

The Software Interface to the 3D-Force Microscope

David Marshburn*
Chris Weigle*
Benjamin G. Wilde*
Russell M. Taylor II*

Kalpit Desai*
J. K. Fisher†
Jeremy Cribb‡

E. Timothy O'Brien†
R. Superfine‡

Dept. of Computer Science
University of North Carolina at
Chapel Hill

Dept. of Biomedical Engineering
University of North Carolina at
Chapel Hill

Dept. of Physics and Astronomy
University of North Carolina at
Chapel Hill

ABSTRACT

We have developed a real-time experiment-control and data-display system for a novel microscope, the 3D-Force Microscope (3DFM), which is designed for nanometer-scale and nanoNewton-force biophysical experiments. The 3DFM software suite synthesizes the several data sources from the 3DFM into a coherent view and provides control over data collection and specimen manipulation. Herein, we describe the system architecture designed to handle the several feedback loops and data flows present in the microscope and its control system. We describe the visualization techniques used in the 3DFM software suite, where used, and on which types of data. We present feedback from our scientist-users regarding the usefulness of these techniques, and we also present lessons learned from our successive implementations.

CR Categories: H.1.2 [Models and Principles]: User/Machine Systems; C.3 [Special-purpose and application-based systems]: Real-time systems, I.3.7 [Three-Dimensional Graphics and Realism]: Virtual Reality, J.3 [Computer Applications Life and Medical Sciences]

Keywords: applications of visualization, multimodal visualization, haptics, force, scientific visualization, interactive graphics, virtual worlds, microscopy.

1 INTRODUCTION

A novel microscopy manipulation system based on magnetic beads has been developed to address the limitations of atomic-force microscopy and laser-trapped bead manipulation systems. A prototype of the 3DFM hardware was described by Superfine et al [17]. An upcoming publication describes the microscope hardware and the physics of its operation [6]. An earlier book chapter, which describes several microscope control systems, includes a brief description of the software system for the prototype 3DFM, along with the lesson learned from the time-multiplexing force display [21]. This paper describes the visualization and control subsystems in detail and provides several lessons learned from the iterative design process with the developers of the microscope hardware and our scientist-users.

* Email: {marshbur, weigle, wilde, taylorr, kvdesai}@cs.unc.edu

† Email: {jgator, jcribb}@email.unc.edu

‡ Email: {obrient, rsuper}@physics.unc.edu

A 3DFM tracks micron-scale paramagnetic beads in a specimen and applies forces to those beads by generating magnetic field gradients. This microscope enables experiments not possible with other types of force microscopy, such as atomic-force microscopy, by placing the probe – in this case, a bead – inside the specimen with no mechanical contact between the probe and the rest of the instrument. Furthermore, the forces generated are specific to the paramagnetic beads, as opposed to laser-trap systems, in which the forces affect all particles encountering the laser trap. These differences enable the microscope to probe freely-diffusing beads in fluids [11], beads attached to beating cilia on lung-cell cultures [14], beads attached to DNA, and potentially beads moving within or underneath living cells.

The benefits found from 3D graphics-plus-force display in our earlier nanoManipulator system (see [19], [5] and [18]) led us to design a corresponding interface for the new 3D force microscope. Our aim is to place a scientist virtually within the specimen, in control, while experiments are happening in order to enable rapid hypothesis formation, data collection and analysis while the specimen is mounted in the 3DFM.

Section 2 describes the 3D-force microscope in sufficient detail to understand our software development for the device. Section 3 describes the suite of software as a whole as well as several of the component pieces of software in detail. Section 4 describes the largest piece of software in the system, the so-called 3DFM-UI, along with the visualization results. Section 5 describes some ideas for future development of the 3DFM software suite.

2 THE 3D-FORCE MICROSCOPE

Figure 1 shows the 3DFM composed of three subsystems: magnetic, tracking and imaging. Most of the hardware of these subsystems is mounted on a commercially available inverted optical microscope. See Fisher et al [6] for details of the microscope hardware.

The magnetic subsystem applies magnetic field gradients to the specimen volume, thereby creating forces on paramagnetic-bead probes, in a manner similar to Bausch [2,3]. A custom magnetic stage with six individually driven coils creates magnetic flux. The coils are magnetically coupled to machined magnetic poles, which draw the magnetic flux from the coils to the specimen and create the magnetic field gradients that drive the beads. New poles are produced in custom configurations to optimize forces for each experiment, and these poles can be positioned both above and below the specimen to produce forces in three dimensions. The coils can be driven at up to 10 kHz through a custom voltage-to-current

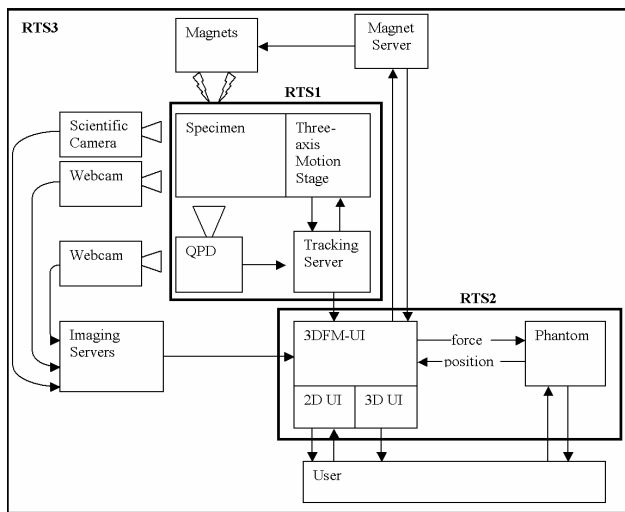


Figure 1: The diagram shows the distributed nature of the 3DFM, including the three separate real-time control systems that are embedded within the instrument. The first system, RTS1, includes tracking and stage control. The second real-time system loop, RTS2, provides stable haptic feedback to experimenters specifying forces by hand. The third real-time system, RTS3, includes all of the equipment in the system *and* a scientist-user. A scientist is presented the data from the disparate parts of the system and then determines some action to take. By moving the haptic device on the Phantom-control computer, for instance, a scientist causes commands to be sent to the magnet-control computer.

transimpedance amplifier (because of this amplifier, we tend to think and talk about controlling the voltage to the coils, even though the current creates the magnetic field and flux).

