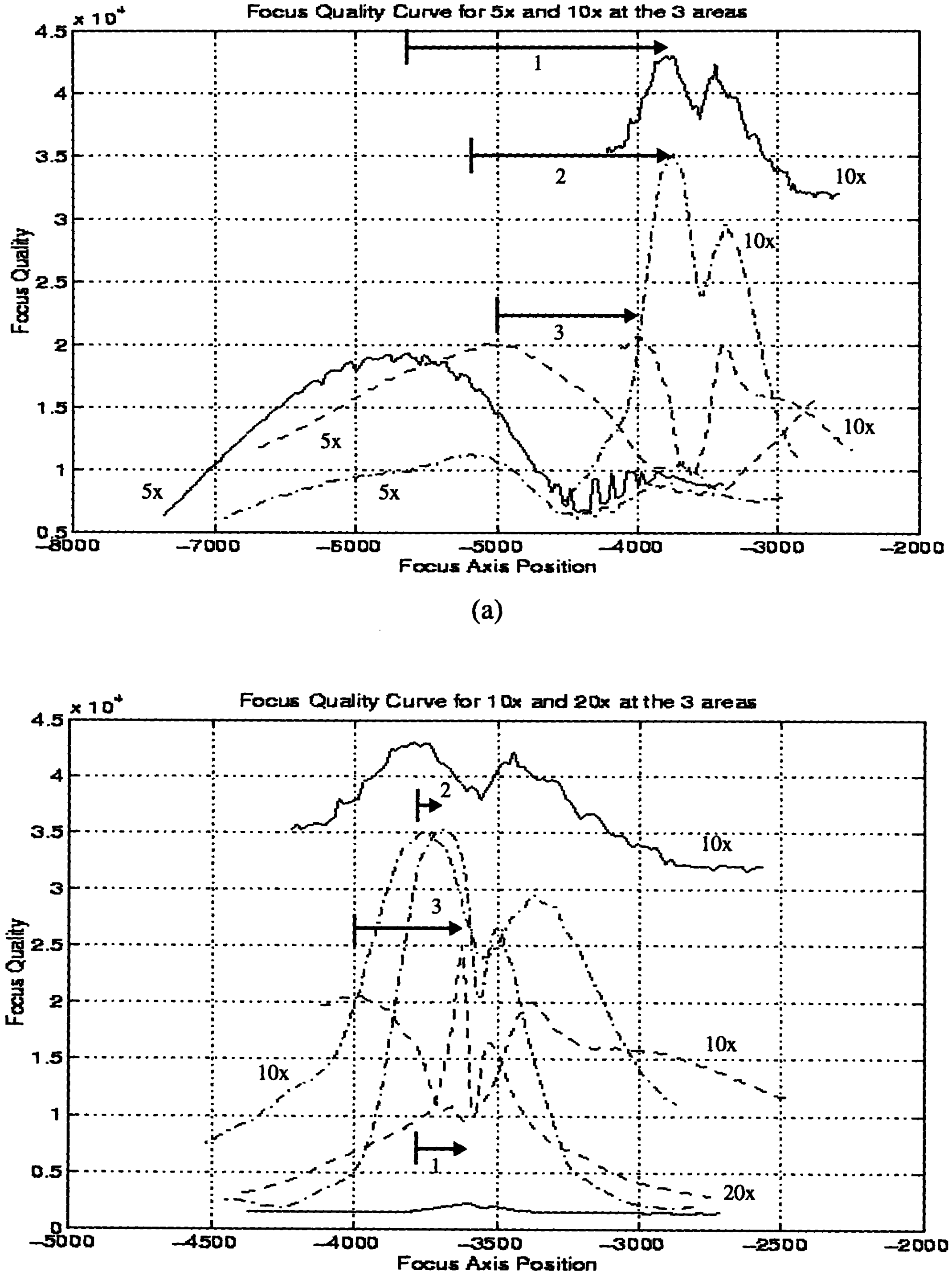
(b)

objective	infocus f-position	offset (.1 microns)
5x	-5048	-
10x	-3768	1280
20x	-3699	69
50x	-3663	36
100x	-3609	54

(c)

Because the offsets in the above examples are found by using the auto focus algorithm, it might be possible that the auto focus algorithm reached not the optimal focus positions. Thus the offsets which calculated from the difference between the two focus positions in both magnification objectives are incorrect. The best way to avoid the local maximum problem is to obtain the actual focus quality profiles for each magnification and to find in each curve the highest peak which is the optimal focus point.

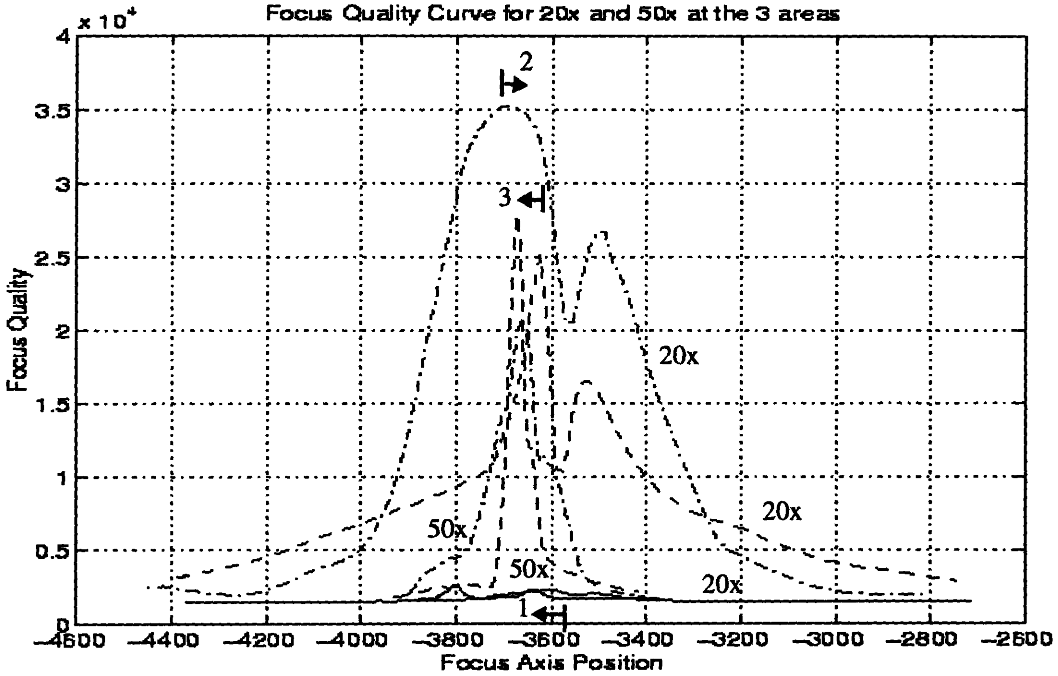
Evidences from the focus quality curves for each magnification at three different areas on the same wafer (Figure 14) show in more details that the offsets among the low magnification objectives (5x, 10x, and 20x) vary in the order of hundred steps, while the offsets among the objectives from 20x objective up vary in the order of ten steps.



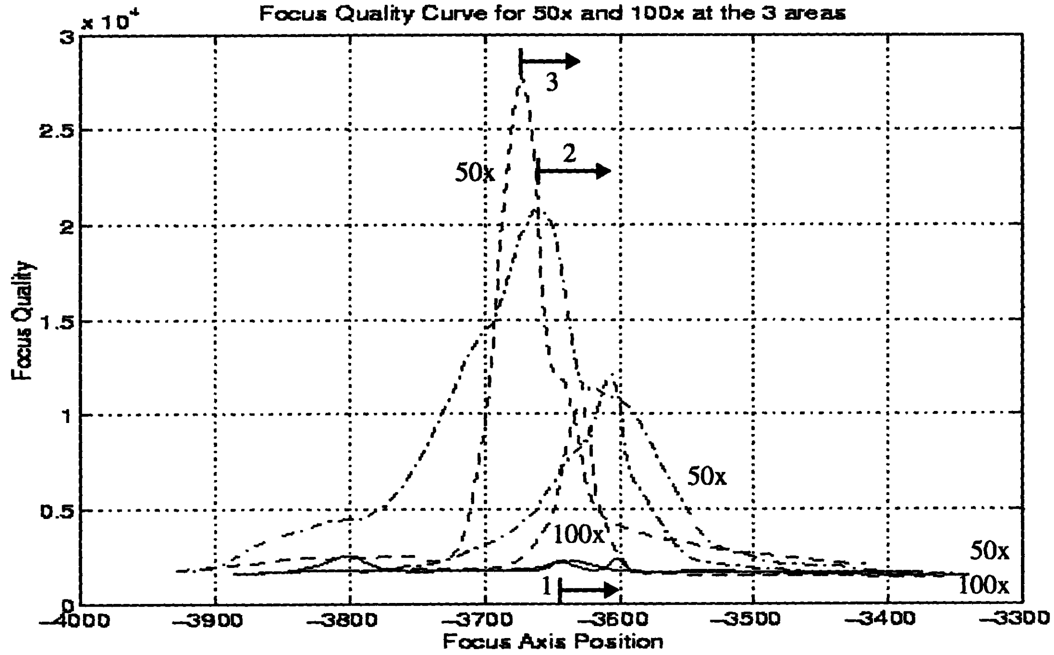
(a)

(b)

FIGURE 14. Comparison of the focus quality curves of adjacent magnifications for three different areas on the same wafer. The arrows show the magnitudes and directions of the offsets. We number different areas into area 1, 2, and 3.


Focus Quality Curve for 20x and 50x at the 3 areas
x 10^4
4
3.5
3
2.5
2
1.5
1
0.5
0
Focus Quality
2
3
1
20x
20x
20x
50x
50x
-4600
-4400
-4200
-4000
-3800
-3600
-3400
-3200
-3000
-2800
-2600
Focus Axis Position

(c)


Focus Quality Curve for 50x and 100x at the 3 areas
x 10^4
3
2.5
2
1.5
1
0.5
0
Focus Quality
3
2
1
50x
50x
100x
50x
100x
-4000
-3900
-3800
-3700
-3600
-3500
-3400
-3300
Focus Axis Position

(d)

Actually the peaks of the lobes on the focus quality curve are all the focus positions but for different levels of features on the wafer. For example, at one peak, the image looks sharp for one layer, but at the other peak, the image of different layer appears sharp. This is just a nature of the focus point of the lens. The lens can zoom at an object at one distance and thus make others objects at different distance out of focus.

After we have known the nature of the offsets, why do we need to approximate the offsets? Because we have a correct offset so that the stage can offset from the current, in-focus position at one magnification to another magnification and still stay in focus. Even we don't know the correct offset but only the approximated value, the microscope might be close enough to the optimal focus position at the new magnification that the autofocus can be done fast and with higher chance of success. The details of the calibration for the offsets will be described in 7.2.4.

There are also offsets in the X and Y coordinates between every pair of objectives. An X-Y offset for a pair of objectives is used so that after rotating the turret to the new objective, the middles point of both images looked through the microscope are still the same point. These offsets depends on the physical position of each objectives in the turret. For the current setting of the objectives, the offsets for each pair of consecutive objectives are shown in Table 5. The X-Y offsets will be accurate if the relation between the number of pixels and the number of microns the stage has moved is accurate[1].

TABLE 5. X and Y offsets for each pair of consecutive objectives for the current microscope.

pair of objectives	offset in X (0.1 microns)	offset in Y (0.1 microns)
5x -> 10x	228	-196
10x -> 20x	-36	-88
20x -> 50x	-980	836
50x -> 100x	1100	-820

---

[1] See 9.2, "Coordinate Transformation" in the client user interface.

6.3 Depths of Focus

Depth of focus is a range in which image still looks sharp and the user can see the features in the image clearly as the microscope stage moves in F-direction. The example of depths of focus for various magnifications, in Table 6, shows that the smaller the magnification, the greater the depth of focus. The depth of focus in a way indicates the tolerance of the quality of the image when the stage position is off from the optimal focus position. Thus, switching from infocus position at high objective to lower objective which has higher tolerance of the image quality, the image likely looks in focus. However, it is less likely from low to high magnification.

