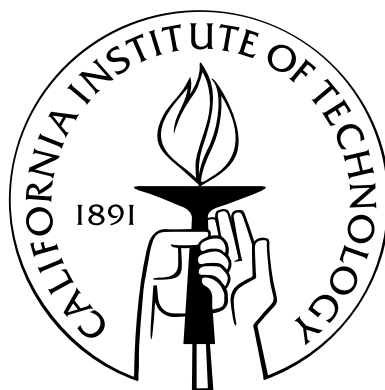


Apertureless Near-Field Optical Microscopy for Fluorescence Imaging

Thesis by
Guillaume Lessard

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy



California Institute of Technology
Pasadena, California

2003
(Defended 21 May 2003)

Acknowledgements

This document is the culmination of six years of work. This small portion of it is my opportunity to thank the friends and colleagues who have helped me through these years. It is a formidable challenge of conciseness.

First, I must thank my advisor Prof. Stephen R. Quake for entrusting me with this project. It has allowed me to bring together many of my seemingly disparate interests into a single piece of work, with a level of integration I would not previously have expected. The idea seemed simple enough at the time...

I would also like to thank Prof. William B. Bridges, who brought me to Caltech. We did not get the chance to collaborate in the lab, but I have always appreciated our conversations.

Thanks also to Prof. Demetri Psaltis, Prof. Axel Scherer, Dr. C. Patrick Collier and Dr. Marc Bockrath, my thesis committee members, for taking time to review my work.

In the lab, I have had the chance to interact with and benefit from the help of many great minds. Early on, fellow lab-mates Hou-Pu Chou and Charles Spence helped me many times to find and implement the right solutions. Hou-Pu, Marc Unger, Emil Kartalov, David Barsic and Marc Adams have all helped me with chemistry and fabrication issues that have been invaluable in making this project work. I have shared many interesting, sometimes memorable discussions with several members of the Quake group on topics scientific and unscientific. I thank Anne, Jim, Charles, Ido, Todd, Carl, Marc, Markus and Shuli for those moments. I must also express my thanks to Connie Rodriguez, our group secretary, who has always been helpful, since my very first day on the Caltech campus.

Even closer than the rest of the Quake group, I have teamed up with several people during these years. I co-built our original AFM setup with Francisco Guzman, then shared many moments of wonder and confusion with Pierre Barritault as we troubleshot our way through the first signs of success of the project. Jason T. Yang partook of the happy and momentous times when we were finally able to say that the Thing *worked*.

I have shared these past two years in the lab with two excellent and imaginative scientists and friends: Larry Wade and Jordan Gerton. There isn't much to say here except that it's been great. Thanks for showing up!

The work inside the lab wouldn't happen without the support of friends on the outside.

Daniel Zimmerman, one of my earliest acquaintances on campus, and Gustavo Joseph are two of my closest friends. I am thankful to Dan for subjecting himself to proofreading my thesis; it is much more readable thanks to his efforts.

Prashant Purohit and Patrick Hung are also prized friends, always ready to contribute time, summarize the situation and generally chip in. Thanks for the 88th key, Patrick!

Many have let themselves be dragged to jazz concerts at Catalina's, the Jazz Bakery and beyond; David Barsic and Ben Miller have been some of my favorite victims on that front, and the music was that much better.

Vilia and Jonas Zmuidzinas have been good and generous friends. They, along with their daughters Lina and Regina, have welcomed me in their home uncountably many times and made me feel part of their family.

Another friend from my earliest days at Caltech deserves mention. Father Brian Muzas, with his spectacular career change from rocket scientist to priest, has inspired me with his faith and courage.

Finally, I would like to thank my parents and my brother Pascal. They set high standards, they supported me, I could not have hoped for more.

Abstract

This thesis describes the development of a new type of optical microscope, an apertureless near-field microscope for fluorescence imaging (FANSOM). We have demonstrated that it is possible to use the probe of an atomic force microscope (AFM) to create a detectable modulation of the fluorescence of a nanometer-scale fluorescent object, and that it is possible to use that modulation to produce images by raster-scanning. Our results show that such a microscope is capable of a lateral resolution better than 20 nm.

The processes that cause this modulation act for very small probe-sample separations, smaller than 20 nm. We have demonstrated FANSOM contrast using two completely different effects, one being a fluorescence inhibition, the other being a fluorescence enhancement generated by electric field enhancement. We have obtained contrast ratios of 1.90:1 using the fluorescence inhibition effect, and 5:1 using the fluorescence enhancement effect.

Fluorescence can be inhibited by the proximity of a metallic probe. We have mapped the vertical profile of this phenomenon, and have shown that its shape is dependent upon the method of illumination. With evanescent illumination, the closest range of the interaction can exhibit some fluorescence enhancement that partially cancels the fluorescence inhibition effect.

When a metallic or dielectric probe approaches a sample illuminated by an evanescent illumination field polarized parallel to the probe (i.e., vertically with respect to the surface), field enhancement occurs. A local fluorescence enhancement is detected as a result of the field enhancement. The range of this effect is less than 10 nm, located mostly under the AFM probe; the optical images acquired using this contrast mechanism exhibit a lateral resolution equal to or higher than the topographic resolution measured by AFM.

The design of the data acquisition system allows us to obtain precise mappings of signal intensity to probe-sample separation, more precise than have previously been achieved in an apertureless near-field microscopy system. It has allowed us to obtain repeatable approach curves that differ clearly from one another when the type of illumination or probe is changed.

Contents

Acknowledgements	iii
Abstract	v
Table of Contents	vi
List of Figures	x
List of Tables	xii
1 Introduction	1
1.1 Near-field Optics	1
1.2 Apertureless Near-field Optical Microscopy	2
1.2.1 The Game of the Name	3
1.2.2 Early Developments	3
1.3 Fluorescence Microscopy	4
1.4 Fluorescence Apertureless Near-field Microscopy	5
1.5 FANSOM Basics	6
1.6 Conceptual Design of the FANSOM	7
1.7 Outline	8
2 The Early FANSOM	9
2.1 Experimental Apparatus	9
2.1.1 Home-built Atomic Force Microscope	9
2.1.1.1 Mechanical System	9
2.1.1.2 Probe Holder	10
2.1.1.3 AFM Optical System	11
2.1.1.4 Feedback Electronics	12
2.1.2 Inverted Optical Microscope	13
2.1.3 Data Acquisition	16