The tracking subsystem tracks the position of a bead with 5-nanometer resolution in three dimensions using laser-scattering techniques identical to Allersma et al [1] and Gittes et al [8,9]. A laser beam is focused on a bead within the specimen, scattering the laser light in an interference pattern onto a quadrant photodiode (QPD). As the bead moves relative to the laser beam, the pattern of scattered light on the QPD changes. From the change observed by the QPD, the system calculates the three-dimensional motion of the bead [4]. The QPD is sampled, and the bead displacement estimated, at a rate of 10 kHz. As bead motion is detected, the tracking subsystem continually adjusts a three-axis motion stage on which the specimen rests to keep the tracked bead centered within the stationary laser beam. Commands are sent to the stage at a rate of 2-4 kHz.

The imaging subsystem observes and records optical images of the specimen and microscope using several different cameras. A Roper CoolSnapHQ scientific camera captures video at a rate up to 20 mega-samples per second with a maximum resolution of 1932 x 1040 pixels and 12 bits per pixel. The camera can detect several fluorescent spectra in addition to visible light. For specimens with rapidly moving components, a Pulnix PTM-6710 scientific camera produces video data at a rate up to 120 frames per second at a resolution of 648 x 484 pixels and 8 bits per pixel. Finally, cheap, web-cam-quality cameras view several parts of the system for diagnostic purposes and tracking lock-on; one images the back of the QPD to observe the diffraction of the laser around the bead, and another images the specimen along an optical path split from that of the high-quality cameras.

2.1 On the Data Generated by the 3DFM

Each subsystem described above generates a different type of data. All data are logged using features available in the Virtual Reality Peripheral Network (VRPN) library [20]. A microsecond-accurate time stamp is associated with each datum.

The magnetic subsystem produces signals that control the energization of the magnet system's coils. Every time the voltages across one or more coils are changed, the voltages across all six coils are recorded.

The tracking subsystem operates in two modes of data collection. In both modes, a number of parameters important to initializing and maintaining tracking are logged, including laser intensity and feedback gains. In the low-bandwidth mode, the 3D position of the stage and the four QPD values are stored at the stage command rate of 2-4 kHz. In the high-bandwidth mode, 3D stage position plus QPD values are stored at the full QPD rate of 10 kHz. All of this information is logged to a single file.

The imaging subsystem also produces a significant amount of data. For data integrity, full-frame, uncompressed images are saved.

2.2 On the Distributed Nature of the System

All subsystems require real-time control. The tracking subsystem requires the full processing power of a CPU to interpret QPD signals into bead positions at 10 kHz. The tracking and imaging subsystems require high-bandwidth access to storage. The 3D graphics and haptic loops (see section 4) each require a dedicated CPU.

Given the considerable data rates and real-time requirements for controlling all pieces of microscope hardware, we chose to distribute this load over several computers. The control system for the 3DFM is composed of seven processors in five cooperating computer systems, each of which performs one or a few dedicated real-time computing tasks. These machines form a distributed system (see Figure 1) that enables a user to run the microscope from a single computer, which communicates with the other computers over a switched, 100-Mbit/second Ethernet.

Techniques to run a single program on more than one networked computer are well known; we use a client-server model to establish connections between hosts and a peer-to-peer message-passing protocol within VRPN. All of the network communication in our system takes place using VRPN.

The use of dedicated subsystem-server computers also enables flexibility: in a laboratory with several 3DFM installations, the user interface can be connected to the servers of any 3DFM in the laboratory for one experiment while traditional 2D or optical stack microscopy experiments can be conducted simultaneously on the other 3DFM systems.

3 THE 3DFM SOFTWARE SUITE

Each of the several programs that run the microscope is described separately below with lessons learned following. The tracking server, magnet server and scientific-video server are all written in C++. The software that brings all of these systems together is described in Section 4.

3.1 Tracking Server

The tracking server reads excitation values from the QPD as well as position values from the three-axis position stage using a National Instruments analog-to-digital (A/D) board. It

computes the displacement of the bead from the laser beam center and sends commands to the stage using a National Instruments digital-to-analog (D/A) conversion board to re-center the bead.

The bead is initially centered using a 3-knob dial box and tracking is initiated by releasing a spring-loaded foot pedal. The three dials control the x -, y -, and z displacement of the stage. A scientist adjusts the dials to move the stage until the to-be-tracked bead appears to be centered in the laser beam. The foot pedal is then released, and the tracking server locks on to the bead. For details of the tracking and lock-on algorithms, see Desai [5].

The tracking server reports bead positions using the `vrpn_Tracker` class from the VRPN library at 60 Hz. This rate is much slower than the stage-command, stage-response or QPD-read rates; it was chosen to be compatible with real-time display of bead position. The tracking server also saves data to VRPN-format log files, as discussed in section 2.1.

3.2 Magnet Server

The magnet server receives commands for the magnetic subsystem and calculates appropriate outputs to apply to the magnetic coils. It reports magnet-coil excitations using the `vrpn_Analog` class from the VRPN library at whatever rate the coils are changed. Requests to the server are of two types: coil-voltage specifications and force specifications.