TABLE 6. Approximated depths of focus for a plain wafer

objective	depth of focus (0.1 microns)
5x	1200
10x	600
20x	100
50x	6
100x	3

6.4 Built-In Safety Mechanism

One of the important features of the microscope is that all objectives are recessed so that the objectives are protected, and, at 5x and 10x objectives, there is a mechanical stop to protect the wafer as well. Thus, the 5x and 10x objectives cannot crash into the wafer. For other objectives, we need to have a safety-check filter that ensures the safety of the wafer.

6.5 Limitations and Constraints to the Movement of The Microscope

The big offset errors at 5x and 10x objectives mean that it is difficult to find the correct offsets from these low objective to high objectives such as 50x and 100x. As the offset errors at 5x and 10x is in order of hundred steps, if the offset is off by a few hundred steps from the true value, the stage might move up so high that 100x objective might crash into the wafer. Therefore, we do not know the correct offset for safe switching from a small objective to a high objective and possibly cannot do a successful automatic focusing at high magnification because the stage is too far from the optimal focus point.

However, from the offset errors from 20x objective to 50x and 100x objectives are in the order of ten steps, and there is enough space to tolerate the error in the offset. Thus, we have good approximation of the offsets to 50x and 100x from 20x and we can use the 20x objective as a doorman to the high magnifications. This means that before switching to 50x or 100x from the current 5x or 10x objective, the microscope must change to 20x objective first and try to get to the focus position at 20x before it can change to 50x or 100x objective.

Chapter 7

Automated Microscope Controller Module: Software Implementation Details

The automated microscope controller module is the heart of this thesis because this is the new part for the remote microscope system. This chapter will describe the implementation of the microscope controller software. The microscope controller processes the user's requests for positioning the microscope, changing magnification, and automatically focusing, and thus prepares the microscope for a frame grab.

Therefore, the main responsibility of the microscope controller is to calculate a series of microscope commands that will carry out the user's requests. This series of commands includes commands such as move the stage, rotate the turret, and do auto focus. Because the safety of both the wafer and the objectives is the most important issue, the movement of the microscope must follows several safety rules. These safety rules are set according to the constraints of the current automated microscope described in the previous chapter such as focus position of each magnification lens and the depths of focus. The rules are also constrained by the nature of the auto focus algorithm.

7.1 Tools to Operate the Automated Microscope

However, first the microscope controller needs to have higher-level functions or tools that it can use to operate the microscope. These tools hide the RS-232 serial interface to the automated microscope at the back so that the controller does not need to worry about how each tool is actually implemented but by the functional interface uses them directly to operate the microscope.

The package of tools that the controller uses to operate the automated microscope and their brief description is shown in Table 7. "SmartMove" is called every time a "GRAB-FRAME" or "ZOOMGRAB" request is received by the microscope controller. Using the movement algorithm described in the next section, it positions the microscope at the requested position and magnification safely and calls "TryToDoFocus" routine if autofocus is requested. "SafeMove" moves the stage to the new position and magnification safely by switching to a lower magnification if needed. "MoveTo" moves the microscope stage to the desired position by first transform the system coordinate to the microscope coordinate (see 7.2.5), and issues command to the automated microscope via the serial interface. The automated microscope in turn tells the motor drivers to move. MoveTo procedure returns when all the stepper motors stop moving.

TABLE 7. Package of tools to operate the automated microscope.

Procedure	**Brief Description**
int SmartMove (char *nx, char *ny, char *nf, char *nmag, char *focus)	Contain the movement algorithm that issues commands to move the microscope to new position with new magnification and do auto/manual focus.
int SafeMove (char *nx, char *ny, char *nf, char *nmag, int autofocus);	Move the microscope to the new position and magnification. If the current magnification is high (50xx and 100x), switch to a lower objective (20x) first before moving.
int MoveTo (char *nx, char *ny, char *nf, int ignore_nf, int icmag, int autofocus, int checknf)	Move the microscope to the new position and magnification directly without switching to 20x first. Check and adjust for the F-position according to the F-limit.
void ChangeMag (char *nmag)	Change to new magnification.
void SwitchTo5X (void)	Turn the turret until it reaches 5x objective. This function is used for the first time the microscope operates or the microscope has been halted.
int TryToDoFocus (void)	Try to do auto focus. Return 0 if the attempt(s) fails.
void ReadyForNewWafer (void)	Prepare the stage for loading a new wafer.
void DoCalib (char *doit)	Calibrate the offset after the new wafer has been loaded.
void OffsetCalib (void)	Do calibration for the offsets.

"ChangeMag" routine rotates the turret the the requested magnification objective by calculating how many times to rotate the turret relative to the current magnification stored in CMAG global variable. For each turn of the turret, ChangeMag must wait for at least 1.2 seconds before it can issue another command to rotate the turret. If the next command to rotate the turret is issued less than this time limit, the automated microscope ignores the command, and thus the magnification the microscope is actually in and the one the controller thinks the microscope is in will be different.

"TryToDoFocus" procedure attempts to get to the optimal focus position by using the autofocus algorithm described in 6.1. When the "LOADNEWWAFER" request is received, the controller calls the "ReadyForNewWafer" procedure to lower and to move out the stage preparing it for loading a new wafer. Then, the user will send a "CALIB" request with a variable telling the microscope to do offset calibration for the new wafer or not. "OffsetCalib" does calibration for the offsets of the stage focus position between objectives. The algorithm is explained in 7.2.4. Usually, it takes about five minutes to do the offset calibration.

7.2 Movement Algorithm

The previous chapter describes the features of the microscope that constrain the movement of the microscope. This section describes the movement algorithm the microscope controller uses. The main issue the controller must consider is how to move the microscope and to change the magnification safely, but fast enough and with higher chance of successful autofocusing.

The controller must ensure that the series of commands issued to the automated microscope is safe both to the wafer and to the objectives. Since the microscope does not know the topography of the wafer, it can only assure that the wafer is not crashed if the microscope is near the optimal focus position. At 5x and 10x objectives, the wafer is guaranteed to be safe. Because the room between the 50x or 100x objective and the stage when the microscope is in focus is very narrow, to move to the new (x,y) position with this high magnification objective is dangerous. The microscope might need to switch to a lower magnification objective first, and then move safely under this new objective to the new (x,y) position and try to do autofocus there in order to position the stage in a safe position.

If successful, it can switch back to the high magnification objective.

The time for the automated microscope to process the commands must also be minimal. This can be done by minimize the number of commands especially those that require a considerable time, for example, rotating the turret and autofocusing. In fact, there is a trade-off between safety and speed. For example, the microscope might assume that, at high magnifications, it can move the stage safely within a small area in X and Y directions around the current (x,y) position with no need to switch to the lower magnification first, move to new (x,y) position, do auto focus, and switch back to the high magnification. This assumption will safe the time considerably, but in exchange for the safety.

To increase the chance of correct auto focusing, the starting F-position before doing auto focusing should be close to the optimal focus position. The controller can estimate the F-position the microscope stage should be by interpolation from the closest points that it has already found the focus positions.

7.2.1 Focus Map

To secure and speed up the auto focusing, the positions of all the optimal focus positions that have already been found by the autofocus algorithm are stored on a map, called "focus map." The idea of keeping the map which represents the actual topography of the wafer is proved to be very useful. The microscope can use the known points on the map to estimate the F-position at the new (x,y) position.

The focus map approximates the X and Y coordinates of the wafer by a grid map. It keeps the (x,y,f) position and magnification of the infocus points. We assume that each point on the focus map represents the F-position of all the points closed to it. In another word, every point (x,y) is quantized to the nearest grid point. The size of the grid depends on the roughness of the wafer. The rougher the wafer terrain, the smaller the grid size.

An example of the focus map (Figure 15) shows four points currently on the map. The optimal focus positions are known for each points. The magnifications at which the focus positions are found also kept. For example, at x=600 and y=200, the focus position is -3632 under magnification 100x. These focus points are calculated from the auto focusing algorithm. The user cannot manually set an infocus point into the Focus Map.

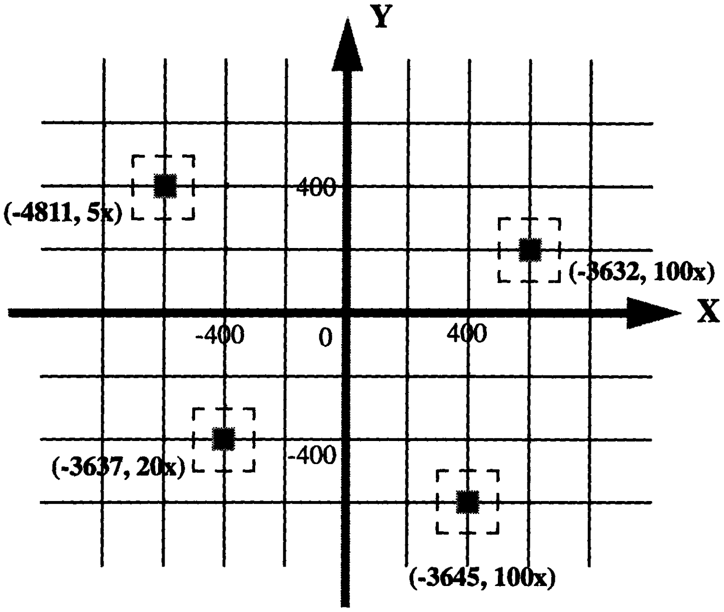


FIGURE 15. Example of the Focus Map

If several focus positions under different magnifications of the same (x,y) point are known, for simplicity the infocus f-position of the highest magnification is kept because the highest magnification has the smallest depth of focus; therefore, it is most likely that the microscope is still in focus at other lower magnifications if the stage just offsets the F-position. Actually, given the fact that the offsets are not correctly known, the focus map should keep the infocus positions of the lower magnifications as well so that the microscope can come back to the focus position exactly (no need to use the incorrect offset). However, this would complicate the programming.

The focus map is implemented as a hash table which stores the focus position and magnification at the quantized (x, y) position (the grid point). To read the estimated focus position at any magnification, the microscope just adds the offset from the magnification at which the focus position was recorded to the new magnification. The focus position at the (x,y) position gets updated if the current magnification is equal or higher than the one already found.

7.2.2 Assumptions about Safe Movement

Under the same magnification, the stage can move safely and directly to the new point which is within the "safe" box (Figure 16) around the current position and does not need to switch to lower magnification first. The height of the safe box is the limit on F-direction previously mentioned. The F-limit depends on the distance from the objective to the stage when in focus. The stage cannot move up beyond this limit. The F-limits for 5x and 10x objectives are the mechanical stop. The current implementation sets the distance that the stage can move upward or downward in F-direction with 20x objective to be 1,000 steps from the reference F-position, 200 steps with 50x objective, and 100 steps with 100x objective. The reference F-position is either found from the autofocus algorithm or estimated by interpolation of the nearby points. This flexibility for the stage to move from the focus position allows the user to adjust the focus of the image manually.

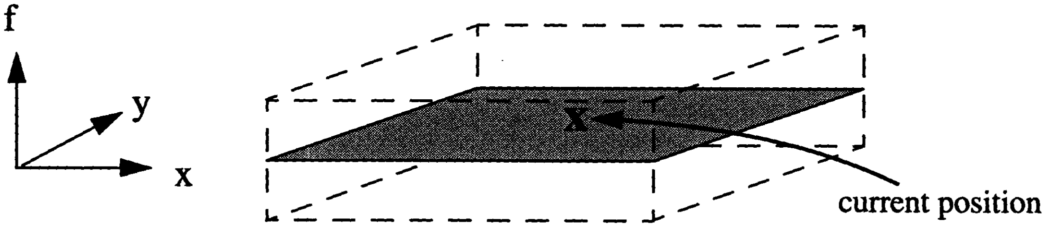


FIGURE 16. Example of the "Safe" box around the current position of the microscope. It is assumed that the microscope can move safely within the safe box.

The X and Y sizes of the safe box depend on the topology of the wafer and the magnification. For example, under same magnification, the safe box of a flat wafer can be much larger in X and Y directions than that of a rough wafer such as a MEMS wafer. Also, since there is small room from high magnification objectives to the wafer, the X-Y size of the box should be small enough so that the variance in height of the features within the box is not greater than this space.

Right now, we still need to find out the proper size of the safe box for each type of wafers. If the size of the box is too large, it is dangerous. If too small, it is likely that the microscope has to switch to the low magnification before moving. This slows down the response to the user's request. The type of wafer (MEMS or flat) is asked before calibration the new wafer, and the size of the safe box will be adjusted accordingly.