2.1.3.1	AFM Driver and Topographical Data Acquisition	16
2.1.3.2	FANSOM Data Acquisition	17
2.1.3.3	Data Handling and Visualization	20
2.1.4	Alignment of the FANSOM	20
2.2	Experimental Results	20
2.2.1	Early Results	21
2.2.2	FANSOM with Gated Photon Counting	24
2.3	Limitations	27
2.3.1	Atomic Force Microscope	27
2.3.2	Alignment and Drift	28
2.3.3	Limited Phase Sensitivity	28
2.4	Conclusion	28
3	Second-Generation FANSOM	29
3.1	Atomic Force Microscope	29
3.1.1	Issues with the DI System	30
3.1.2	Vibration Isolation	31
3.1.3	Suppression of Amplified Spontaneous Emission	31
3.2	Inverted Optical Microscope	32
3.2.1	Vibration Isolation	32
3.2.2	Illumination and Beam Steering	33
3.2.2.1	Beam Shaping	34
3.2.3	Optical Detection System	36
3.3	Data Acquisition	37
3.3.1	Nature of the Data	37
3.3.2	Synchronization	37
3.3.3	Interpreting the Data	38
4	Mathematical Modeling and Data Processing	40
4.1	A Model for the FANSOM Data	40
4.2	Time-Binned Data Acquisition Model	42
4.2.1	Modeling FANSOM Data with the Poisson Distribution	42
4.2.2	Signal-to-Noise Ratio in Images	44
4.2.2.1	Typical Conditions of Illumination	45
4.2.2.2	Signal-to-Noise Ratio for a Single Object	46
4.2.2.3	Signal-to-Noise Ratio for Multiple Objects	47
4.2.3	Photon Budget and Signal-to-Noise	47

4.2.4	Limitations of the Opposed Bins Method	50
4.3	Fourier Analysis	50
4.4	Correlation Techniques	52
4.4.1	Implementation of Correlation Analysis	53
4.5	Probe/Sample Approach Curves	54
4.5.1	Computation of Approach Curves	54
5	Experimental Results	56
5.1	Sample Illumination	56
5.1.1	Beads under Illumination	56
5.1.1.1	Direct Illumination	56
5.1.1.2	Annular Total-Internal-Reflection Illumination	56
5.1.1.3	Crescent Illumination (Directional-TIR)	58
5.1.2	AFM Probes: Imaged Backscatter	59
5.1.3	Fluorescence Pattern and AFM Probes	59
5.2	Fields, Probes and Fluorescence	59
5.2.1	Fluorescence Inhibition	61
5.2.2	Field Enhancement	66
5.3	FANSOM Images	69
6	Conclusion	76
6.1	Summary	76
6.2	Outlook and Future Directions	76
A	AFM Probes	78
B	Sample Preparation Methods	81
B.1	Glass Cleaning Methods	81
B.1.1	The RCA Cleaning Procedure	82
B.1.2	HCl/NaOH Cleaning Procedure	83
B.2	Fluorescent Bead Sample Preparation	83
C	Fabrication of Aluminum Masks	85
C.1	Mask Design	85
C.2	Wafer Preparation	86
C.3	Lithographic Transfer	86
C.4	Aluminum Etching	86
C.5	Mask Removal	87

D	Data Formats and Data Analysis	88
D.1	Data File Formats	88
D.1.1	AFM and Optical Images	88
D.1.2	Timestamp Data Files	92
D.2	FANSOM Data Analysis Scripts	94
D.2.1	Photon Phase Offset Calculation	94
D.2.2	Image Generation Functions	96
D.2.3	Approach Curve Generation Functions	100
E	Data Acquisition Code	103
E.1	AFM and Optical Data Acquisition	103
E.1.1	Optical Scans	112
E.2	FANSOM Data Acquisition	119
	Bibliography	126

Figures

1.1	Illustration of probe-induced modulation	6
1.2	Schematic of the FANSOM apparatus	7
2.1	Picture of the homemade AFM and FANSOM system	10
2.2	Picture and diagram of the homemade AFM-probe holder	11
2.3	Transmission spectrum of the laser diode clean-up filter	12
2.4	Diagram of the correction circuit for the AFM feedback system	14
2.5	Diagram of the inverted optical microscope	15
2.6	Transmission spectrum of the detector bandpass filter	16
2.7	Diagram of the demodulation circuit for analog FANSOM	18
2.8	Images of a fluorescent 100 nm bead taken with the analog FANSOM	22
2.9	Cross section of an analog FANSOM image and far-field image	23
2.10	Decay of the modulation in the fluorescence spectrum vs. time	24
2.11	Images of a fluorescent 60 nm bead taken with the digital FANSOM	25
2.12	Normalized fluorescence as a function of probe-sample distance	26
3.1	Picture of the second-generation FANSOM apparatus	30
3.2	Transmission spectrum of the dimension head's laser clean-up filter	32
3.3	Picture of the optical microscope setup in the second-generation FANSOM	33
3.4	Diagram of the inverted optical microscope for the second-generation FANSOM	34
3.5	Full beam illumination at the sample plane	35
3.6	Polarization behaviour of a totally internally reflected ray	36
3.7	Diagram of the clock synchronization circuit	38
4.1	Variation of SNR as q_0 and λ vary, for a single object	46
4.2	SNR curves for various numbers illuminated objects	48
4.3	Variation of the SNR as the number of illuminated objects grows	48
5.1	Image of a 20 nm fluorescent bead with direct illumination	57
5.2	Image of a 20 nm fluorescent bead illuminated by a TIR ring	57

5.3	Image of multiple 20 nm fluorescent beads illuminated by TIR	58
5.4	Image of backscattered photons from a PIT probe	60
5.5	Image of a 20 nm fluorescent bead illuminated by an annular TIR beam, in the presence of a platinum/iridium-coated AFM probe.	61
5.6	Phase offset histogram for a 20 nm bead modulated by a silicon probe	62
5.7	Effect of a PIT probe on a 20 nm bead excited by Gaussian illumination	63
5.8	Effect of a PIT probe on a 20 nm bead excited by annular TIR illumination	64
5.9	Effect of a PIT probe on a 20 nm bead excited by TIR illumination with polarization parallel to the sample plane	65
5.10	Effect of a silicon probe on a 20 nm bead excited by TIR illumination with a polarization perpendicular to the sample plane	67
5.11	Effect of another silicon probe on a 20 nm bead excited by TIR illumination with a polarization perpendicular to the sample plane	68
5.12	Effect of a PIT probe on a 20 nm bead excited by TIR illumination with polarization perpendicular to the sample plane	70
5.13	FANSOM image of a 20 nm bead with a PIT probe, in fluorescence inhibition mode . .	71
5.14	Cross sections of the images from Figure 5.13	72
5.15	FANSOM image of a 20 nm bead with a silicon probe, in fluorescence enhancement mode	73
5.16	Cross sections of the images from Figure 5.15	74
A.1	TEM images of metal-coated AFM probes at 260 000 \times magnification	79
C.1	A mask design for illumination beam shaping	85

Tables

4.1	Photon detection rates for various objects	45
A.1	Characteristics of silicon probes	78
B.1	Nominal characteristics of IDC latex beads	84

Chapter 1

Introduction

Ever since Leeuwenhoek used a single-lens microscope to make the first observations of bacteria and single cells in the late seventeenth century [1], the microscope has been among the most important tools for discovery in the biological sciences. As technology has progressed, allowing for better resolution, better fields of view and improved methods of contrast, microscopes have been applied to the observation of smaller and more complex samples, elucidating previously unknown mechanisms and structures. The optical microscope is often said to be the most successful scientific instrument in history.