For coil-voltage requests, the magnet server passes excitation values directly to the appropriate channels on a D/A board. Force specification is more complicated due to the fact that, for a given force request, some pole configurations allow multiple solutions while other pole configurations allow no solution. Furthermore, this problem does not have a readily accessible, closed-form solution for varying pole geometries. See Vicci [23] for details of the magnetic force model. Finally, we desire the transitions in coil excitation to be relatively smooth in time. We describe three solutions used in the force-specification mode in the following paragraphs.

Our first and simplest solution was to energize the single pole closest to the requested force direction. When the magnet server received a force request, the pole with the highest normalized dot product between the desired force direction and the pole direction was energized with a voltage proportional to the magnitude of the requested force vector. This solution was developed for a four-pole geometry on the first 3DFM prototype, which was not able to pull in arbitrary directions, and is referred to as the *force-projection solution*. The solution produces excitations varying smoothly in time except at the borders (in orientation space) between pole directions. It does not pull in the direction requested.

Our second solution was to multiplex in time across the poles closest to the requested direction. For this solution, the normalized dot product between a requested force vector and each pole direction is computed, and the three poles most closely aligned with the requested force direction are selected. We repeatedly cycle through those poles, energizing each for a short time (such as 100 ms) in succession and in proportion to the normalized projection of the requested force vector onto that pole direction. This *time-multiplexing solution* is approximate but produces a net force in the direction requested when averaged over time periods of tenths of a second or longer. It is relatively smooth when averaged over time but discontinuous at short time scales on the order of the cycle time.

Our third and final solution is an optimization routine that solves for coil excitations satisfying a specific force request. This *force-solver solution* relies on a closed-form solution to the “forward” problem of calculating force vectors given coil excitations. A model based on magnetic monopoles at specified pole locations admits a rapid, closed-form solution to this forward problem. The rapid, closed-form, forward solver together with a simplex-downhill traversal of coil excitations provides the “inverse” solver for this force-solver solution. The initial voltage estimate is a positive voltage on the pole closest in orientation to the desired force. The solver produces the set of excitation voltages whose force is within a threshold of the desired force, or returns the closest force that is attainable. Whenever the requested force changes gradually, the previous solution is used as the starting location. This tends to select the particular solution whose voltages differ least from recent excitations. It produces forces in the desired direction with smooth transitions, whenever they are attainable with the given pole configuration.

3.3 Scientific-Video Server

The scientific-video server transmits images from a scientific camera to the computer running the 3DFM-UI using the `vrpn_Imager` class from the VRPN library. The server also stores time-stamped copies of the image data for later analysis or replay with the 3DFM-UI. We currently have servers for Roper Scientific cameras, Pulnix cameras with EDT boards, SPOT cameras, and DirectX cameras.

The long exposure times required for fluorescence-microscopy experiments limit the image data rate to about 5 monochrome frames per second. Switched 100Mbit Ethernet connections between the computers easily accommodate this data rate. Brightfield experiments using the Pulnix camera produce up to 120 high-resolution frames/second, which is too fast for human perception even if it were possible to display or transmit at that rate.

To limit the amount of data sent, we currently use a “client quench” method whereby frames of video are only sent when requested by the client application. In this method, the number of outstanding frames is limited to one; intervening frames are not sent, preventing latency due to buffering and enabling responsive updates at high capture rates. We plan two other methods to limit data transmission to reasonable bandwidths and display rates. The first would be to send only a sub-region of the video at displayable rates. The second approach, dubbed “security-camera mode,” would store data continuously in a circular buffer on the server until a user sees an interesting event. In a step away from the real-time nature of the system, the full resolution video would then be sent to the client for replay at standard frame rates or in slow motion, time-aligned with other data sources from the experiment.

3.4 Successes and Failures

3.4.1 Eyes-free operation

Tracking initiation requires two things: alignment of a to-be-tracked bead within the stationary laser beam and a trigger to initiate tracking. The earliest implementation required users to type in x - y - z displacements for the motion stage. Slider bars were soon added as another means to move the stage, in addition to the precise numeric entry boxes. With these, users initiated tracking by clicking on one of several radio buttons with a mouse. Typing stage displacements was awkward; moving sliders was only a little better. Aiming at a radio

button with the mouse was also difficult. The real problem is that our scientist-users need to be able to watch the image of the laser beam as diffracted by the bead; they need eyes-free adjustment of the stage and eyes-free triggering.

We added a spring-loaded foot pedal to replace mouse clicks as a means to trigger tracking. Initially, depressing the pedal initiated tracking, but it was still necessary to use the mouse to stop tracking. This was still not optimal because it often takes several tries to initiate tracking. Users had to remove eyes from the QPD display, foot from the foot pedal and click on a radio button to stop tracking. One user also reports that it was difficult to find the pedal with his foot after having moved to use the mouse. Now, depressing the pedal stops tracking, and releasing the spring-loaded pedal initiates tracking. To repeatedly try to initiate tracking, the foot hovers over the pedal, and the eyes stay on the bead image. We then replaced the sliders with a set of dials to move the stage, with one dial for each of the x-, y- and z axes. Now, the hands stay on the dials while the eyes stay on the image.

This is still an Etch-a-Sketch interface, with artificially independent axes. To make the interaction more natural and intuitive, we added a Logitech Magellan to the tracking server. This is a strain-sensitive device that acts as a velocity control. We used only the three translation axes of this six-degree-of-freedom input device.