7.2.3 Movement Strategy

When a new request to move the microscope and change magnification arrives at the microscope controller, the controller uses the movement algorithm to derive a series of commands that will do the request. Because there is no built-in safety mechanism to protect the wafer at the 20x, 50x,and 100x objectives, the main function of the movement strategy is to protect the wafer at these objectives. With the 5x and 10x objectives, the microscope can move the stage safely already.

The initial implementation of the movement strategy is somewhat too conservative. To change magnification to a higher one, the microscope has to be in focus at every magnification that is between the current magnification and the desired one. To make sure that the objective will not crash into the wafer, the stage must be in the focus position first before it can rotate to higher magnification. The microscope is allowed to move under the same objective to the new point within a safe box. It does not use the focus map that keeps the (x,y) positions of the already-known focus positions. Thus, it could not estimate the new f-coordinate under the new magnification objective. The way to move to the new (x,y) point outside the safe box is to switch to a lower magnification objective (such as 5x, 10x, or 20x) first if the current objective is 50x or 100x. This is because it is dangerous to move with 50x or 100x objective, but it is very safe to move at 5x and 10x where the wafer is protected by the mechanical stop. Then, the microscope moves to the new (x,y) point and tries to rotate the turret to the desired magnification which might include trying to do auto focus first, then switching to higher magnification if the desired magnification is higher than this current one.

This initial implementation is proved to be ineffective since it does not use the information of the focus positions the microscope had already found. This information can be used to estimated the f-coordinate for the stage to be under the new magnification objective. The current implementation uses the focus map and an interpolation algorithm to estimate the f-coordinate from the close points. Thus the microscope can sometimes move to the new (x,y, estimated f) position with the new magnification directly, with no need to switch to the lower magnification first, move, and switch back to the high magnification again as in the initial implementation. This speeds up the time to go to the desired position and magnification; however, at the same time, the wafer is prone to crash if the estimated f-coordinate is significantly wrong from the actual value. Therefore, we must be extremely cautious on how to estimate the new focus position.

In the current implementation, the movement algorithm, whose diagram is shown in Figure 17, basically checks if the new point is within the safe box of the current (x,y) position for the new magnification. If it is and is not changing magnification, the microscope can directly move to the new (x, y, f) position. If the new F-position exceeds the limit on F at that current magnification, the new F-position is adjusted to be at the limit. The new F-position is not ignored in this case of same magnification. However, if the new point is in the safe box for the new magnification and the new magnification is lower than the current one, the microscope can change to the new magnification and move to new (x,y) position (ignore the new f-position which is not good anymore for the new magnification since it does not take into account the offset from the current magnification to the new one).

If the microscope needs to change to a higher magnification objective even at the same (x,y) coordinates, it has to estimate the F-position for the new objective from the focus map. It either looks up if the microscope has already been in focus under any magnifications at that (x,y) point or interpolates the new F-coordinate from the closest points on the focus map. The reason the microscope does not just offset the current F-coordinate and rotate the turret to the new higher magnification immediately is that the microscope may be out of focus at the current magnification, and thus it may be dangerous for the wafer at the new magnification.

An example will help to illustrate this point. Let the microscope currently be in focus under 5x objective at F=-4811 and the desired magnification be 20x. The user can then manually move the stage up to the mechanical stop (limit on F for 5x), F=0. Now, it is almost certain that rotating the turret to 20x, even with the offset adjustment (offset = 1100), the wafer will be crashed because the stage is too far up (the stage is now at F=0, but the focus position for 20x should be around -3500.). However, if the microscope looks up the new F-coordinate from the focus map, it will find that it has already been in focus under 5x objective at this (x,y) position with F=-4811, thus the stage should be at F=-3711 (which is calculated from -4811 + offset of 1100) for 20x objective. Thus, it can move to this F-coordinate first then rotate the turret to the 20x objective. Not only the microscope will be safe but also it will be near the focus point at 20x by using this method.

If the microscope has not been in focus at this (x,y) position yet which means that there is no point in that (x,y) position on the focus map, the new f-coordinate for the new magnification is estimated from the closest points on the focus map. The algorithm for estimation will be described in section 7.2.6. If the new (x,y) point is outside the safe box for the new magnification of the current (x,y) position, the f-position is estimated as well by either looking up the map or estimating from the closest points.

The function of the focus map lookup is to find if there are any points on the map that are within the safe box surrounding the new (x,y) point. If there are, the algorithm uses the f-coordinate of the closest point to represent the f-coordinate at the new (x,y) point. To be cautious about how to estimate the f-position by interpolation, the closets points that will be used for interpolation must be within a bound of the new point. This is because it is dangerous to use the points that are too far apart to do interpolation. If the points used for interpolation are too far apart from the new point they need to estimate, the microscope can still estimate the f-coordinate by interpolation but cannot directly move to the new point and rotate the turret to the desired objective (this is called "fast" move). It has to do "safe" move which is switching to the safe objective (such as 5x, 10x, or 20x) and then moving with this safe objective to the new (x, y, estimated f) point and switching to the desired objective later.

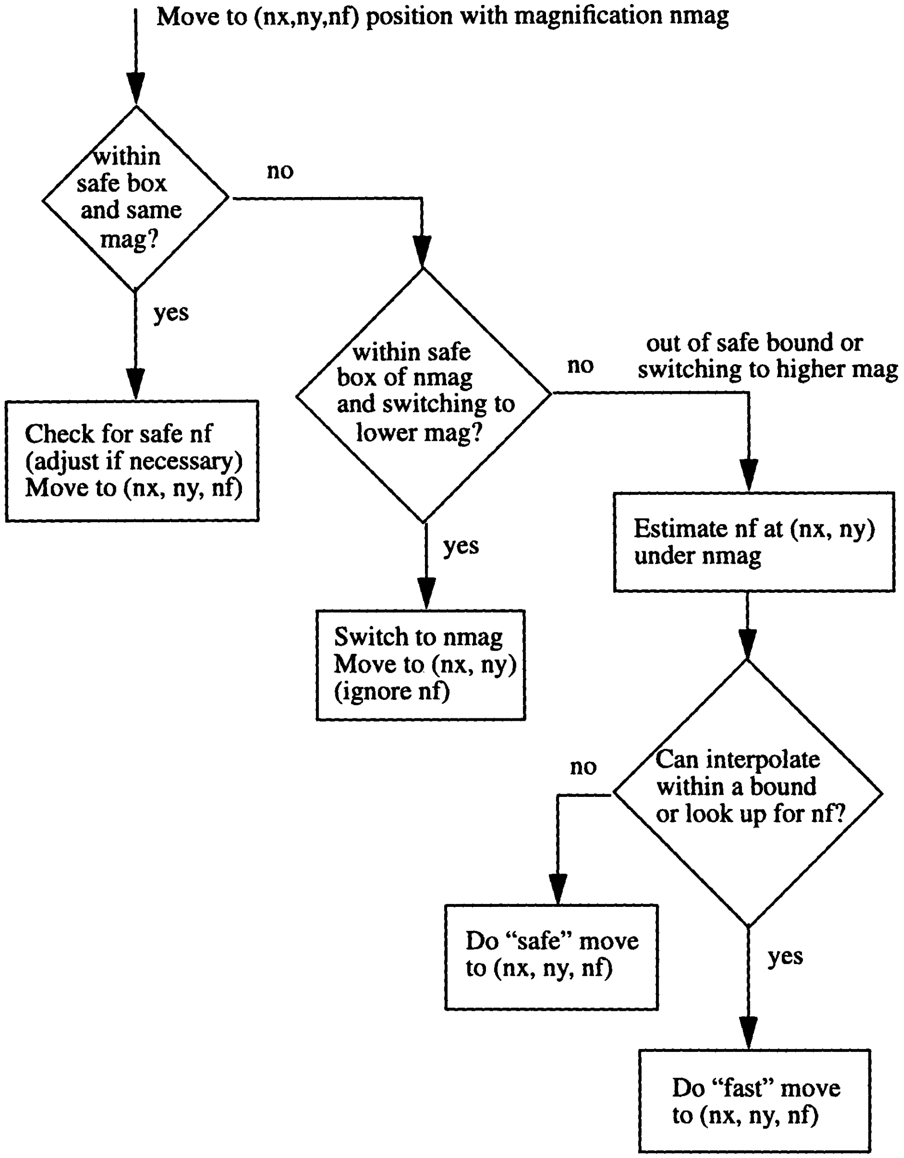


FIGURE 17. Diagram of the movement strategy used by the Microscope Controller to move to new position, (nx, ny, nf), and with new magnification, nmag.

The algorithm of the safe move is shown in Figure 18. Basically, the rule for the safe move is that the microscope stage cannot move with higher magnification objective than the safe objective, Safe_Mag which is either 5x, 10x, or 20x. If the current magnification is lower than the Safe_Mag, move with this magnification, and change to the new magnification later. If the current magnification is higher than Safe_mag but the new magnification is higher than the Safe_Mag, change to the new magnification and then move. The lower the magnification, the safer it is to move the stage, but the longer the time needed to finish the request because it needs time for extra turns of the turret. For a rough wafer, Safe_Mag should be 10x or 5x.

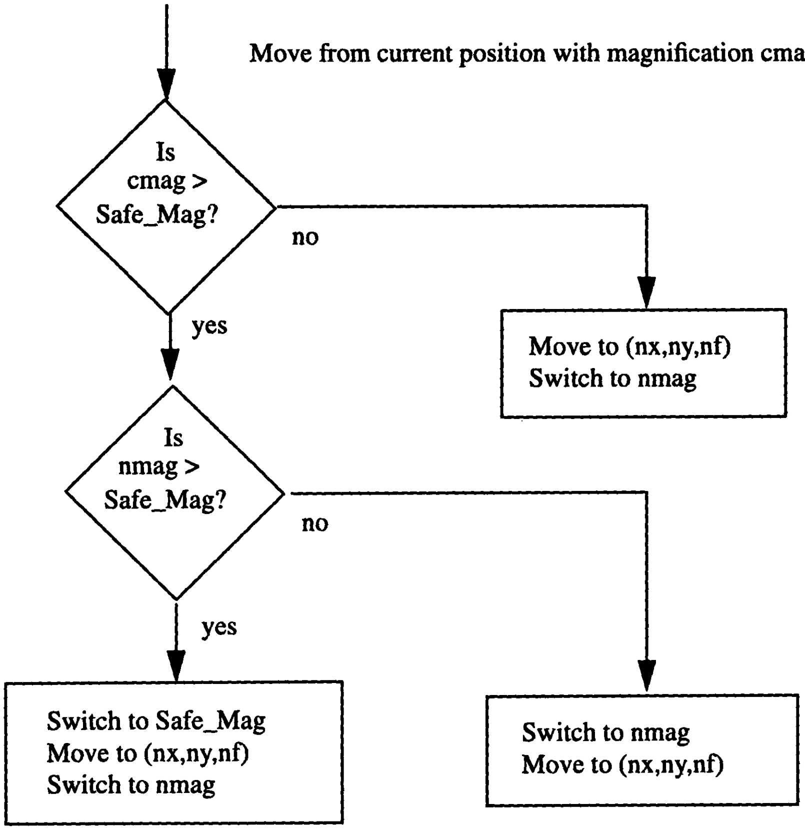


FIGURE 18. Diagram for the "Safe" move algorithm. The microscope wants to move from the current position with current magnification, cmag, to the new position (nx,ny,nf) and new magnification, nmag. Safe_Mag can be 5x, 10x, or 20x.

Figure 19 shows the algorithm to change to the new magnification. As we have already known that changing to a lower magnification can be done directly, with no need to stop at the middle magnification to refocus. The current microscope uses 20x objective as the "doorman" objective to the higher objectives.[1] Before changing from 5x and 10x objectives to 50x and 100x objectives, the microscope must first be able to be in focus at the doorman objective (20x). If it cannot be in focus at 20x, it returns FAIL and stays there with 20x objective. It does not change back to the original objective. Each rotation step of the turret to a new objective includes moving the stage for X, Y, and F offsets.