During the twentieth century, optical microscopy went from having a monopolistic hold on small-scale imaging to being seemingly passé, its resolution lagging far behind that of a number of newer techniques, from transmission electron microscopy to scanning tunneling microscopy. The resolution of optical microscopy has stopped progressing, as modern fabrication techniques now routinely produce images at the ultimate resolution allowed by the diffraction limit.

Optical microscopy has not lost its popularity, since it has a number of advantages over the newer techniques. It is still desirable to use light as a contrast mechanism in a realm of very small scales, much smaller than diffraction allows: optical information carries precious chemical fingerprints, providing material information that is difficult to obtain otherwise.

Higher energy electromagnetic sources can be used to obtain high-resolution information with methods analogous to optical microscopy, but the energy of the photons rapidly becomes sufficient to destroy chemical bonds. The advantages of optical microscopy disappear, just as they were not retained by electron microscopies.

1.1 Near-field Optics

In the early 1980s, optical observations made a leap in resolution thanks to the new technique of near-field scanning optical microscopy (NSOM). The idea is to illuminate a sample with a localized light source whose dimensions are much smaller than the light's wavelength. In practice, this

has been done with an elongated, metal-coated optical fiber, at the end of which a minuscule, nanometer-scale aperture is fabricated. The aperture, used as a localized light source, is scanned in close proximity to the observed sample in order to obtain a high-resolution profile of the sample's interaction with light. Images are obtained by raster-scanning the probe over a sample.

The earliest discussion of a possible method for near-field optical microscopy appeared as early as 1928, when Synge suggested the illumination of samples through a minute hole at the apex of an otherwise opaquely coated cone [2]. The idea was seemingly ignored, probably for practical reasons, and similar ideas were published again in the 1950s [3, 4]. Following that, subwavelength resolution was demonstrated with the use of acoustic waves [5]. The earliest implementation of a scanning near-field microscope was demonstrated in 1972 by Ash and Nichols [6], when they produced line scans of metal gratings with a resolution of $1/60^{\text{th}}$ of a wavelength using 3 cm radiation.

The first optical near-field microscopes were developed in 1982 [7, 8], early in the flurry of development of scanning-probe technologies fueled by the resounding success of the scanning tunneling microscope. The instrument's probes were metal-coated optical fibers, as described above; Pohl and Denk [9] called the invention an "optical stethoscope" in analogy to a physician's stethoscope, which enables the detection of long-wavelength acoustic waves using a small aperture. To this day, most of the work conducted on near-field optics uses fiber-based probes similar to those developed by Denk et al.[9] and Lewis et al.[8, 10] for their early NSOM work.

The "aperture" probes of NSOM run into a physical limit for resolution, albeit one much higher than the diffraction limit. If the metal coating of the fiber were a perfect conductor, no field would penetrate the metal and aperture sizes would not have a lower bound; in practice, though, the metal coating is a relatively poor conductor at optical frequencies, and electromagnetic fields penetrate the metal. The effective aperture size is limited by this penetration, even for arbitrarily small physical aperture sizes. The best material available is aluminum; of all usable metals, it allows the least penetration of electromagnetic energy at optical frequencies. Using aluminum, effective aperture sizes have a lower bound of approximately 30 nm. It is noteworthy that many early NSOM papers have claimed resolutions better than 30 nm, a result that was later disproved and attributed by Hecht et al. [11] to a common imaging artifact that can obscure the genuine NSOM contrast.

While the 30 nm limit of NSOM represents a tenfold resolution improvement over the diffraction limit, it became a new limit to overcome in optical microscopy.

1.2 Apertureless Near-field Optical Microscopy

Another offspring of the scanning tunneling microscope, the atomic force microscope (AFM), was developed in the late 1980s. The AFM enables the formation of images by scanning the topography

of surfaces with a very gentle contact probe; the use of this type of probes was suggested for high-resolution optical observations by Wickramasinghe [12, 13]. In the proposed configuration, the subwavelength-sized AFM probe introduces a short-range perturbation to an optical field. The perturbation is detected, and its magnitude used as the image data. If the interaction range of the measured perturbation is short enough, Wickramasinghe suggested that this “apertureless” form of optical microscopy would likely surpass the resolution capability of aperture probes.

The key concept of apertureless NSOM is the periodic perturbation of the interaction between a sample and its illumination, using a sharp probe. The modulation so produced is detected in the far field and separated from a low-frequency signal using a synchronous detection scheme such as lock-in amplification.

1.2.1 The Game of the Name

The term *near-field optical microscopy* was chosen for the aperture-scanning family of instruments in reference to diffraction theory. In those instruments the high-resolution contrast is formed when objects obscure the illumination radiation in the near-field region beyond the aperture, before the radiation pattern stabilizes into a diffraction-limited pattern we normally detect and perceive. In light of this fact, can a new family of optical microscopes based on short range perturbations be called “near-field?”

The consensus among microscopists is “yes.” Paesler and Moyer [14] define near-field optics as *the branch of optics that considers configurations that depend on the passage of light to, from, through, or near an element with subwavelength features and the coupling of that light to a second element located a subwavelength distance from the first*. That definition has much merit, as it avoids the pitfall of using terminology that could be interpreted differently in the various subfields of electromagnetics, such as optics or antenna theory.

With this definition in mind, we can see the microscope proposed by Wickramasinghe [13] can legitimately be called an apertureless near-field microscope.

1.2.2 Early Developments

Various modulation methods have been proposed for apertureless NSOM. The original proposal by Wickramasinghe’s team [13] involves a dual modulation technique wherein the sample and the probe are dithered at different frequencies; the short range interaction is expected to manifest itself as an heterodyned signal of the two oscillation frequencies. The observation is done at the illumination wavelength.

Wickramasinghe’s team showed images that suggested the proposed microscope had the ability to resolve 5 nm features, but the contrast mechanism of their instrument was not well explained,

either in the initial [13] or the follow-up [15, 16] publications. The particular modulation technique used by Wickramasinghe has not been used by other groups in later publications.

Another apertureless technique, based on the periodic occlusion of an illuminating beam by a sharp metallic probe, was published by a group led by Boccara [17, 18, 19]. Like the Wickramasinghe technique, the Boccara technique did not have many followers. The published results showed some optical contrast, generally at a disappointing level of resolution. Furthermore, there was a lack of compelling evidence that the contrast was really optical, as the samples used usually had complete correlations between their topographical and optical images, with an incomplete justification for the optical contrast on the length scale shown.