3.4.2 “Intuitive” solutions

To our surprise, our scientist-users strongly prefer the dial interface to the Magellan! They report difficulty moving along one axis without accidentally moving along the other two axes; in particular, accidental movements in x or y are confounded with motion in z. Accustomation to the Magellan mitigates this; projecting the Magellan’s displacement onto the nearest major axis would remove the problem. The second complaint was that motions were not repeatable. Easily repeatable motion turned out to be very important; our scientist-users want to be able to observe the effect of small displacements by going back and forth between two positions. Even though the Magellan returns to its rest position when released, it is a velocity-control device and reversible motion is difficult to achieve.

The intuitiveness of solution turned out to be underconstrained. In particular, all of the previous methods of moving the stage (text-entry boxes, slider, dials) explicitly controlled position, so the issue of position vs. velocity control had never arisen. In addition, the perceived benefit of free-form 3D motion turned out to be a drawback for the particular task for which the Magellan was intended.

3.5.2 Data integrity

Video data from the system is our most difficult data storage problem: video data arrives at 100 times the rate of all other data sources in the system combined. We would like to use common video and image compression methods with high compression ratios in order to bring the video data rates in line with our other data sources. Unfortunately, such methods are lossy. Our scientist-users are uncomfortable, at best, with schemes in which the only recorded data are manipulated versions of raw data, especially when it is not possible to reconstruct the raw measurements. Because analyses are done on the video data we collect, we store uncompressed, full-frame video and warn our users to be very frugal in the amount of video they store. We also have several large, portable drives that are used to temporarily store video data for review.

4 THE 3DFM-UI

The 3DFM-UI collects and coordinates the display of all data from an experiment and enables control over the microscope from a single computer. This synthesis gives the 3DFM much of its power and ease of use. In the following sections, we discuss the program’s visual and haptic components.

The 3DFM-UI is written in Java. We use the VRPN library, with its native C++ code wrapped in Java, to communicate with the servers described in section 3. The 2D parts of the interface are written with Java’s Swing toolkit. The 3D parts of the interface are written using the Visualization Toolkit (VTK) in its Java-wrapped form.

4.1 The Scope of the 3DFM-UI

The 3DFM-UI provides:

- Control of magnet excitations via programmed force trajectories or in a free-hand fashion.
- Real-time visual display of the path taken by the tracked bead during an experiment.
- Visual indication of the force acting on the tracked bead.
- Display of 2D video captured during experiment, aligned with the bead-tracking data.
- Multiple views of the several data types.

A scientist controls forces on beads in a specimen using the Phantom force-feedback controller or via predefined functions. The scientist’s intentions are mapped to control commands and force feedback appropriate to the task is displayed. Although the haptic interface provides the most flexible and immediate control for exploration, the automatic and semi-automatic modes are the ones most often used for experiments that search parameter spaces or that include repeated studies on multiple specimens.

4.2 The 2D Interface

As figure 2 indicates, the 2D interface consists of standard widgets found in any user-interface toolkit. We also developed three novel widgets to display data peculiar to the 3DFM:

4.2.1 Magnet pole monitor bars and controls

On the left side of the 3DFM-UI are several color-bar displays that show the voltage currently applied to each magnet-drive coil. Each bar is increasingly filled as the voltage on the corresponding coil rises. These are most useful when the magnets are controlled in a coil-specification manner. They also proved to be a useful “sanity check” when the magnets are operated in a force-specification manner.

4.2.2 Bead-step Histograms

The histograms on the right side of 3DFM-UI show binned counts of observed bead velocities in X, Y and Z. There are well-known velocity distributions for freely diffusing particles. Based on the forms of these histograms, our scientist-users make inferences about the bead motion, the effects of magnetic actuation, and even specimen preparation. For instance, a normal distribution in X and Y with a very narrow distribution in Z can indicate that the tracked bead is stuck on the specimen chamber floor.

4.2.3 Stripchart

Stripcharts are fairly common in scientific data collection and instrumentation. However, we found no available Java