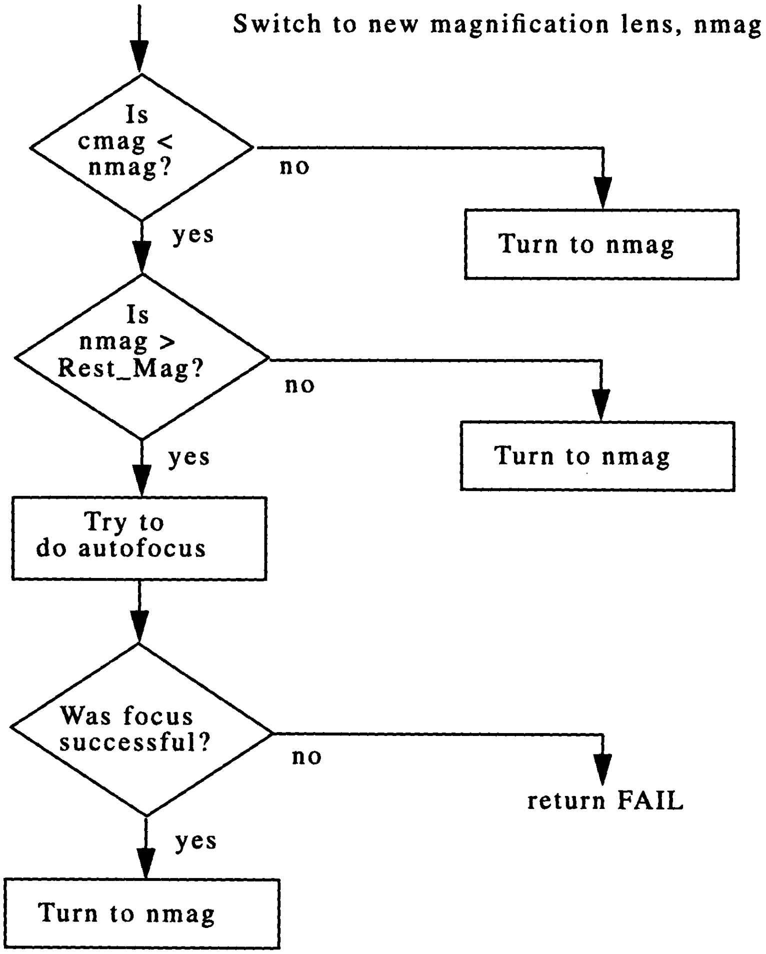


FIGURE 19. Algorithm for changing magnification.

7.2.4 Offset Calibration

In the current microscope system, the offsets are approximated during the calibration performed after loading a new wafer. The offset between a pair of objectives is an average

---

1. The reason is explained in section 6.5.

of all the offsets for this pair the microscope has collected during calibration. During the calibration, the microscope moves to a few different areas on the wafer to collect samples of optimal focus positions for every magnification. For each area, the microscope first tries to do focus at the lowest magnification (5x). If the attempt is successful, the microscope keeps the focus position, f0, and rotates the turret to a higher magnification (10x) while moving the stage by the amount of the offset for 5x to 10x, which is initially set as 0. Then it tries to do auto focusing. If successful, the microscope keeps the focus position, f1, for 10x. It finds the current offset which is the difference between the current position and the previous position at 5x, f1-f0. The new offset from 5x to 10x will be the average of all the offsets for 5x to 10x the microscope has stored up to the current offset. Then the microscope change magnification and moves the stage by the offset for 10x to 20x, and repeats the procedure until it reaches 100x and successfully does an autofocus at 100x. At this point, the microscope has successfully recorded a whole set of optimal focus positions from 5x to 100x. If in any magnification, the autofocus attempts fails, the microscope pans to a neighbor area and starts recording a whole set of optimal focus positions. Thus the offset between a pair of objectives is an average of all the offsets for this pair the microscope has collected during calibration.

In the current implementation, the microscope tries to record three whole sets of the optimal focus position from 5x to 100x. The X-Y positions of the sampled points are shown in Figure 20. The microscope will try to get a whole set of the focus positions at each dash rectangular area. Within the same rectangular area, it have three tries to get the whole set of focus positions before it moves to the new rectangular area. In Figure 20, the microscope starts taking the first sample at point labeled 1.0 (the middle of the stage). If it cannot get the whole set of focus positions at 1.0, it moves to point 1.1 and tries to get the whole set there. If successful, it moves to point 2.0 in the next rectangle. It repeats the process until it has completed the three rectangular areas, and then moves back to the first X-Y position that it has successfully reached the focus position under 5x. The values of the distance of the rectangles, “a”, and the distance of the points within the same rectangle, “b”, depend on the size of the wafer. While the microscope is taking samples of focus positions, the focus positions are also recorded on the focus map. Thus, the offset calibration is also good for building a focus map which will be used for estimation of the focus positions at new X-Y positions in the future.

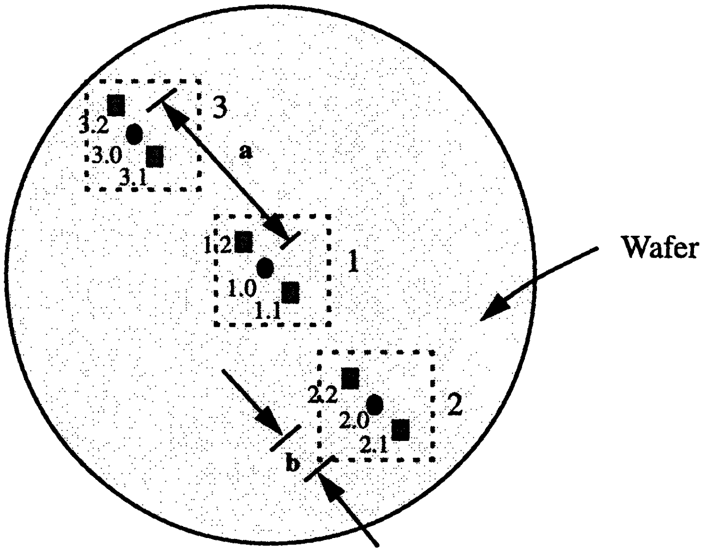
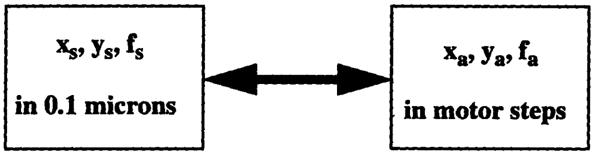


FIGURE 20. Locations of sampled points used for the offset calibration.

The above algorithm to find the offsets still uses the focus positions from the autofocus algorithm, not from the focus quality curves. Thus it might have some incorrect focus positions that result in incorrect offsets. The better way is to make sure that the autofocus really reaches the optimal focus positions by obtain the focus quality profile around the focus position. In addition, the calibration time needs to be considered. The more samples of offsets collected, the more accurate the estimate, but the more calibration time needed.

7.2.5 Coordinate Transformations

The remote microscope uses the X, Y, and F coordinates whose units are in 0.1 microns. The position used between the client programs, the microscope server, and the controller is in the global coordinate system. Because the coordinates used in the displays of the client user interface are in pixels, and in the actual microscope are offset and scaled by the step size of the X-axis and Y-axis motors, two transformations are needed: one between the display coordinates and the global coordinates; the other between the global coordinates and the actual microscope coordinates. The first transformation is discussed in the client user interface (section 9.2). The second transformation is described here. The second transformation between the system coordinates and the microscope coordinates is shown in Figure 21.



$$x_s = (x_a * \text{XYStep}) - \text{offset}$$

$$y_s = (y_a * \text{XYStep}) - \text{offset}$$

$$f_s = f_a \quad * \text{FStep}$$

FIGURE 21. Transformation from the system coordinates, (x_s, y_s, f_s) to the microscope coordinates, (x_a, y_a, f_a). XYStep is 4 (one step of X-axis and Y-axis motors is 0.4 microns/0.1 microns unit) and FStep is 1 (one step of F-axis motor is 0.1 microns/0.1 microns unit).

Because of the X, Y, and F offsets, the actual X-Y microscope position differs from the position the global microscope coordinate system used in the client's interface, the server, and the microscope controller (the F-values are the same in both coordinate systems). The offsets the stage moves when changing to new magnification is like a transformation from the user coordinates to the actual microscope coordinates[1]. The actual microscope coordinates must be transformed back to the global coordinates when the actual position of the microscope is read from the automated microscope.

The microscope keeps the total offsets in X and Y directions with the total offsets in X, Y and F all be zero at 5x objective. For example, from Table 5 the total offsets from 5x to 20x objectives in X and Y directions are 192 and -284, respectively. If the microscope is at x=100, y=40 when it is under 5x, the actual position (in 0.1 microns) under 20x will be x = 100 + 192 = 292 and y = 40 + (-284) = -244. To transform the actual microscope coordinates back to the global coordinates, the actual coordinates must be subtracted by the total

---

[1] after scaled accordingly to be in unit of 0.1 microns. It will be assumed for simplicity that the microscope coordinate has been transformed to 0.1 micron unit.

offsets, for example, the global coordinates under 20x should be x = 292 - 192 = 100, and y = -244 - (-284) = 40 as it is before under 5x. Thus, the global coordinate is kept consistent under the two objectives.

The transformation from the global coordinate system to the microscope coordinate system is done only at the functions that write commands to the automated microscope to move the stage. The system can be easily modified for other step motors which have different resolutions of stepper motors.

7.2.6 Estimation of Focus Position

To estimate the F-position, the movement algorithm either looks up the focus map or uses interpolation/extrapolation of the closest existing points on the focus map. This f-position is likely to be close to the optimal focus position if some cautions are put in the interpolation algorithm.

Because it is not safe to directly rotate the turret from the low magnification objectives (5x and 10x) to the high magnification objectives (50x and 100x), but it is safe from 20x objective to 50x or 100x objective, the microscope uses 20x objective as the doorman objective. Before it changes to the high magnification, it must be in focus at 20x first. Moreover, since the microscope stays almost in focus when switching from the focus position at higher magnification to a lower magnification, we prefer the f-coordinates that are the focus positions of equal or higher magnifications than the new magnification to estimate the focus position at the new (x,y) point. Moreover, only for look up but not extrapolation, we prefer the f-coordinate that is the focus position of the 20x, 50x, or 100x objective because the errors of the offsets among these objectives are only in the order of ten steps. However, it is too dangerous to use the focus positions of the lower magnifications than the new one for extrapolation because the offset errors can be multiplied.

When the algorithm looks up the focus map, it searches if there are any points within the safe X-Y bound of the new (x,y) point. If there are any, it chooses the f-coordinate of the closest point (give priority to the f-coordinate of equal or higher magnification than the new magnification) to the new (x,y) point to calculate the f-coordinate at the new point. Because the f-coordinate on the closest point, called f_0, may be for another magnification, the offset from that magnification to the new magnification is added to the f-coordinate of the closest point, f_0, to get the estimated f-coordinate at the new (x,y) point with the new magnification. This offset-adjusted function is used every time for the translation of the f-coordinate for one magnification to the f-coordinate for another magnification. If the new magnification is 50x or 100x and the magnification of the closest point is smaller than 20x, then the microscope must be in focus at 20x first; thus, the estimated f-coordinate will be for 20x objective, not the destined objective.

If the algorithm can estimate the focus position at the new point, the microscope will move to the new position by the “fast” move. Using the fast move, the microscope usually moves the stage directly to the desired position with the new magnification. But if the stop at 10x or 20x is needed, it switches to that magnification first. This ability to move in X and Y directions directly for the same magnification even at the high magnification is effective and saves time for the scanning application in the future.

If the algorithm cannot find any points within the safe bound, it searches for at least two closest points (again giving priority to the points with equal or higher magnifications that the destined magnification) in a bigger X-Y bound, called “interpolation bound” (about 4-6 times bigger than the normal safe X-Y bound). These two or three qualified points in the bound will be used for interpolation/extrapolation. Depending on the magnifications of these points, the microscope might need to be in focus at 20x first if the destined magnification is higher than 20x. For example, if the destined magnification is 100x and at least one of the two or three points is for the 5x or 10x magnification, there might be a big error in the interpolated focus position; thus, it is safer to stop at 20x first.