A third technique was proposed by an Osaka University team led by Kawata [20, 21]. Their technique is often called *frustrated total-internal-reflection near-field microscopy*. It consists of using a tapping AFM probe to locally perturb the optical evanescent field at a glass-air interface above a prism.

The published results of the frustrated-TIR method showed some subwavelength contrast, but suggested a probe-sample interaction length of 200 nm vertically. Published frustrated-TIR results also used a side detection scheme with a relatively small collection efficiency. The collection efficiency of a side detection scheme would be a problem for fluorescence detection, while interaction distances of 200 nm would inevitably limit the attainable resolution.

1.3 Fluorescence Microscopy

Molecular fluorescence is a light-matter interaction by which a molecule reemits light after having absorbed a shorter wavelength photon. The molecule gets kicked into an excited state by the incident photon, then settles to its first singlet state (a metastable state) through vibrational relaxation. After a short pause of on the order of a nanosecond, the molecule returns to its ground state by one of several competing mechanisms. One of these mechanisms is the emission of a photon—fluorescence emission.

Fluorescence was first observed by a Spanish physician, Nicolas Monardes, in 1565 [22]. Based on his reports, others—including Newton—investigated the phenomenon, but it was not understood. Following important observations by Brewster and Herschel in the nineteenth century, Stokes demonstrated in 1852 that fluorescence was an emission of light following an absorption of light [23]. Stokes is also responsible for coining the term *fluorescence*. In the early twentieth century, the development of fluorescent staining led to the use of fluorescence to outline structures that would otherwise be difficult to see in biological samples. Fluorescence staining later evolved into number of of tagging techniques, e.g., protein tagging, where proteins can be used to make tiny fluorescent beacons track the inner workings of cells. Fluorescent tags are widely used by bi-

ologists; the procedures by which they are applied are well known. In fact, judging by the breadth (and thickness) of the catalogs of companies like Molecular Probes, fluorescent tagging techniques are a big industry.

Nowadays, the detection of a single fluorescent molecule has become a practical matter in a variety of environments [24, 25, 26]. This represents the highest possible sensitivity with respect to fluorescence; fluorescence detection techniques have reached the best possible “height” discrimination. In the case of lateral resolution, however, the ultimate has not yet been reached.

1.4 Fluorescence Apertureless Near-field Microscopy

The project chronicled in this thesis started with the stated goal of producing an instrument capable of the ultimate possible resolution and sensitivity in imaging fluorescent samples. The intention is to provide a sharper “eye” than ever before for the observation of biological molecules.

The FANSOM project aims to leverage existing knowledge in fluorescence microscopy, making AFM-like resolution available to biology research in a way that can be sensitive to the function of the observed samples.

Depending on the resolution attained, the FANSOM could find applications in DNA sequencing or proteomics. Other proposed applications include studies of the molecular interactions within the aqueous region of a cell membrane. These proposed applications have in common the exploitation of the FANSOM’s potential for very high resolution and its ability to exploit the chemical sensitivity of fluorescent tags.

Even with a limited improvement in resolution, the FANSOM could be a step forward for near-field biological imaging. Metal-coated fiber probes used as illumination sources have the problem of exhibiting strong local heating which can be detrimental to samples. The apertureless approach does not require as much illumination power, making local heating a minor issue.

Other groups have worked concurrently with this project towards fluorescence sensitivity in apertureless NSOM. The most notable success was obtained by the team of Sánchez, Novotny and Xie, with their development of an instrument demonstrating the enhancement of two-photon absorption in the near-field [27], resulting in fluorescence emission.

Other investigations of fluorescence in the proximity of probes have also been reported by Hayazawa, Inouye and Kawata [28], and the team of Gallagher and Nesbitt at the National Institute of Standards Technology (NIST) [29, 30]. The imaging method presented by Hamann et al. exhibits subwavelength contrast, but the subwavelength features are accompanied by large interference patterns on a scale comparable to the wavelength of the illumination light. These interference fringes would make imaging difficult in a practical application, since the superposition of fringes from multiple samples would obscure the structures of interest.

1.5 FANSOM Basics

The key concept of apertureless NSOM is the periodic perturbation of the interaction between a sample and its illumination, using a sharp probe. The modulation so produced is detected in the far field and separated from a potentially much larger low-frequency signal using a synchronous detection scheme such as lock-in amplification.

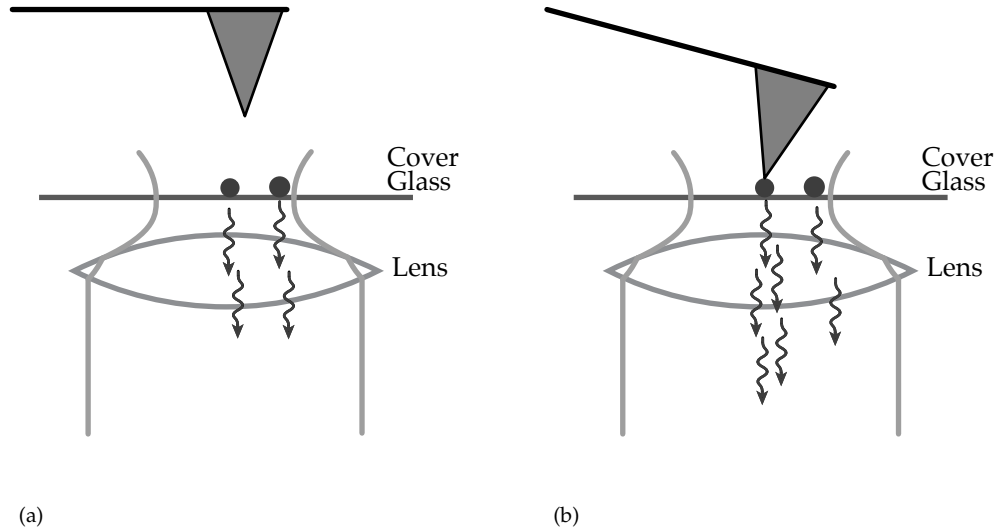


Figure 1.1: Illustration of probe-induced modulation. In (a), the probe has no effect on the fluorescence of the illuminated objects. In (b), the probe locally changes the fluorescence rate of one of the illuminated objects.

In the left half of the figure above, two identical objects are illuminated by a laser beam. In the far field, the two objects are not resolved. When a probe comes close to one of the objects (right half of the figure), it causes a local change in conditions, and the amount of light detected in the far field is changed. If the probe were stationary, it would not be possible to differentiate the two objects after the light has propagated to the far field.

If the probe were to oscillate rapidly between the two states shown at Figure 1.1, the optical properties of the object beneath the probe would be modulated, and it would become possible to quantify the local change of the object's optical properties that is due to the probe. The properties of other illuminated objects would not be modulated, allowing discrimination between objects. An atomic force microscope operating in the tapping mode is an excellent candidate to provide that modulation.