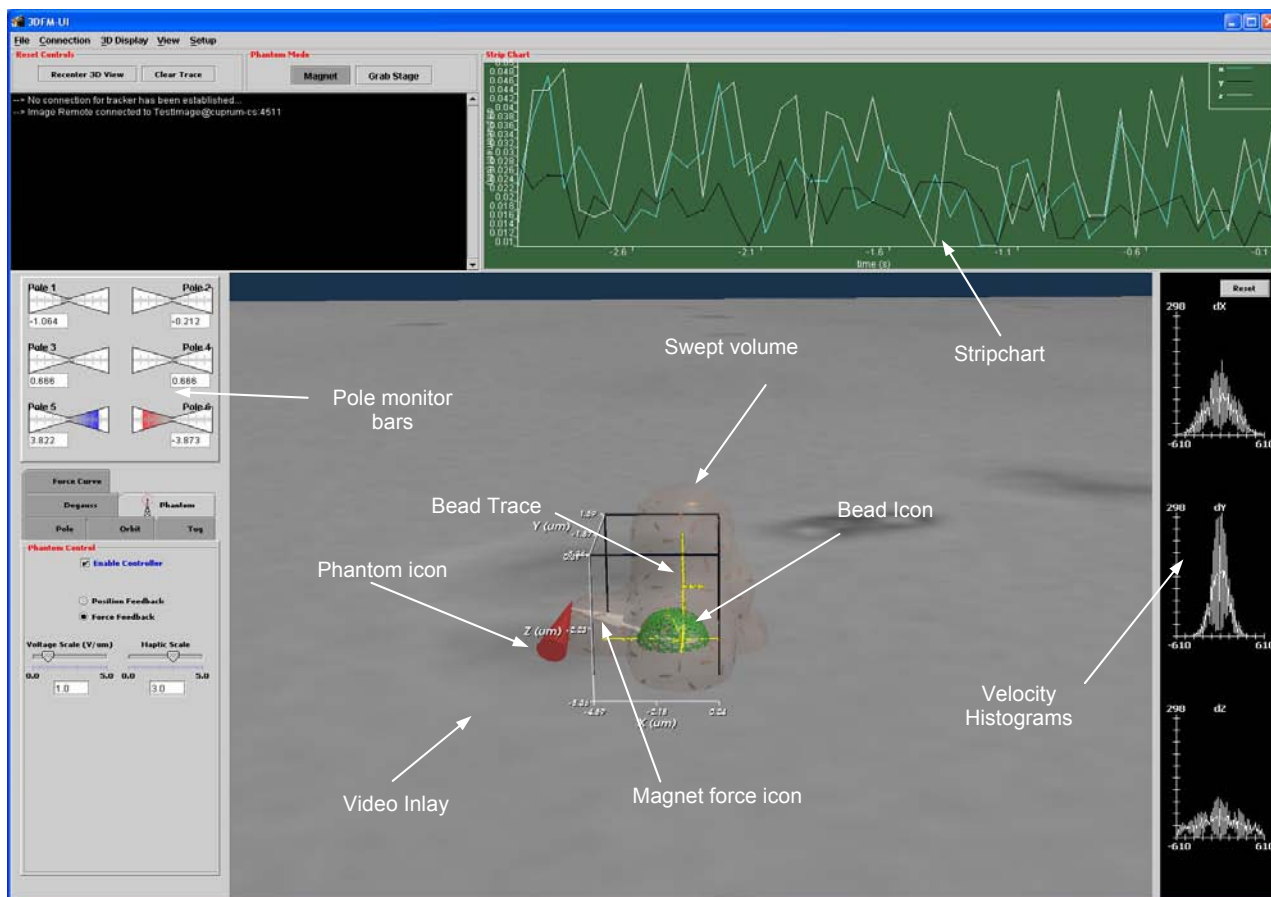


Figure 2: The 3DMF-UI (see Section 4) is annotated with labels describing control of visualization and microscope parameters and connections to the various other parts of the 3DFM software suite. The labels and arrows are overlaid in this figure.

widgets for graphing dynamic data vs. time in the stripchart fashion familiar to our scientist-users so created our own at our users' request. The stripchart graphs the x -, y - and z components of tracked bead motion vs. time and updates this graph dynamically as more tracking reports are received. It displays the most recent points, either a given number of points or for a given time.

4.3 The 3D Interface

Bead trace and bead icon: A yellow polyline traces the path of the bead over time, containing a segment for each position reported by the tracker server. A green wireframe sphere scaled to match the size of the tracked bead indicates the current bead location at the head of the trace and provides a sense of scale to the convoluted bends in the trace. Because the sphere is scaled proportionately to the physical bead size, there are important interactions with the video inlay (see below).

Visible axes, bounding box and physical-unit scales: Quantitative estimation is necessary in addition to qualitative display. Numbered scales are drawn on the bounding box around the bead trace and bead icon. The user-hand icon and the video inlay are excluded from this bounding box and scaled axes since they often extend well beyond the bead trace (which is what the scaled axes are intended to measure). The axes indicate microscope directions, providing orientation even in rotated views of the scene.

Magnet-force icon: This icon appears when a user uses the haptic device to specify magnetic forces (See section 4.4). A white arrow rooted at the current bead position displays the force being applied by the magnets. As the tracking program updates the bead position, the arrow is translated with new bead position; the direction does not change. As a user moves the input device, thereby changing the generated force vector, the direction of the arrow icon is updated. Originally, we displayed the forces returned by the magnet server. However, at that time, the magnet server used the multiplexing solution, and the arrow would jump around as the server time-multiplexed among magnetic poles. While the information we presented was true and correct in full detail, it was useless to our scientist-users, who were at a loss to interpret the rapidly jumping arrow. This icon now shows the force requested from the server while the pole monitor bars show jumping voltages.

User's hand icon: A cone icon appears when the haptic device is used to specify forces. The cone tip follows the Phantom's pen position and orientation. In force-drive mode, this serves only to verify visually the user's sense of hand motion. In position-drive mode (described in section 4.4), it indicates the position in the sample toward which the bead will be pulled.

Video inlay: We display an optical microscopy image, received from the video server, inlayed in the 3D display. The video plane is registered with the bead trace and the bead icon; it effectively follows the bead around. This is a correct behavior because the 3D motion stage moves to keep the bead cen-

tered in the laser beam and, hence, in the same spot in the camera's image.

Shell of swept volume: An estimated volume of bead exploration can be displayed. This accessibility surface represents the boundary of a volume, the interior of which has been explored by the tracked bead. For each position recorded along the bead trace, a 3D-Gaussian splat proportional to bead size is accumulated into a volume using the `vtkGaussianSplat` operation. An isosurface is then extracted from the volume.

An opaque surface would occlude the bead trace and other parts of the 3D interface. Rendering the accessibility surface with uniform translucency is not desirable because the shape of translucent objects is difficult to interpret, especially away from object silhouettes. Wireframe rendering is also a poor choice. In both cases, there are insufficient shape cues from illumination to perceive the surface shape well.

We implemented Interrante's method of curvature-directed strokes [13] to display this shell of swept volume. Curvature-directed strokes place textured strokes on an exterior translucent surface to enable better shape perception of the exterior surface without completely occluding the interior geometry. The strokes follow first principal-curvature direction, which is the direction of largest-magnitude local surface curvature. There are numerous methods for computing such textures. (Interrante [12], Interrante [13], Gorla [10], Turk [22])

Stereoscopic display: We use VTK's built-in ability to drive StereoGraphics CrystalEyes LCD shutter glasses and Stereo3D stereo emitters connected to an Nvidia Quadro-class graphics cards. Stereo turns out to be especially important in this application, where the lack of parallel structures and known scale indicators makes shape determination difficult.