The linear interpolation/extrapolation of the focus position (nf) of the new point (nx, ny, nf) with three points whose coordinates are (x1, y1, f1), (x2, y2, f2), and (x3, y3, f3), respectively, is uniquely expressed as:

$$nf = a1*f1 + a2*f2 + a3*f3;$$

where a1, a2, and a3 are determined from the following nonsingular set of equations:

$$a1*x1 + a2*x2 + a3*x3 = nx,$$

$$a1*y1 + a2*y2 + a3*y3 = ny,$$

$$a1 + a2 + a3 = 1.$$

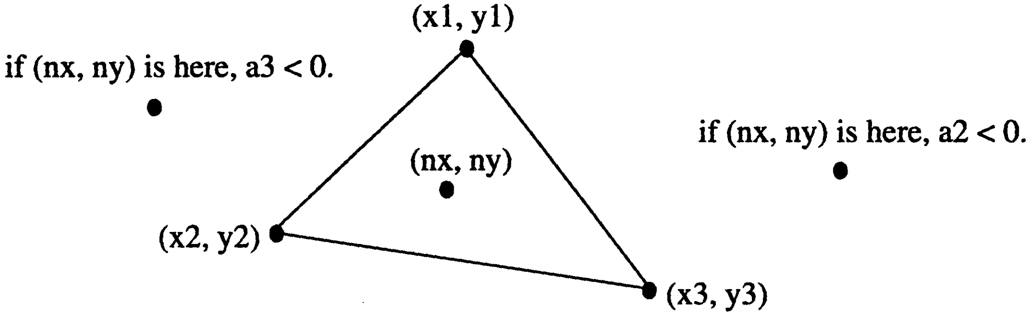


FIGURE 22. Three-point linear interpolation/extrapolation

The set of equations has a unique solution when the three points do not lie in the same line. We will call an "interpolation" when the new point is an interior of the shaded triangle if all of the a's is greater than zero and an "extrapolation" when one of the a's is negative; the coefficient correspoding to the vertex that is opposite to the new point will be negative [Dah74]. If the three points are colinear, we will use a two-point linear interpolation/extrapolation.

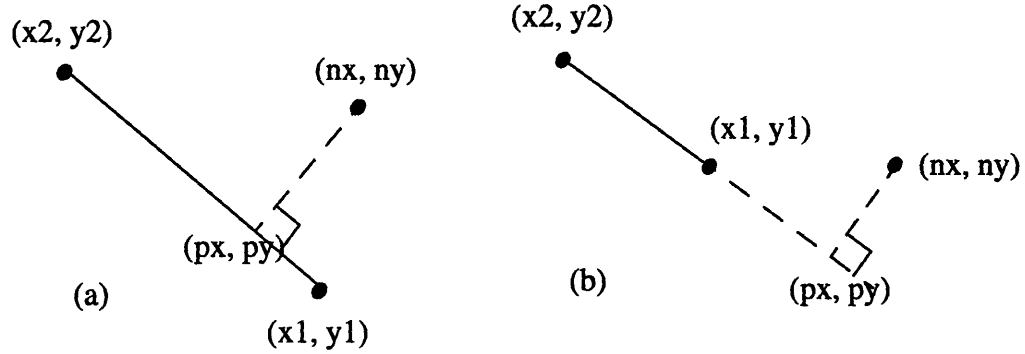


FIGURE 23. Two-point linear interpolation/extrapolation

The two-point linear interpolation/extrapolation (Figure 23) approximates the focus position nf at the new point (nx, ny) by the estimated focus position at the point (px, py) which perpendiculars to the line joining (x1, y1) and (x2, y2). The f-coordinate at (px,py) is estimated by interpolation of the f-coordinates of points (x1, y1) and (x2, y2). If the two points are not the same point, nf is uniquely expressed as:

$$nf = a1 * f1 + a2 * f1.$$

where, let the slope $m = (y2 - y1) / (x2 - x1)$.

If m is very big (i.e. the line joining (x1, y1) and (x2, y2) is almost vertical), $a2 = (ny - y1) / (y2 - y1)$,

else $a2 = ((((m^2 *x1 + m*(ny - y1) + nx) / (1 + m^2)) - x1) / (x2 - x1)$.

Some cautions are needed for extrapolation because the closets points on the focus map that will be used for extrapolation might be too close to each other and too far from the new point whose focus position needs to be estimated. Thus the extrapolation might be dangerous. The algorithm should make certain that the points are not too close to each other. The current implementation of interpolation and extrapolation simply chooses the closest points that are not closer to each other than the safe bound[1]. Also, it makes sure that these points are not far from the new point by limiting these points to be only within the interpolation bound. If the first two closest points are within the same safe bound of

---

[1] The size of the safe bound depends on the destined magnification.

each other, the further one from the new point will be discarded, not even used as the second closest point. This method of finding the closest points is used even for interpolation.

If at least two closest points to the new (x, y) point are not found within the interpolation bound of the new point, the algorithm searches for the three closest points which can be anywhere on the focus map. These three points are checked if they are qualified for interpolation/extrapolation. The rule for qualification in the current implementation is that the coefficients a's in the three-point interpolation/extrapolation formula are not less than -1. Thus if the new point (nx, ny) is in the bigger triangle of Figure 24, the algorithm can use the three points for extrapolation. If we decrease the limit of the coefficients to be no less than -0.5, the triangle will be smaller; thus, it is safer for extrapolation. If the new point is not in the triangle, the point that corresponds to the coefficient a_i that is the minimum (which is less than the limit of -1) is dropped. The remained two points are checked for two-point interpolation. The rule of qualification is similar to the one in the three-point interpolation which is the none of the coefficients a's is less than the limit of -1. If the remained two points are not qualified for extrapolation, the focus position of the closest

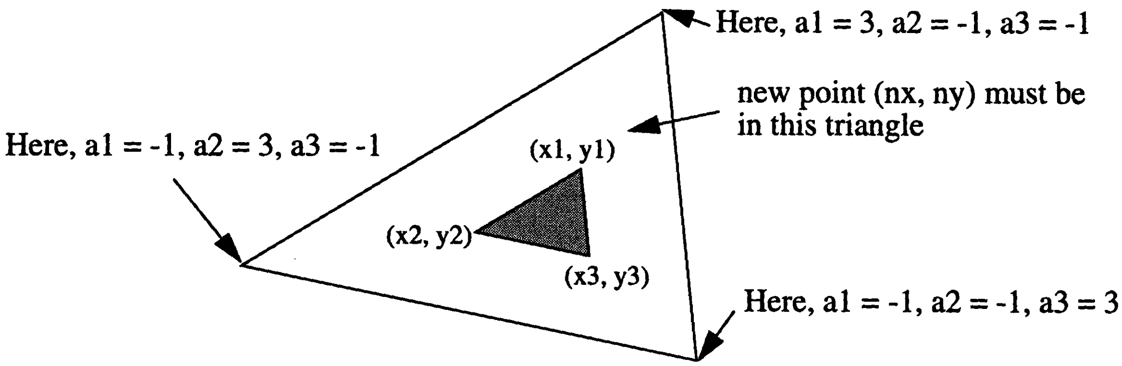


FIGURE 24. The rule for qualification of the three points used for extrapolation. The bigger triangle represents all the points that none of the coefficients in the three-point interpolation/extrapolation is less than -1.

one is used to represent the focus position of the new point.

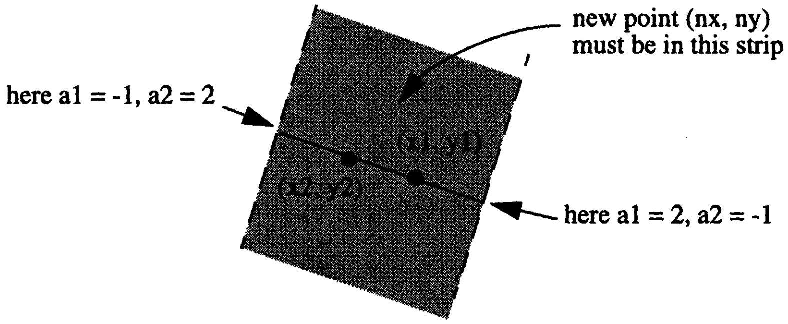


FIGURE 25. The rule for qualification of the two points used for extrapolation. The shaded strip represents all the points that none of the coefficients in the two-point interpolation/extrapolation is less than -1.

After the microscope has estimated the focus position, the microscope will move to the estimated focus position at the new (x, y) point by the "safe" move. The safe move means the microscope switches to a lower magnification (equal or smaller than 20x) first, moves, and tries to find the optimal focus position at 20x before switching to the destined magnification.

Finally, the overall picture of the algorithm for estimation of the focus position at the new (x, y) point is shown in Figure 27.

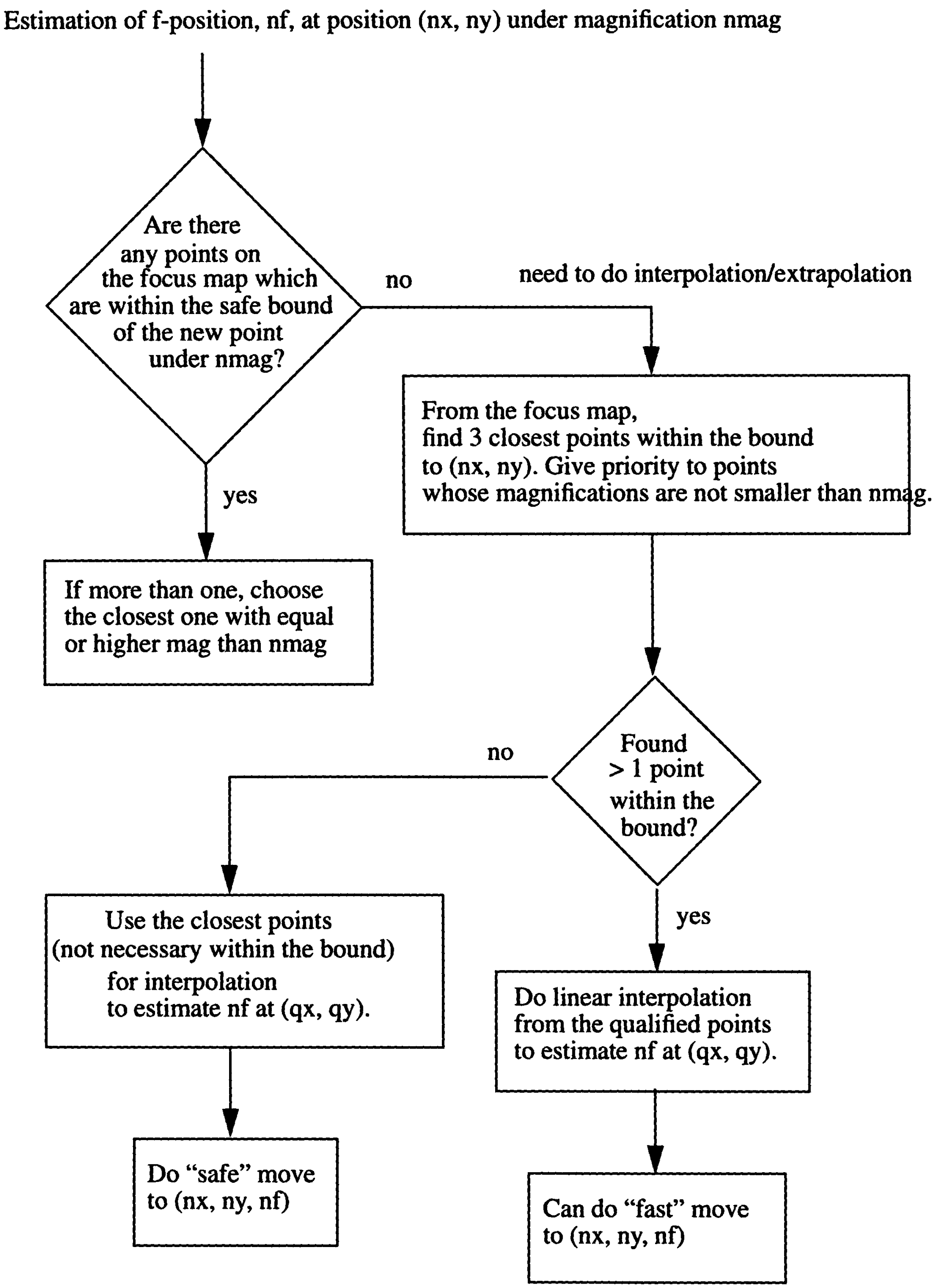


FIGURE 27. Algorithm for estimation of focus position at the new (x, y) position and new magnification, nmag.