In the tapping mode of operation, also called intermittent-contact mode, the probe is made to oscillate vertically at its resonance frequency and brought close to a surface until it strikes the surface at the bottom of each of its oscillations [31, 32]. The oscillation amplitude is then kept constant

by using the height of the probe assembly as a feedback control parameter. The average probe-to-sample separation is therefore also kept constant, since it is directly related to the oscillation amplitude.

1.6 Conceptual Design of the FANSOM

In the FANSOM, the optical interaction of interest is the excitation–deexcitation cycle of fluorescent particles. We probe the space near a fluorescent sample with an AFM tip, and detect variations in the amount of fluorescence we detect. The conceptual construction of the FANSOM is shown in Figure 1.2.

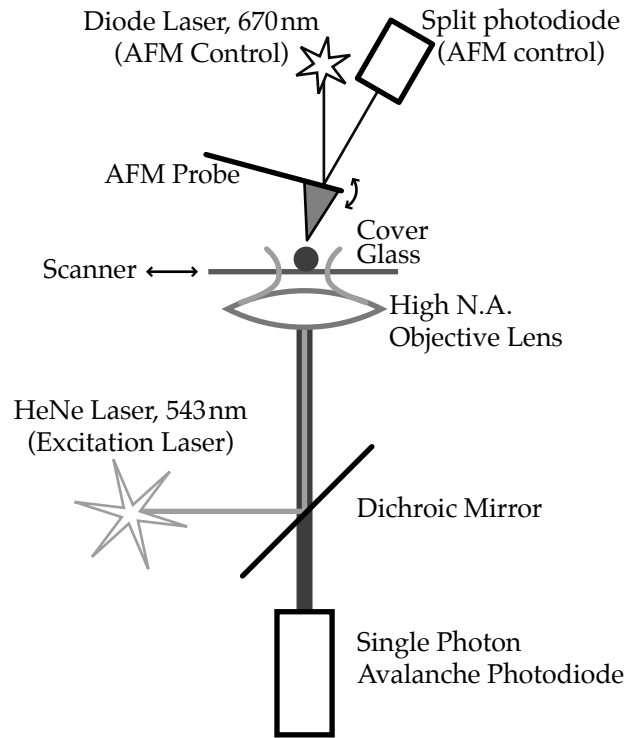


Figure 1.2: Schematic of the FANSOM apparatus.

A green laser (543 nm wavelength) illuminates a fluorescent sample through a high numerical aperture (N. A.) microscope objective; the fluorescence is collected by the same objective, and is detected by a single-photon detector.

The output from the single-photon detector contains the information we’re after—the modulated fluorescence. We wish to obtain the best possible measurement of this quantity, both from

the physical and the signal processing points of view—we want first to obtain a clean and relevant signal, and then extract as much information as possible from the signal.

1.7 Outline

In the next chapter, we describe the early versions of the FANSOM experimental apparatus along with the corresponding initial experimental results. Then, in Chapter 3, we describe a newer incarnation of the experimental apparatus, based on a commercial atomic force microscope—a major overhaul of the instrument. Then, Chapter 4 is devoted to data modeling and data processing, and results are presented in Chapter 5.

Chapter 2

The Early FANSOM

Efforts to realize a working apertureless NSOM began with a design centered around a home-built AFM. One of the main reasons to use a home-built AFM was cost, but another benefit of that approach was that it provided much-needed flexibility during the exploratory phases of the project. Although the performance level of a commercial instrument would have been superior, such an instrument would not have provided a comparable level of control of all the applicable operational parameters.

2.1 Experimental Apparatus

2.1.1 Home-built Atomic Force Microscope

Our home-built AFM was designed as a sample-scanning device operating in tapping mode [32]. Early attempts with contact mode AFM imaging showed that latex beads, the samples we intended to use, were dragged around by a contact probe. The original work in apertureless NSOM having used oscillating probes [13], it was logical to pursue the development of the tapping mode of operation for our AFM: it would likely solve two problems at once, by simultaneously preserving sample stability and potentially modulating the near-field FANSOM signal.

2.1.1.1 Mechanical System

An essential basis for an atomic force microscope is a stable and stiff construction, with good isolation from the acoustic noise generated in the environment. However, one of the requirements of our AFM was that it be positioned on top of an inverted microscope, which included a fairly large optical system. This required a compromise on the acoustic isolation, as a result of which the microscope was simply bolted on a floating optical table for easy coupling with an optical system. Such an optical table provides good low-frequency isolation, but poor isolation at frequencies over 100 Hz.

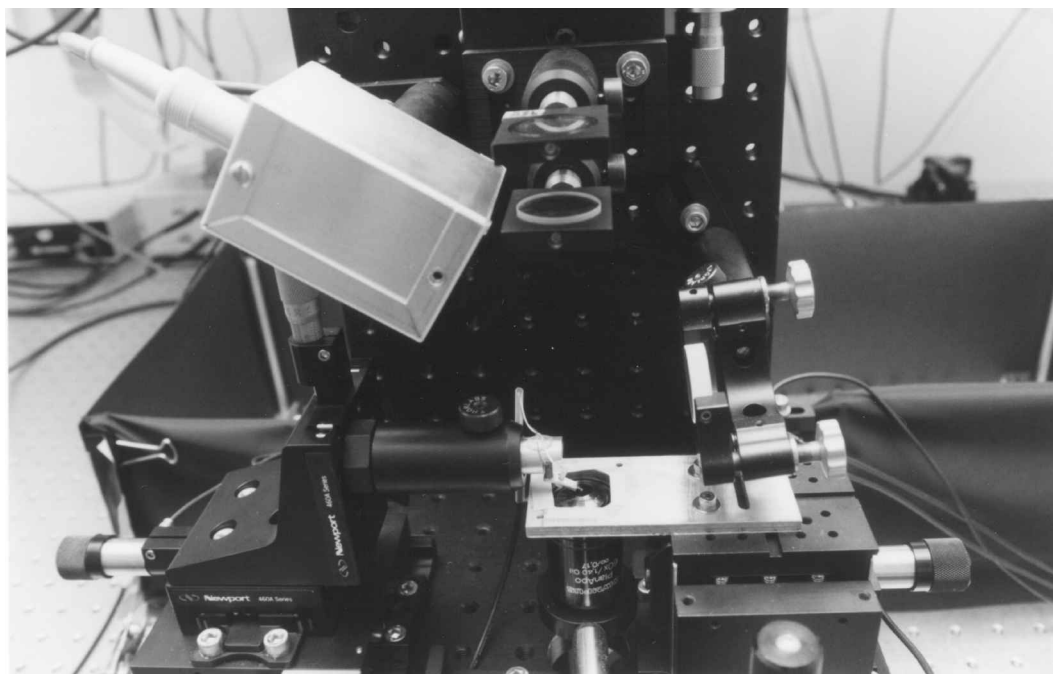


Figure 2.1: Picture of the homemade AFM and FANSOM system.