4.4 The Haptic Interface

We use a SenSable Technologies Phantom Desktop device for six degree-of-freedom input and three degree-of-freedom haptic feedback when controlling the magnet subsystem. We provide two forms of magnetic control with the Phantom. The first specifies a force to apply to beads in a specimen; the second attempts to apply forces such that a tracked bead is directed toward a specified position.

In the force-specification mode, one end of a virtual spring remains clamped at the location where a user pressed the Phantom button; the other end of the virtual spring tracks the user's hand position as reported by the Phantom. Force on the bead is a scaled version of the vector from the "start" position to the current hand position. The negative of this force is presented to the user via the Phantom.

In position-specification mode, the force to be applied to the bead is proportional to the difference between the bead location and the present probe location, scaled by a linear factor. The force fed back to a user is the opposite of that presented to the bead. Note that the force presented to a user changes with a change of either the user's hand or the bead position. Feeling this in 3D results in a haptic display that matches the case where the bead is being pulled around by a spring. As a result, users can feel the motion of the bead. The bead-position update rate is well below the fusion frequency of the human haptic sense, so the results feel like small pops at up to 60 Hz, rather than smooth bead motion; however, big jumps, which often indicate significant events in experiments, are easily felt as sharp changes in force.

4.5 Successes and Failures

4.5.1 Veracity of data representation

Due to the evolution of the magnet server (see section 3.2), we ended up with several methods to display the force on a tracked bead when the magnet server is operated in force-specification mode. The first method corresponded to the force-projection solution, wherein we project the requested force to the nearest pole and apply a constant force in that direction. We haptically displayed this projected force to users so that they were aware of the direction in which they were really pulling. Sharp discontinuities in force were a side effect of this when the requested force crossed the plane between any two poles while applying high forces. This resulted in poor control, instability and confusion on the part of our users, even though it faithfully represented the force on the bead.

The second method corresponds to the time-multiplexing solution. It used a time-multiplexing method to present the most faithful force representation to a user by driving the haptic display with the same alternating force sequence used to drive the magnet cores. This resulted in force display that, both visually and haptically, was uninformative and difficult to control.

The third method was used later for the time-multiplexing solution and also for the force-solver solution. This method displays the requested force and was most satisfactory. The actual sequence of forces is recorded to the experiment log so that analysis can be done using detailed force information. For the time-multiplexing solution, a user can see from the pole-display bars that the pole excitations oscillate, whereas the pole excitations are static for the force-solver solution. The force arrow in the 3D display and the force presented in the haptic display are stable but only show the intended force, not the result.

4.5.2 The progress and quality of an experiment

Providing feedback to users while an experiment is running greatly speeds the scientific process. Instead of realizing, for instance, that a particular specimen was not viable during off-line analysis perhaps three days after an experiment, users can often find out a few minutes into an experiment and take appropriate action then.

Bead trace: Our users report that the form of the bead trace informs them of their progress on an experiment. Bead drift due to fluid flow is easily seen in the 3D display; likewise, lack of bead motion under the application of forces is easily detected. Depending on the specifics of the experiment, this may indicate a bead likely to be of interest or may indicate problems with specimen preparation that render the currently tracked bead uninteresting or the entire specimen invalid.

Histograms and the shell of swept volume: Given well-known velocity distributions for freely diffusing beads, it is easy to see from the velocity histograms if a particular tracked bead is stuck on some surface; likewise, for tethered-bead experiments, it is easy to see if the bead is freely diffusing. Our users also report that the shell of swept volume can be interpreted as a 3D histogram, in that the relative spread of the blob informs them about the diffusion of the bead.

Video inlay: Our users report that the video inlay provides context. In particular, it enables users to see the reaction of beads other than the one currently tracked, showing the relative motion among many beads and informing the scientists if some observed phenomenon is unique to the tracked bead or is a common among many beads. In one experiment with DNA

beads tethered to a surface, users were able to see in the video inlay that some beads behaved differently in the presence of applied magnetic forces. The different behavior confirmed suspicions that some beads were tethered by multiple DNA strands while some beads were singly tethered.

The wide view given by the video inlay is also useful for finding a next bead to be tracked in a series of force experiments in some specimen.

4.5.3 Combining data streams

Several data stream combinations have been particularly useful.

Magnets plus bead trace: This shows the correspondence (or lack of correspondence) between the force applied and the system response. It is validating and satisfying to see a bead move in the direction pointed by the magnetic force icon. Likewise, in some experiments, users can see what voltage must be applied to induce bead motion.

Video inlay plus bead trace: Several users report that having a live video plane in the 3D display area provides a relative scale to the motion of a tracked bead as well as a sense of proportion for the experimental specimen.

Additionally, users have reported that they are able to re-initiate tracking using only the video inlay. Tracking is “lost” when the bead, for any reason, moves outside the laser beam far enough that the QPD cannot detect and correct for the bead motion by moving the stage (and bead) back into the laser’s path. Because the bead in the image and the bead icon/trace head are necessarily aligned by virtue of the fact that the video camera and the laser share an optical path through the sample, users can adjust the stage position to re-align these after tracking is lost and then re-initiate tracking. Without the video inlay, users have to leave the 3DFM-UI console to re-initiate tracking at the computer running the tracking server, where video images of the QPD and the microscope oculars are visible.

4.5.4 Multiple views of the same data

The 3DFM-UI provides two displays of the same data in several cases; this has been done at the request of our users. The stripchart and the bead trace both show the same numbers. The bead trace provides the primary view of bead-motion data, while the stripchart makes visible significant short-time motion (“pops”). The stripchart also provides recent history when a long, convoluted bead trace has accumulated in the 3D display and the users don’t want to clear the trace and lose overall context.