7.3 Dynamically Handlings of Messages

After the server has connected to the microscope controller, the controller sends the initial value of the microscope state to the server. The server will send this state information to the first client. Then the server and the microscope controller wait for the client requests. The messages that the controller communicates with the server are listed in Table 8.

TABLE 8. Messages used between the microscope controller and the server

messages from Server to Controller	messages from Controller to Server
no reply.	command: "INIT" XPOS: current x-coordinate YPOS: current y-coordinate FPOS: current f-coordinate MAG: current magnification
command: "GRABFRAME" XPOS: requested x-coordinate YPOS: requested y-coordinate FPOS: requested f-coordinate MAG: requested magnification FOCUS: method of focus (auto/manual)	command: "GRABFRAMEREPLY" XPOS: current x-coordinate YPOS: current y-coordinate FPOS: current f-coordinate MAG: current magnification FOCUS: method of focus (auto/manual) used.
command: "ZOOMGRAB" XPOS: requested x-coordinate YPOS: requested y-coordinate FPOS: requested f-coordinate MAG: requested magnification FOCUS: method of focus (auto/manual)	command: "ZOOMGRABREPLY" XPOS: current x-coordinate YPOS: current y-coordinate FPOS: current f-coordinate MAG: current magnification FOCUS: method of focus (auto/manual) used.
command: "LOADNEWWAFER"	no reply.
command: "CALIB" DOIT: do calibration	command: "LOADNEWWAFERREPLY" XPOS: current x-coordinate YPOS: current y-coordinate FPOS: current f-coordinate MAG: current magnification FOCUS: method of focus (auto/manual) used.

The "SmartMove" routine containing the movement algorithm is called to process the client request "GRABFRAME" and "ZOOMGRAB." When it is done, the current position (corresponds to the requested position) and magnification of the microscope and the method of focus used is sent back with "GRABFRAMEREPLY" and "ZOOMGRABRE-

PLY." Note that when moving to the requested position, the microscope might miss by a few motor steps, thus it is a good idea to tell the clients what the actual position of the microscope is when the image is returned. At the server side, the "GRABFRAMEREPLY" and "ZOOMGRABREPLY" will be packed with a new image.

"LOADNEWWAFER" command prepares the microscope by calling the "ReadyForNewWafer" routine which will rotate the turret to 5x objective, lower the stage, and move it out. After the wafer has been loaded, the "CALIB" command is sent from the client, through the server to the controller. The "DoCalib" routine is called to handle the calibration of the offsets for this new wafer. If the user does not want to do the offset calibration, the stage moves back to the previous position before the stage is lowered. After loading a new wafer is completely done, an image will be returned with the "LOADNEWWAFERREPLY" command to the clients. Hence, the "LOADNEWWAFERREPLY" is similar to the "GRABFRAMEREPLY."

7.4 Communication with the Automated Microscope

The microscope controller sends commands to the automated microscope and waits to receive a reply back. The communication from the controller process which runs in OS/2 on a PC to the automated microscope is by the RS-232 serial communications. OS/2 provides a library of communication functions that simplify the handling of the communication port. It uses this communication port just as a regular file. After the communication port is opened, it returns a file handle that the controller can write, and read via this file handle.

The current settings for the RS-232 serial interface are: baud rate at 9,600, 8 data bits, no parity bits, 2 stop bits, no write time-out, and read time-out at 0 seconds. OS/2 provides the DosDevIOCtl function that can set and read these parameters of the communication port [Bla93]. For example, to set the baud rate, the controller must issue the commands:

```
usBaudrate = 9600;
DosDevIOCtl (hFile, IOCTL_ASYNC, ASYNC_SETBAUDRATE,
             (PVOID)&usBaudrate, sizeof(USHORT), NULL, NULL, 0L, NULL);
```

FIGURE 28. An Example function provided by OS/2 for setting the baud rate of the RS-232 communication

The "SerialSendAndReceive" routine in the controller program handles the communication to the microscope. It sends a string buffer and waits to receives a string buffer reply. Because the automated microscope uses the carriage return as the end of each command, this routine must ensure to put a carriage return at the end of every command. Each reply from the microscope is ended with a line feed or a carriage return. Because OS/2 handles the serial communications asynchronously, the controller does not know when the reply is arrived, and thus it has to wait for each reply by checking if it has yet received a complete reply.

Because we want the asynchronous serial device driver to handle only the communication port we are using, each sending must be encapsulated in a critical section[1]. The same for each receiving. In the future we might want to let the user to be able to stop the frame grab request. Then the controller will be interrupted, but we must ensure that the sending and replying of each command is complete with no interruption because there might be a confusion of which command matches to which reply if the sending and receiving reply is interrupted in the middle. Thus, we might want to encapsulate both sending a command and receiving a reply into the same critical section.

---

[1] See [Blain 93].

Chapter 8

Software Implementation of the Modified Microscope Server

The microscope server is responsible for handling client requests, relaying client requests to the microscope controller module and the video module, and book keeping of the clients using the microscope [Kao95]. The server has been mostly kept unchanged form the initial implementation, especially the book keeping functions for the clients, because most of the functions are still useful and important for the overall operation of the system.

The main modification in the server program is how the server handles the frame grab request. In the initial version of the microscope server, the server handles the client requests for grabbing new microscope images by waiting for an attendant to move the microscope and then notifying the video module to grab a frame from the video camera. In the current automated version, the server handles the client request for grabbing a new image by relaying the request to the microscope controller. The controller moves the microscope to the desired position and magnification and then instructs the server that the microscope is ready for a frame grab from the video camera. Other minor modifications are book keeping for microscope current position and magnification and more messages sent to the video module.

8.1 Maintaining Current States of Images and Microscope

Because the server must be able to bring up to speed any new clients that connect to the microscope during the middle of a conference inspection, the server maintains state structures that keeps the image information (see [Kao 95]). The added state information to each image structure is f-position of the microscope when the image is taken, method of focus (auto or manual), time when the image is taken, and an indicator of which image is the most recent one. The time stamp will be useful when recording images into a sequence in time. For the microscope state, we must keep the X-Y-F position, and the magnification. The microscope state is updated every time the server receives the "GRABFRAMEREPLY," "ZOOMGRABREPLY," and "LOADNEWWAFERREPLY" messages from the controller because the microscope is in the new position and magnification then.

8.2 Handling of Client Requests

The server uses the "new_message" routine to handle messages from the clients. The handling involves just relaying the messages ("GRABFRAME," "ZOOMGRAB," "LOADNEWWAFER," "CALIB") to the microscope controller. It calls the "new_message_scope" routine to handle the messages from the controller. For the "GRABFRAMEREPLY," "ZOOMGRABREPLY," and "LOADNEWWAFERREPLY," the routine updates either the GLOBAL or ZOOM state structure, and the microscope state structure. The "LOADNEWWAFERREPLY" is exactly identical to "GRABFRAMEREPLY." Then, it sends a character ('0xff') to the video module to initiate a frame grab and sends the information about color of the new image and whether to use a new color map for the new image. The video module transmits the GIF image back to server when it is done. Then, the server sends the reply back to all the clients. An example of the routine for "GRABFRAME" message is shown in Figure 1.

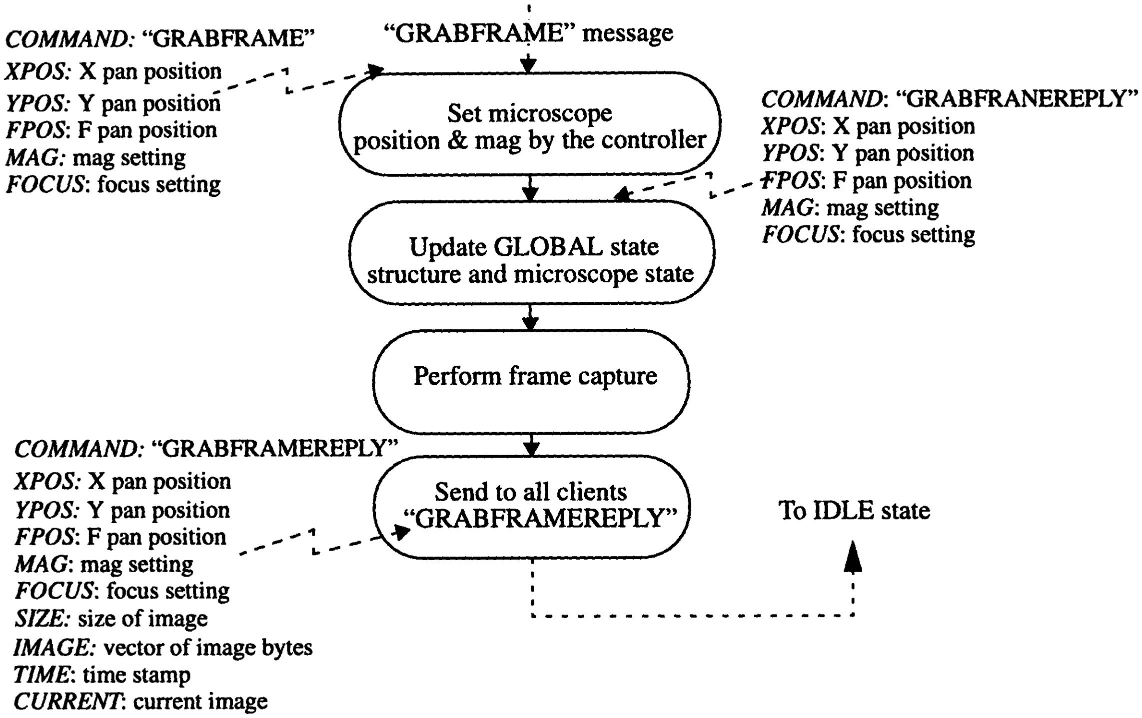


FIGURE 29. Model of execution after receiving "GRABFRAME" message from daemon (originating from Tcl/Tk Client).

8.3 Video Module

The video module is written as a stand-alone process because we still cannot integrate the server and the image capture routines into a single process. Its main responsibility is to use the Truevision Targa+ frame grabber board to capture image from the video camera and converts the TGA format (15 bits per pixel) image into the GIF format[1](8 bits per pixel) image. In the initial version, the video module stores the TGA format image to disk, then the server reads the TGA file and converts it to a GIF format. James Kao has modified the communication such that now the video module, which now includes the GIF format conversion, communicates with the server via socket but not by the CHAT messaging interface. Although the CHAT message passing interface is very easy to use, it needs to pack the large image into an object and then converts it to a buffer to send over a socket.

---

[1] See Compuserve Inc., "Graphics Interchange Format: Programming Reference."

Thus, it is slower than to just directly send the binary stream of image over the socket.

To avoid saving image to hard disk first before converting to GIF format, we have included the GIF format conversion into the video module. The video module reads directly the raw RGB pixels of the captured image and converts it to a GIF format by using the conversion function we have modified from the standard GIF format conversion program[1]. The "PPM_RGBToColorGif" routine converts the raw RGB pixel image into a color GIF format image, while the "RGBToGrayGif" routine converts the raw pixels to a gray GIF image.

For the conversion to a color GIF image, the raw pixels which have the maximum of 2^{15} colors must be quantized to the maximum colors of $2^8 = 256$ colors used by the GIF format[2]. The quantization which involves searching for the best colormap representation of all the colors in the raw image and quantizing the colors usually takes several seconds. As we have noticed that the colors of an area on the wafer are roughly the same, we do not need to calculate a new colormap every time we do conversion from the raw image to GIF format. Instead, we can use the color map from the previous conversion, and can generate a look-up color table that matches every colors to the colors in the colormap. This color table substantially saves the time to find the closest representation color in the colormap, although at the first time it takes longer to generate the color table. The user can change the color table easily. All these parameters for the conversion to color GIF image are the parameters of the "PPM_RGBToColorGif" routine. The function prototype of the

---

[1] The standard conversion programs are taken from XV. They are xv24to8.c and xvgif.c.

[2] Actually, we use only 200 colors.