The instrument was built from a collection of opto-mechanical mounts from Thorlabs Inc. and Newport Corporation, as shown in the photograph of Figure 2.1. The sample scanner and the probe holder assembly were mounted face-to-face on an optical breadboard. The components necessary for the optical system to detect cantilever deflections were mounted on a second breadboard attached perpendicularly to the first one.

Two parts of the system needed to be actuated during normal operation: the AFM probe holder and alignment assembly (pictured on the left) needed to be adjusted during the various system alignment steps; the sample scanner (pictured on the right, below the mirror) scanned the sample through the detection region during the acquisition of images. The AFM alignment assembly was a combination of translation stages providing three-axis movement. The sample scanner was a three-axis flexure stage (Thorlabs MDT630), off which a sample holder was cantilevered in order to provide clearance for a microscope objective. The X and Y axes of the sample scanner and the three axes of the probe holder were movable with combination actuators incorporating piezoelectric stacks and adjustment screws.

2.1.1.2 Probe Holder

AFM probes are sold as an integrated assembly in which the nanometer-sized probe is attached to a micrometer-sized cantilever, in turn attached to a millimeter-sized substrate that must be securely mounted in the instrument.

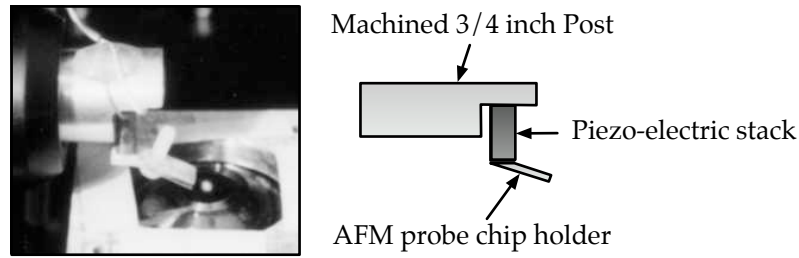


Figure 2.2: Picture and diagram of the homemade AFM-probe holder.

The probe holder was built from a machined 3/4 inch post to which a piezoelectric stack was attached using an epoxy. At the other end of the piezoelectric stack, a small assembly of aluminum and spring steel served as the attachment point for the probe assembly. The piezoelectric stack on the probe holder fulfilled three functions: fine adjustability for the sample approach phase, excitation of the cantilever for intermittent contact operation, and application of a variable offset allowing the probe to track topographic features.

The probe holder was made to be easily removable in order to facilitate the exchange of samples. It attached to the alignment assembly with a standard Thorlabs Inc. 3/4 inch post holder.

2.1.1.3 AFM Optical System

Tip-sample distance regulation in the AFM was done using the standard method of beam deflection, wherein a laser beam is reflected off the end of the AFM probe's cantilever and is detected far away from the probe by a split photodiode position detector.

The laser used to detect the cantilever deflection was a readily available 670 nm laser diode, focused onto the cantilever using a 100 mm lens. In addition to this lens, a narrow bandpass filter centered at 670 nm was positioned between the laser and the cantilever. The purpose of this filter was to block the amplified spontaneous emission (ASE), the photons emitted by the laser crystal but not amplified by the laser effect. A laser diode emits a relatively small but detectable amount of such photons at wavelengths in the neighborhood of the laser line. In the case of a typical 670 nm diode, ASE energy can be detected between 600 nm and 700 nm. If left unblocked, the number of ASE photons detected by our detector between the wavelengths of 600 nm and 620 nm would be of the order of 10^5 and 10^6 per second; considering that the signal we wished to observe with our system is on the order of 10^4 photons per second, it was clearly necessary to use a bandpass filter to "clean up" the laser beam.

The cleaned-up laser beam is focused onto the end of the cantilever, whose top surface is reflective (being the surface of a silicon crystal). The reflection is then detected at a distance using a split photodiode, taking advantage of a geometric amplification.

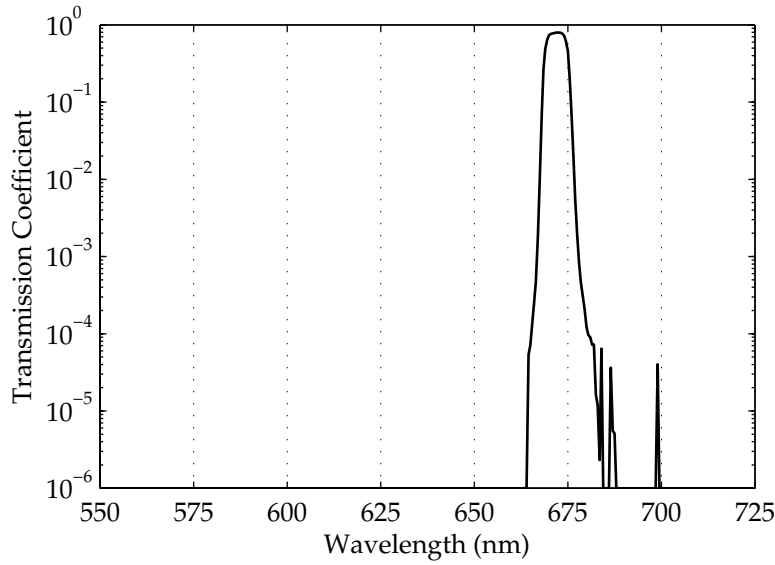


Figure 2.3: Transmission spectrum of the optical filter (Chroma HQ670/10, batch number 10762) used as a clean-up filter for the laser diode. The energy present outside of the laser line, if left unblocked, would be sufficient to drown the signals of interest in the optical end of the FANSOM.

Given D , the distance from the probe to the photodiode, and l , the length of the AFM cantilever, vertical displacements of the AFM probe are amplified by a factor of $2D/l$ at the split photodiode. The cantilever length l is equal to $100\text{ }\mu\text{m}$ for the most commonly used cantilevers, and in our case the distance D was approximately 150 mm , resulting in an amplification factor well over 10^3 . With a typical oscillation amplitude in the 50 to 100 nm range, the resulting oscillation of the reflected spot is over $100\text{ }\mu\text{m}$, an easily detectable displacement.

The photocurrents from each side of the photodiode were converted to voltages V_{high} and V_{low} using a simple current-to-voltage conversion circuit (a basic op-amp application). The voltages were then distributed to the feedback electronics, where the quantity $(V_{high} - V_{low})$ was interpreted as a reading of the AFM cantilever displacement¹.