The magnet-pole-indicator bars and the 3D force icon also display essentially the same information, although additional information is required to calculate the force from the coil excitations shown by the pole-indicator bars. Our users requested that we show the individual coil excitations in addition to the derived quantity of force-on-bead. Given that we eventually showed the requested force instead of the actual force in the 3D display, it was critical that we have this second means of showing magnet excitation. They provide two viewpoints of the same data, but in different spaces.

4.6 Additional Lessons Learned

Veracity vs. Accessibility: While it is useful to show aggregate high-level information about commands sent to the microscope, we found that our scientist-users also want to see all commands and details. In particular, the 3D display of the force arrow was not sufficient, even though it provides what

most would consider a more intuitive visualization of the action of the the magnet system on a bead; our users also wanted to see the energization of individual poles.

On other occasions, the “show everything” axiom was problematic. When the magnet server operated in time-multiplexing mode, the force arrow and the haptic display would cycle rapidly as the magnet server iterated over poles (see section 3.2). No user felt able to interpret the results displayed this way. Eventually, we haptically presented the force requested from the magnet server, rather than the force the magnet server applied. In a similar vein, showing all the video data when collected at a high frame rate is not possible, and we can only show some subset in time or space or both.

Eyes-free control: Eyes-free control was important to our scientist-users. The foot pedal and dial box combination are an example of eyes-free control for tracking initiation. The Phantom is an example of eyes-free control for force specification. These provide alternatives to aiming at mouse-based user-interface elements, which requires that the eyes move from the experiment to the controls.

Java: We were initially concerned about using Java for real-time control of our instrumentation. The fact that VTK wraps native calls for large rendering operations ameliorated this concern. Explicitly invoking the garbage collector avoids most interruptions for memory management. We have found no performance problems in the 3DFM-UI related to the interpreted or just-in-time-compiled nature of Java. Additionally, the extensive Java class library, especially the Swing user-interface library, was a great advantage and joy while developing the 3DFM-UI.

VTK: Although VTK is well-designed for rendering large data sets, it was not designed to be coupled to continuous incoming and accumulating data streams. To avoid VTK slowing down as it rebuilds the line trace each frame for an ever-growing bead path, we group the path into segments of fixed length and then render the list of segments. With this addition, the toolkit has served our needs well.

VRPN: Having a distributed-device package at hand greatly eased our development process and made it simple to distribute our considerable but loosely-coupled compute load over several computers. The logging features of VRPN let us easily store microscope data.

The high-level device abstractions provided by VRPN (Tracker, Button, ForceDevice, Analog, AnalogOutput, Imager, Poser) to which we committed for communication between programs allow us flexibility when developing new versions of servers and the 3DFM-UI or when changing specific algorithms internal to any of these programs. Two particular cases bear mentioning. First, the magnet server (see Section 3.2) exposes its two command modes, coil-voltage specification and force specification, as an AnalogOutput and a ForceDevice, respectively. Whichever interface has received a command most recently controls the magnet electronics, allowing us easily to switch command modes.

The second case is our on-going conversion of the 3DFM-UI and tracking server for use with a laser-trapping microscope. This microscope estimates forces based on the motion of a bead in a laser beam while a second laser beam provides restoring forces constantly over time. In this case, the tracking server also becomes the force server, a generalization of the magnet server, using the ForceDevice interface (but not the Analog/AnalogOutput interfaces). The difference in server is transparent to the 3DFM-UI, which continues to send and receive force commands and responses. Though this lesson

may have been often stated before, good software engineering pays off in spades and is well worth the time invested, especially when research systems become real applications with real users.

5 ACKNOWLEDGEMENTS

This work was supported by NIH P41 EB002025-21 RS/RMT as well as an NIH instrumentation development grant.

This work was possible only in the context of a number of students and collaborators within the multidisciplinary UNC Nanoscale Science Research Group. Especially important were the contributions of Leandra Vicci to design and control of the magnet system and the contributions of Kurtis Keller to the system mechanical design and fabrication.

Jeremy Cummings designed, built, and tested the first prototype 3DFM system and road-tested our early attempts at user interfaces. Tatsuhiro Segi and Steve Piantadosi contributed significantly to the design and implementation of the 2D user interface. Chris Dwyer constructed the sea-of-screens interface to the five computers that run the instrument that enabled us to get through the early months before the interfaces were consolidated.

David Hill designed and carried out the bead rheology experiments and formulated the analysis of Brownian viscosity studies. Gary Bishop and Greg Welch advised Kalpit Desai as he designed and implemented the “agnostic” tracking algorithm that is critical to the successful operation of the instrument.

Sreeja Asokan, Jing Hao, and other students used the 3DFM for a variety of materials-science and biology experiments, providing “road tests” of the instrumentation.

We would also like to thank the anonymous reviewers of this paper for their many detailed, insightful and helpful comments.