PPM_RGBToColorGif routine is:

```
int PPM_RGBToColorGif (byte *GifArray, int Width, int Height, byte *ImageArray,
    int newcolors, byte ***ctable,
    byte *rmap, byte *gmap, byte *bmap, int UseOldCTable);
```

where GifArray is pointer to array of resulted GIF format image,
Width is the width of the image (which is 640 pixels, in this case),
Height is the height of the image (which is 480 pixels in this case),
ImageArray is pointer to array of the raw image pixels,
newcolors is the number of colors in the GIF image,
ctable is the lookup color table (3-D array),
rmap is the color map for red tone,
gmap is the color map for green tone,
bmap is the color map for blue tone,
UseOldCTable indicates if we want to use the old color table.

The user of the remote microscope can choose either gray or color image and choose to generate a new color table. These two pieces of information are packed within the "GRABFRAME" and "ZOOMGRAB" commands. They are extracted by the server and sent to the video module via the socket right before the video module grabs a frame.

Chapter 9

Software Implementation of the Modified Client User Interface

This chapter will describe the modification of the initial version of the client user interface. The modification is needed to enable the user to easily control the automated microscope. The description of the initial version of the client interface is in [Kao95]. The initial client interface has been altered to accommodate the focus axis coordinate of the microscope, the auto and manual focus, and loading a new wafer.

Each display (Window A and Window B) has its own state variables that keep the information of the microscope when the image is taken and the cursor position on the image. These variables are X-Y-F position, magnification, time stamp, and current X-Y coordinates of mouse position in each window.

TABLE 9. List of Variables related to the functions of the automated microscope

Variables	Brief Description
GX, GY, GF	(x,y,f) microscope coordinates of the middle point of the image when the image in window A is taken.
GM	magnification of the microscope when the image in window A is taken
GTIME	Time when the image in window A is taken.
GXlocal, GYlocal	Current X-Y coordinates of mouse position in window A.
ZX, ZY, ZF	(x,y,f) microscope coordinates of the middle point of the image when the image in window B is taken.
ZM	magnification of the microscope when the image in window B is taken
ZTIME	Time when the image in window B is taken.
ZXlocal, ZYlocal	Current X-Y coordinates of mouse position in window B.

TABLE 9. List of Variables related to the functions of the automated microscope

Variables	Brief Description
CURXPOS, CURYPOS, CURFPOS	Current (x,y,f) position of the microscope
CURM	Current magnification of the microscope
NEWXPOS, NEWYPOS, NEWFPOS	New (x,y,f) coordinates to position microscope in the subsequent grab
SETM	New magnification setting for subsequent grab
WHICHONE	Equal to "BIG" if the subsequent grab is displayed in window A, and "SMALL" if the subsequent grab is displayed in window B.
SETFOCUS	Focus setting for subsequent grab (either MANUAL or AUTO)
CTABLE	Equal to 1 if wish to create a new color table in subsequent grab
COLOR	Equal to 'c' if the subsequent is in color., and 'g' if in gray.
OUTLINEX, OUTLINEY	Current (x,y) position of the cursor (in pixels)

9.1 Handling Inputs from the Client Interface

When a user wishes to grab a new microscope image, he/she must specify the magnification and the X-Y-F pan position, the window in which the new image will be displayed, and manual or auto focus. If the user wishes to change magnification the value of the F-position will be ignored. The user can select microscope magnification by selecting one of the radiobuttons under the "SET MAGNIFICATION" label. When the user selects one of these button magnifications, the client process sets the SETM variable to the corresponding magnification and also automatically resizes the rectangle (by calling the "do_outline" routine) around the cursor outlining the region which will be zoomed into.

The user can set the new X-Y-F position of the microscope by entering the new coordinates in the SET POSITION box. This sets the NEWXPOS, NEWYPOS, and NEWFPOS variables to the corresponded values. The coordinates are in the unit of 0.1 microns. This position will be the position of the middle point of the next captured image. The user can also set the X-Y position by clicking the left mouse button on either one of the display windows. This puts an arrow cursor on that position on the image and also puts the corresponded X-Y microscope coordinates for NEWXPOS and NEWYPOS variables. The microscope coordinates are derived from the pixel coordinates of the Tcl/Tk display (see 9.2). If the new magnification is higher than the magnification of the image (GM or ZM), a rectangle outlining the region which will be zoomed into also appears with the cursor at its middle.

When selecting either window A or B to display the next image, the WHICHONE variable is set accordingly. The SETFOCUS variable has value of MANUAL if the manual focus button is chosen, and AUTO if the auto focus button is chosen. The user can also choose to use a new color map for the next image and choose either gray or color image. After all the setting of parameters for the next microscope image is ready, the EXECUTE button allows the user to start grabbing a new microscope image. The code for the EXECUTE button is shown in FIgure 30. It first sends a busy message to the server to be broadcasted to all clients. Then, it chooses the command either "GRABFRAME" if the WHICHONE variable is "BIG" or "ZOOMGRAB" if the WHICHONE variable is "SMALL." It packs all the parameter settings into the message and sends the request to the server.

The "load a new wafer" button allows the user to prepare the microscope stage for a new wafer by sending a "LOADNEWWAFER" request to the server. Another window is propped up and tells the user to click the OK button if he/she wishes to do offset calibration. The "CALIB" request will be sent to the server.

```
    set message “CURRENTLY EXECUTING MICROSCOPE CAPTURE. PLEASE WAIT”
    busy $message
    #addendum to pass busy signal to all users
    object_create daemon_obj
    object_slot_set daemon_obj COMMAND BUSY $OB_ASCII 5
    object_slot_set daemon_obj MESSAGE $message $OB_ASCII [expr 1+[string \
        length $message]]
    set buffer [ascbuf_from_object daemon_obj size]
    puts $accepted_socket $buffer
    object_rm daemon_obj

    puts “\n** Request sent: grab a frame**”
    object_create daemon_obj
    if {$WHICHONE == “BIG”} {
      object_slot_set daemon_obj COMMAND “GRABFRAME” $OB_ASCII 10
  } else {
      object_slot_set daemon_obj COMMAND “ZOOMGRAB” $OB_ASCII 9
}
    object_slot_set daemon_obj XPOS $NEWXPOS $OB_ASCII [expr 1+[string \
        length $NEWXPOS]]
    object_slot_set daemon_obj YPOS $NEWYPOS $OB_ASCII [expr 1+[string \
        length $NEWYPOS]]
    object_slot_set daemon_obj FPOS $NEWFPOS $OB_ASCII [expr 1+[string \
        length $NEWFPOS]]
    catch {object_slot_set daemon_obj CTABLE $CTABLE $OB_ASCII \
        [expr 1+[string length $CTABLE]]}
    object_slot_set daemon_obj COLOR $COLOR $OB_ASCII \
        [expr 1+[string length $COLOR]]
    catch {object_slot_set daemon_obj MAG $SETM $OB_ASCII [expr 1+[string \
        length $SETM]]}

    object_slot_set daemon_obj FOCUS $SETFOCUS $OB_ASCII [expr 1+[string \
        length $SETFOCUS]]

    set buffer [ascbuf_from_object daemon_obj size]
    puts $buffer
    puts $accepted_socket $buffer
    object_rm daemon_obj

    # reset CTABLE in the image quality button
    set CTABLE 0
```

FIGURE 30. Code that is executed when the EXECUTE button is pressed.

9.2 Coordinate Transformation

As mentioned earlier, there must be a coordinate transformation between the display X-Y coordinate system (which is in unit of pixels) and the X-Y global microscope coordinate system (which is in unit of 0.1 microns). This transformation is needed for conversion of the mouse position on either display windows to the microscope X-Y position. The display coordinate system (640x480 grid) is used to actually position the pointer or rectangular visual aids within the display canvases that hold the microscope images. The Tcl/Tk software has an access to the x-y pixel coordinate, where the upper left hand corner is (0,0) and the lower right hand corner is (639, 479).

Because the (GX, GY) corresponds to the global microscope coordinates of the middle point of the image in window A (same for window B), the display coordinates (320, 240) which are the pixel coordinates of the middle point on the display thus is equivalent to (GX, GY) in the global coordinates. From the measurement of the distance in 0.1 microns and the distance on the displays, it is estimated that at 100x magnification, 0.7520915[1] pixels are equal to 0.1 microns (or 1 in the global coordinate). Thus, the transformation from the display coordinate system to the global coordinate system is:

xwafer = GX + (x - 320) * (100 / (PIX * GM))

ywafer = GY + (y - 240) * (100 / (PIX * GM))

FIGURE 31. Transformation between the pixel coordinate system and the global coordinate system. PIX is the number of pixels on the display per 0.1 microns in the actual movement of the microscope. At present, PIX is 0.7520915. (case shown for point pixel (x,y) in window A, similar for Window B)

The relation between the number of pixels per 0.1 microns must be calibrated carefully. The way to calibration is to take a mark on the wafer, move the stage by a specified number of steps and measure the number of pixels the mark has moved.

---

[1] This conversion number is very important to the consistency of the two coordinates. This number still needs more calibration.

9.3 Handling Messages From Server

More messages have been added for communication between the server and the client. The first message from the server to the clients is "SETSCOPESTATE." This message is needed for the new client that just connects to the microscope. Actually, the information of the microscope state is already in the most recent image of the two images (window A or B) sent to the client after connection (the CURRENT variable in the GRABFRAMEREPLY and ZOOMGRABREPLY messages indicates which is the most recent image). However, for the first client, there is no image to send and thus it is better to explicitly set the microscope state.

The current and new states of the microscope are set to be equal to the values of XPOS, YPOS, and FPOS after the frame grab reply has been received by the client user interface program. The LOADNEWWAFERREPLY is identical to the GRABFRAMEREPLY because they both contains information for the image to be displayed in window A. The handling of the ZOOMGRABREPLY is essentially the same as that of GRABFRAMEREPLY. When a client is not in control, all the buttons and entries are disabled except the QUIT_APPLICATION, SAVE A, SAVE B buttons, to ensure that only one client is in control of the microscope.

Chapter 10

Possible Improvements and Future Work

Although the current automated remote microscope has successfully incorporated the automation of the microscope into the initial architecture, there are still possible improvements and future works which will make the remote microscope a more useful and effective inspection tool for both remote and local fabrication of the integrated circuits. More useful functionalities will be built upon the existing automation capability of the automated remote microscope.

10.1 Autofocus

Given the fact that the safety of the wafer mainly relies on the effectiveness of the autofocus algorithm to position the microscope stage properly and safely, it is important to have a reliable and robust autofocus algorithm that almost guarantees to find the true optimal focus position. Although the current autofocus algorithm that comes with the Beltronics automated microscope is quite robust and effective, more improvements are still possible.

To obtain optimal focus performance, the parameters of the autofocus algorithm which are number of samples and the fine and coarse distance in the fine and coarse searches must be optimized for each lens of the microscope because each magnification lens has different depth of focus [Bel96]. Although I believe that these parameters are chosen reasonably well for the current autofocus algorithm, more examination might be put into optimizing these parameters. The optimized scan range should not be too big since it takes more time to do the scan and not too small since it will not get to the optimal focus position.

Actually, the scan range the stage moves in F-direction during an autofocus attempt needs not to be fixed as it is in the current implementation. The autofocus algorithm might adjust the scan range according to the available information of the nearby known focus positions which are stored on the focus map. For example, if the optimal focus position at the new point is well approximated, the scan range for autofocus could be small and the algorithm would find the optimal focus position easily and fast. But if the optimal focus position is badly approximated, the scan range must be big. Thus a measurement of confidence in the estimation of the focus position at the new X-Y position is needed.

However, a more important worth of effort will be in finding a correct set of the offsets between different objectives. The correct offsets would increase the chance of successful autofocus. Because there might be several local maximums on the focus quality curves, the calculated offsets might be incorrect. Although these local maximums are actually the optimal focus positions for different levels of features on the wafer, there should be a way to calculate the offset with small offset error. The current algorithm to find offsets just takes average of these possibly incorrect offsets. In addition, this happens only once during the offset calibration before inspecting a new wafer. It would be better to dynamically calculate the proper offsets as the new optimal focus positions are founded, not only during the calibration, but throughout the inspection process.