2.1.1.4 Feedback Electronics

In the tapping mode of atomic force microscopy, the probe assembly is driven to oscillate at the resonance frequency of its cantilever. As the probe approaches and touches the sample, the amplitude and phase of the oscillation change in response to surface-related force gradients [33]. For simplicity, we chose to neglect changes in the oscillation phase and in the cantilever's resonance frequency. We implemented a feedback loop based only on the probe's oscillation amplitude, since

¹This interpretation of split photodiode signals is inexact. The position along the split photodiode position detector is given by $(V_{high} - V_{low}) / (V_{high} + V_{low})$; however, single-chip divider circuits with a bandwidth sufficient for tapping-mode frequencies (up to 350 kHz) were either unavailable or unreasonably expensive.

the goal of the instrument was likely to be fulfilled by such a simplified approach.

The first and most critical part of the feedback electronics system generates a variable offset that depends on the probe's oscillation amplitude. The signals V_{high} and V_{low} from the photodetector are used to generate an amplitude signal, which is used by the proportional-integral correction circuit to generate a correction signal. This correction signal is fed back into the system via the probe holder piezoelectric stack, where it translates into a change in the height of the AFM probe, thus restoring its oscillation amplitude to a target value. A block diagram of the correction circuit appears in Figure 2.4.

A second circuit provides signals to the piezoelectric stack located on the probe holder (see Section 2.1.1.2). It adds together three signals. The first is an oscillation that drives the AFM cantilever at its resonance frequency; we used one of the DS3XX series of function generators made by Stanford Research Systems to generate this oscillation. The second is a manual offset that allows the operator to move the probe up and down during the tip-approach and tip-disengagement steps of instrument operation. The third is the variable offset provided by the correction circuit. The application of the variable offset closes the feedback loop, allowing the oscillating AFM probe to remain at a constant average height with respect to the sample surface.

2.1.2 Inverted Optical Microscope

Optical observations in the FANSOM are made with a sample-scanning inverted optical microscope. Fluorescence is excited using a green helium-neon (HeNe) laser emitting at 543 nm, collected and observed using two sensors, a Geiger-mode avalanche photodiode (SPCM-AQ 131, made by EG&G Canada²) and a charge-coupled device (CCD) camera from Philips.

The illumination system first expands the 543 nm beam from the laser such that the expanded beam nearly fills the back aperture of the objective. This helps in obtaining a tightly focused laser spot. The beam is steered into alignment using a combination of a mirror and a dichroic filter (see Figure 2.5). It is important to note that, in the case of a helium-neon laser used to excite fluorescence, it is imperative to filter the beam to eliminate any photons from the 633 nm resonance. Those photons are present in the laser beam in sufficient numbers to completely drown the fluorescence signal. Fortunately, they are very easy to eliminate using a bandpass filter; we used a bandpass filter centered at 540 nm with a bandwidth of 10 nm.

A critical part of the system is the microscope objective. We initially chose an infinity-corrected 60X objective (Olympus Plan-Apo 60X) for its large N. A. of 1.4, which describes a cone with a half-angle of 1.24 radians. Such a cone subtends a solid angle of 1.35π steradians, or 33.7% of a spherical surface. In contrast, an N. A. of 1.3 subtends 26.1% of a spherical surface. The focus of the objective

²EG&G has been acquired by Perkin-Elmer, and is now known as Perkin-Elmer Optoelectronics.

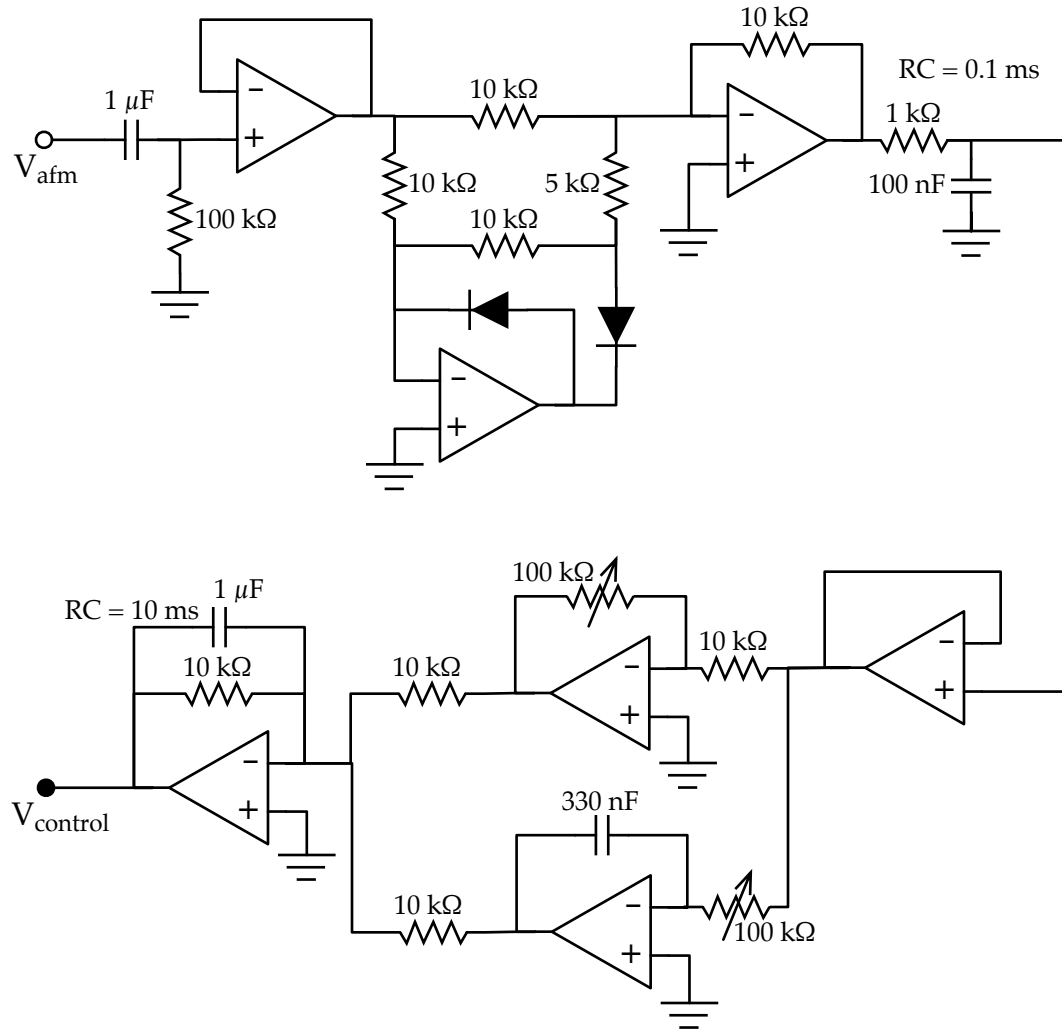


Figure 2.4: Diagram of the correction circuit for the tapping mode AFM feedback system. The input from the AFM is $(V_{high} - V_{low})$. The top part of the circuit is composed of a high-pass filter for removal of the DC offset from the probe signal, with an active full rectifier circuit whose output goes to an energy-storing RC filter. The bottom of the circuit is the proportional-integral correction circuit proper, with its output low-passed to a bandwidth of 160 Hz.