REFERENCES

- [1] M.W. Allersma, F. Gittes, M.J. deCastro, R.J. Stewart, and C.F. Schmidt, Two-dimensional particle tracking of ncd motility by back focal plane interferometry. *Biophys. J.* 74: pp. 1074-1085. 1998.
- [2] A. Bausch, F. Ziemann, A.A. Boulbitch, K. Jacobson, and E. Sackmann, Local measurements of viscoelastic parameters of adherent cell surfaces by magnetic bead microrheometry. *Biophys. J.* 75: pp. 2038-2049. 1998.
- [3] A. Bausch, W. Möller, and E. Sackmann, Measurement of local viscosity and forces in living cells by magnetic tweezers. *Biophys. J.* 76: pp. 573-579. 1999.
- [4] K.V. Desai. Agnostic tracking. In preparation.
- [5] M. Falvo, M. Finch, et al. The Nanomanipulator: A Teleoperator for Manipulating Materials at the Nanometer Scale. Proceedings of the 5th International Symposium on the Science and Engineering of Atomically Engineered Materials, Richmond, VA, World Scientific Publishing. pp. 579-586. 1995.
- [6] J.K. Fisher, L. Vicci, J. Cummings, K. Keller, B. Wilde, E. Timothy O'Brien, K.V. Desai, C. Weigle, G. Bishop, C. Davis, R. Boucher, R.M. Taylor, II, R. Superfine. 3D Force Microscope: a nanometric optical tracking and magnetic manipulation system for the biomedical sciences. To appear in *Review of Scientific Instruments* (accepted 3/25/05). 2005.
- [7] J.K. Fisher, L. Vicci, K.V. Desai, J. Cribb, B. Wilde, K. Keller, E. Timothy O'Brien, K. Bloom, R.M. Taylor, II, R. Superfine. A Three-Dimensional Force Microscope for the Biomedical Sciences. In preparation.
- [8] F. Gittes and C.F. Schmidt, Thermal noise limitations on micromechanical experiments. *Eur. Biophys. J.* 27: pp. 75-81. 1998.
- [9] F. Gittes and C.F. Schmidt, Interference model for back focal plane displacement detection in optical tweezers. *Optics Lett.*, 1998.
- [10] G. Gorla, V. Interrante and G. Sapiro. Texture synthesis for 3D shape representation. *IEEE Trans. on Visualization and Computer Graphics.* Vol. 9, no. 4. pp. 512-524. 2003.
- [11] D.B. Hill, Hiroshi Matsui, Jeremy Cribb, J. Sheehan; A. Yancey, S.I. Massenburg, R. Boucher, and R. Superfine, “Microbead Rheology of Human Mucus: Effect of percent solids on Viscoelasticity and Heterogeneity,” Submitted to *Biophysical Journal*, 2005.
- [12] V. Interrante, H. Fuchs and S. Pizer. Illustrating transparent surfaces with curvature-directed strokes. *Visualization '96.* IEEE Computer Society. pp. 211-218. 1996.
- [13] V. Interrante, H. Fuchs and S. Pizer. Conveying the 3D shape of smoothly curving transparent surfaces via texture. *IEEE Trans. on Visualization and Computer Graphics,* IEEE Computer Society. Vol. 3, no. 2, pp 98-117. 1997.
- [14] H. Matsui, V.E. Wagner, D.B. Hill, U. Schwab, T.D. Rogers, B. Button, R.M. Taylor II, R. Superfine, M. Rubinstein, B.H. Iglewski, R.C. Boucher, “A physical linkage between CF airway surface dehydration and P. aeruginosa biofilms,” Submitted to *Developmental Cell.* 2005.
- [15] A. Mehta, J.T. Finer, and J.A. Spudich, Reflections of a lucid dreamer: optical trap design considerations, in *Methods in Cell Biology.* Academic Press: New York. p. 47-69. 1998.
- [16] G. Sakas, M. G. Vicker, et al. Case Study: Visualization of Laser Confocal Microscopy Datasets. *Visualization '96,* San Francisco, IEEE Computer Society Press. 375-379. 1996.
- [17] R. Superfine, G. Bishop, J. Cummings, J. Fisher, K. Keller, G. Matthews, D. Sill, R. M. Taylor II, L. Vicci, C. Weigle, G. Welch and B. Wilde. "Touching In Biological Systems: A 3D Force Microscope." Proceedings of *Microscopy and Microanalysis 2002,* Quebec City, Canada. 2002.
- [18] R.M. Taylor II, W. Robinett, V. L. Chi, F.P. Brooks, W.V. Wright, R.S. Williams and E... Snyder. The Nanomanipulator: A Virtual-Reality Interface for a Scanning Tunneling Microscope. SIGGRAPH 93, Anaheim, California, ACM SIGGRAPH. 127-134. 1993.
- [19] R.M. Taylor II, J. Chen, S. Okimoto, N. Llopis-Artime, V.L. Chi, F.P. Brooks, M. Falvo, S. Paulson, P. Thiansathaporn, D. Glick, S. Washburn and R. Superfine. Pearls Found on the way to the Ideal Interface for Scanned-probe Microscopes. *Visualization '97,* Phoenix, AZ, IEEE Computer Society Press. 467-470. 1997.
- [20] R.M. Taylor II, T.C. Hudson, A. Seeger, H. Weber, J. Juliano and A.T. Helser. "VRPN: A Device-Independent, Network-Transparent VR Peripheral System." Proceedings of ACM Symposium on Virtual Reality Software & Technology 2001, Banff Centre, Canada. 2001.
- [21] R.M. Taylor II, D. Borland, F. P. Brooks Jr., M. Falvo, M. Guthold, T. Hudson, K. Jeffay, G. Jones, D. Marshburn, S. J. Papadakis, L.-C. Qin, A. Seeger, F. D. Smith, D. H. Sonnenwald, R. Superfine, S. Washburn, C. Weigle, M. C. Whitton, P. Williams, L. Vicci and W. Robinett. Visualization and Natural Control Systems for Microscopy. In: *Visualization Handbook.* C. J. a. C. Hansen, Harcourt Academic Press: 875-900. 2004.
- [22] G. Turk. Texture synthesis on surfaces. *Siggraph '01.* 2001. pp. 347-354. 2001.
- [23] L. Vicci. Analytical bead force model for the 3DFM. UNC Technical Report TR03-029. 2003.