Another possibility to improve the autofocus algorithm is to find a better way to measure the focus quality than the one used in the current autofocus algorithm (which integrates the output of the high pass filtering of the video signal within the window for 100 video frames). Although the algorithm resides in a dedicated hardware and thus has an advantage of fast calculation and access to video images, a more robust autofocus algorithm which must run on the PC may be developed in the future. Actually, there are several criteria to measure the focus quality[1]. Moreover, it is suggested that a focus criteria should measure the signal power of the middle frequency, since defocusing mainly reduces the frequencies around half the cut-off frequency of the optical system [Bod94].

10.2 Client User Interface

A major improvement for the remote microscope would be to have a client user interface written in Java. At the present, only the clients who have pre-installed Tcl/Tk and the client user interface software program can use the remote microscope. However, a future Java implementation of the client user interface will enable clients to use the remote microscope from anywhere on the internet without pre-installation of the microscope client interface software. The client interface will be downloaded into the client machine at the time it is needed. Java is portable across multiple platforms and hardware architectures. Java also provides a interactive graphical user interface which is supported by a rich library of applets and the multithreading built into Java environment. The Java Environment is described as a "new approach to distributed computing."[1]

Because the current client program has a well-defined messaging interface with the server program, changing the Tcl client interface to a Java client interface would not require any changes at the server to the microscope controller as long as the same messaging interface is kept.

10.3 Image Quality

The current remote microscope only supports a GIF format image which is 640 by 480 pixels at 8 bits per pixel[2]. The image captured from the Targa+ framegrabber is actually at 16 bits per pixel. Thus, a considerable colors are reduced. The possible improvement of the image might be employing the JPEG format instead of the GIF format. For storing full-color or gray-scale images of "realistic" scenes such as images from the CCD video camera, JPEG is superior to GIF.

In addition, JPEG provides lossy compression of images to reduce image size. This will reduce the transmission time to send images across the network. The user could, for

---

[1] See [Rus90].

[1] See http://java.sun.com/

[2] The colors are actually quantized to 200 colors

example, specify the compression ratio. This could be useful if the user does not need to see exact details of the image, but just wants to preview the general details to position the wafer to a region where he/she is really interested in. JPEG compresses images very well. For example, JPEG compresses an example image with a reasonable quality by 5 to 1 of the original GIF image. For preview quality, the compression ration can be 40 to 1. This certainly reduces the image size substantially. Thus JPEG would provide us with the reduction in the image size but at the same time with full-color capability at 16 bits per pixel[1].

During a manual focus, the microscope should allow the user to select if he/she wants to view the whole or clipped image. If the user knows where to focus, he/she might choose the area of the image that will be returned. This would save the delay for the new image.

10.4 Future Functionalities

Many future functionalities of the remote microscope will be able to built upon its automation capability. For example, there are some suggestions of having a capability to have a circuit layout of the wafer being inspected, and the user can mouse-click at the interesting point on the layout and will be able to see the corresponding position on the wafer. He/she can check the layout and the actual wafer. To be able to do this, the wafer must be calibrated with the layout correctly. In addition, the layout can be overlaid onto the image of the actual wafer. The Targa+ framegrabber supports this functionality [Tru92].

Another example would be that the user could specify the microscope to go to the preview mode which would automatically move the microscope stage to the next image in either X or Y direction. This would enable the user to preview the wafer until he/she finds an interesting region. The JPEG compressed image would be very useful in this preview functionality.

---

[1] See http://www.cis.ohio-state.edu/hypertext/faq/usenet/jpeg-faq/top.html for information about JPEG image compression.

10.5 Authentication

Before the automated remote microscope is released for on-line use, there must be an authentication mechanism that allows only registered clients to connect to the microscope. This might be in a form of password checking. Right now, there is no authentication mechanism in the remote microscope system.

Chapter 11

Conclusion

The MIT automated remote microscope allows a user to remotely control and view a distant automated microscope over the internet. The remote automated microscope which is the initial remote microscope incorporated with an automated microscope provides a convenient, precise, and computer-controlled operation of the microscope that needs no attendant to control the microscope. Although the remote microscope was primarily designed to be used as a remote inspection tool during a remote fabrication of integrated circuits, it is actually a general tool that will allow any user with an access to an X-window UNIX workstation to view any type of specimen under the microscope. Thus, it could be used as, for example, remote learning tool.

The advantage of the MIT remote microscope is that it is versatile; it operates over internet with a natural user interface to the microscope. A user can set X-Y-F position and magnification of the microscope and request an autofocus for a new image. Although the current cost of the automation of the microscope is quite expensive, the additional cost at the microscope client is virtually free because the the interface is a software package written at MIT. Thus, we hope that this remote microscope will still be a valuable inspection and collaboration tool for the semiconductor filed in the future.

The incorporation of the automated microscope is not a difficult task as the architecture of the initial remote microscope system uses the CHAT messaging interface. However, the autofocus algorithm and the movement algorithm which must concern about the safety of the wafer and the objectives have proved to be complicated and needed a careful consideration.

Although the current automated microscope can be already used as an effective inspection tool, its value will multiply in the future as the computer-controlled operation of the microscope will provide more useful functionalities that require an exact coordinate positioning and computer-controlled movement of the microscope. Examples of these functionalities include storing a sequence of images of the wafer during processing steps for later inspection, scanning the wafer to find areas of interest, computer-based analysis of microscope images such as finding defects on the wafer, and matching and comparing between a circuit layout and the actual wafer after processing steps. These possible applications of the automated remote microscope will prove to be valuable even for inspection within local facility. As new valuable functionalities built into the automated remote microscope, we believe that it will be used effectively as a remote inspection tool and a collaboration tool to help researchers work together during remote semiconductor processing in the coming future.

References

[Kao95] J. Kao, "Remote Microscope for Inspection of Integrated Circuits," S.M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 1995.

[Car95] J. Carney, "An Implementation of a Layered Message Passing System," S.M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1995.

[Dah74] G. Dahlquist, A. Bjorck, translated by N. Anderson, Numerical Methods, Prentice-Hall, New Jersey, 1974, section. 7.7.3.

[Bod94] F. Boddeke, L. van Vilet, H. Netten, I. Young, "Autofocusing in Microscopy based on the OTF and Sampling," *Bioimaging*, 1994, p. 193-203.

[Rus90] J.C. Russ, Computer-Assisted Microscopy: The Measurement and Analysis of Images, Plenum Press, New York, 1990.

[Bla93] D. Blain, K. Delimon, J. English, Real-World Programming for OS/2 2.1, Sams Publishing, Carmel, IN, 1993.

[Bel96] Beltronics, Operation Manual for Beltronics Automated Microscope System, May 1996.

[Ous94] J.K. Ousterhout, Tcl and Tk Toolkit, Addison Wesley, Reading, MA, 1994.

[Tru92] Truvision Corp, Specification Sheet for Targa+ Graphics Engine, 1994.

[Sun95] Sun Microsrosystems, “The Java^{TM} Language: A White Paper,” 1994, 1995.

Appendix A

Using the Automated Microscope Software Package

In order to set up the Targa+ 16/32 framegrabber board, the user must type "tmode 1" after each time the PC is turned on. Then the user must open three windows which will be used for running the video module program, the microscope controller program, and the server program. These programs must be ran in this order respectively. In the future, these three programs might be included into one program.

At the window containing the executable file for the video module program, type "video [video port number]" to start the video program. The video port is the port the video module is waiting for connection from the server. Then, at another window for the microscope controller program, the user should type "scope [controller port number]" to start the program. At this time, the microscope will be doing the initialization routine. After it has finished, it tells the user that the user can start the server program in another window now. The server program is called by typing "server [address of video module] [client port number] [Video port number] [Controller port number]." The address of the video module (which is the same of the address of the controller) is the internet address such as "oolong.mit.edu." It will display that it is ready for client connections.

Setting up each microscope client involves running the Tcl/Tk client interface and the associated daemon process. Currently each of these programs needs to be run in it's own xterm window, mainly in order to display debugging output, although in future releases of the software we may spawn the daemon process directly from the Tcl/Tk client program. The Tcl/Tk client interface process needs to be run first because it was designed to wait for an incoming daemon connection. The Tcl/Tk client is executed by first running "wish" (a windowing shell that includes Tcl/Tk commands), and then typing "source package.tcl" at the prompt. The "package.tcl" file is a short script that loads several files that are needed to run the client interface, and also has a line that sets up the port number to be used by the daemon. After the Tcl/Tk package is loaded, then the client daemon should be run in a different window by typing "client [HOSTNAME] [PORT]." The hostname is simply the name of the computer that the Tcl/Tk client and daemon are running, and the port number is simply the one specified in the "package.tcl" script. Immediately after the daemon is run, the Tcl/Tk client interface should display a window for the user to input the server address and port number.

All three programs (the video program, the controller program, and the server program) use the CHAT message passing tool and thus they must have access to the files in the CHAT tool. The versions of the CHAT tools used were the UNIX CHAT tools ported to OS/2.The CHAT message passing tool consists of the following files:

File	Description
basic_sockets.h	OS/2 CHAT basic sockets header file
basic_sockets.c	OS/2 CHAT basic sockets routines to provide socket connections.
com_handler.h	OS/2 CHAT communication handler header file
com_handler.c	OS/2 CHAT communication handler routines to provide event handling etc.
object.h	OS/2 CHAT object header file
object.c	OS/2 CHAT object routines to provide object message data abstraction

TABLE 10. CHAT message passing tool

The following files are needed to compile the video module software on the PC using IBM Visual C++, which should produce a "video.exe" executable after running the makefile. It utilizes the Targa+ compatible toolkit for OS/2. The following table describes the code elements more carefully:

File	Description
Makefile	Makefile to produce "video.exe"
capcolor.c	Main code for image capture program
ToGif.h	Header files for GIF encoding routines
gifencod.c	Gif encoding routines
ToGrayGif.c	Grey scale conversions
ToPPMColorGif.c	Color conversions
xv24to8.c	Conversion utilities (including quantization utilities)
tardev.h	Targa+ Compatible toolkit header
targaos2.lib	Targa+ Compatible toolkit library routines

TABLE 11. Video module package

The microscope controller program, which runs after the video program and before the server program, is also compiled by the IBM Visual C++ and produce a "scope.exe" executable. The controller consists of the following files:

File	Description
Makefile	Makefile to produce "scope.exe"
scope.c	Main code for the microscope controller program
smart.c	Movement algorithm routines
helper.c and helper2.c	Movement routines that communicate with the automated microscope and autofocus routine.
calib.c	Offset calibration routines
hash.c	Hash table routines
serial.c	OS/2 RS-232 interface routines

TABLE 12. Microscope controller package

The following files are needed to compile the server software on the PC using the IBM Visual C++, which should produce a "server.exe" executable after running the makefile. This following table briefly describes each component.

File	Description
Makefile	Makefile to produce "server.exe"
record.h	Remote microscope record-keeping package header file

TABLE 13. Server package

File	Description
record.c	Remote microscope record-keeping package routines.
server.h	Remote microscope server header file
server.c	Main remote microscope server code

TABLE 13. Server package

The Tcl/Tk code consists of several scripts that need to be loaded into the wish shell, but. the user only needs to source the "package.tcl" file. All of this code (including the CHAT routines) were written using Tcl. Although currently the client software consists of files that need to be sourced into a wish shell, later versions of the code most likely will be launched directly from the UNIX prompt.

File	Description
package.tcl	Short script that sources all necessary files into wish shell
basic_sockets.tcl	Tcl CHAT basic sockets routines to provide socket connections.
com_handler.tcl	Tcl CHAT communication handler routines to provide event handling etc.
object.tcl	Tcl CHAT object routines to provide object message data abstraction
windowinit.tcl	Remote microscope client interface widget initialization
clientele.tcl	Remote microscope client interface main program (dynamic)

TABLE 14. Tcl/Tk client interface package

Finally, the client daemon was implementing using C and the UNIX version of the CHAT utilities with GCC. The client daemon is always run in conjunction with the Tcl/Tk client interface.

File	Description
Makefile	Makefile to produce "client.exe"
basic_sockets.h	CHAT basic sockets header file
basic_sockets.c	CHAT basic sockets routines to provide socket connections.

TABLE 15. Client daemon package

File	**Description**
com_handler.h	CHAT communication handler header file
com_handler.c	CHAT communication handler routines to provide event handling etc.
object.h	CHAT object header file
object.c	CHAT object routines to provide object message data abstraction
client.c	Remote microscope daemon main program

TABLE 15. Client daemon package