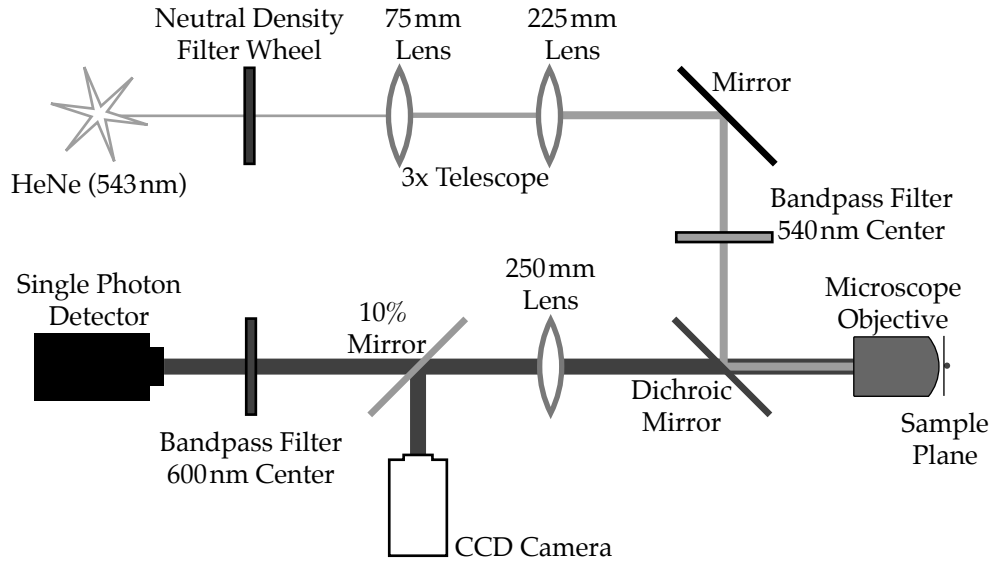


Figure 2.5: Diagram of the inverted optical microscope.

can be adjusted by a combination piezoelectric stack and manual adjustment screw that actuates a vertically mounted translation stage.

The fluorescence is generated at a glass-air interface located at the focal plane of the objective. A substantial proportion of the emitted fluorescent photons travel back through the objective, along with some of the green light reflected at the interface. The light exiting from the back of the microscope objective is filtered initially by a dichroic beamsplitter that transmits 90% of the fluorescent photons while reflecting 95% of the 543 nm photons from the HeNe laser. The remaining light then gets focused onto the photodiode by a positive lens of focal length 250 mm. A beamsplitter diverts 10% of the power to the camera, while 90% goes to the avalanche photodiode. Prior to reaching the photodiode, the light finally goes through a highly selective bandpass filter (see Figure 2.6).

The magnification of the imaging system formed by the objective and the 250 mm lens is 94; Olympus calibrates its objectives for a 160 mm tube lens, and a longer tube length results in a higher magnification.

This optical microscope is very selective in terms of wavelength, thanks to the use of high quality filters. It is also an efficient collector for fluorescent photons emitted at the sample plane. An objective with an N.A. of 1.4 collects approximately 33% of the fluorescence emitted by a sample. The avalanche photodiode detector (SPCM-AQ 131) has a $10\text{ }\mu\text{m}$ square active area, and has a quantum efficiency of approximately 50%. With transmission inefficiencies through the objective, the lenses and the filters taken into account, approximately 8% of all fluorescent photons emitted by a sample are detected.

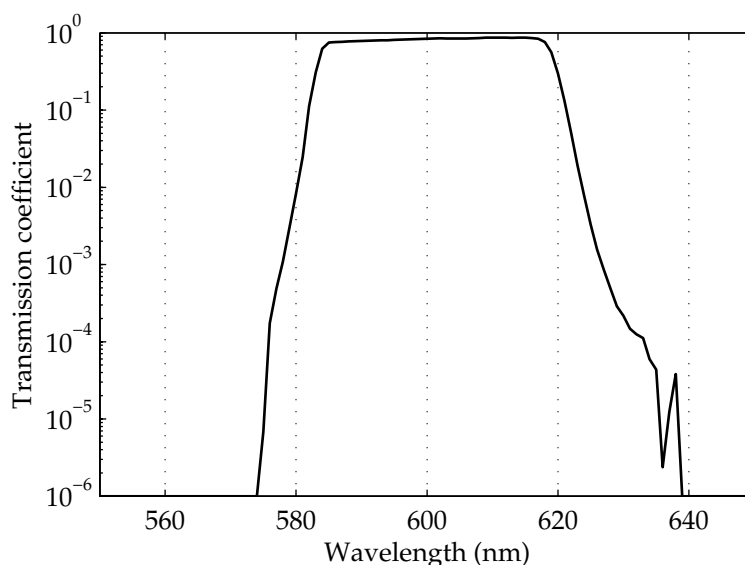


Figure 2.6: Transmission spectrum of the Chroma D600/40M filter (batch number 11915). Two filters like this were stacked in front of the photodiode, in order to block photons that did not originate from the sample fluorescence.

2.1.3 Data Acquisition

Data acquisition in the FANSOM is driven by a software application, with assistance from various hardware. The hardware consisted of a National Instruments™ AT-MIO-16E-1 card, a National Instruments™ PCI-GPIB card, a Stanford Research Systems SR400 event counter, and some home-built circuits. The signal outputs were handled by the AT-MIO-16E-1 and a National Instruments™ Lab-PC-1200 card; their 12-bit outputs drove the fine positioning of the scanning stage and of the AFM probe holder stage, respectively.

The fluid nature of software led to the data acquisition end of the FANSOM varying greatly over the life of the project. This section describes the AFM driver and the two versions of the FANSOM data acquisition software that led to the most interesting results from this embodiment of the FANSOM.

2.1.3.1 AFM Driver and Topographical Data Acquisition

The driver software for the AFM is conceptually separated into two parts: stage movement (signal output) and data acquisition (signal input). The two modules are independent and synchronized.

The stage movement module performs line scans of the X axis, between which it moves the Y axis incrementally. All movements are designed to be smooth, because it was observed that abrupt movements can worsen the drift problems of the stage as well as accelerate the wear on the AFM probe. Consequently, all stage movements are done at a constant speed during a given run of the

AFM.

The data acquisition module samples one channel of topographic data, and acquires each line of data in synchrony with the stage movement. The topographic data is acquired from the output of the correction circuit (see Section 2.1.1.4), which closely mimics the height variations of the surface over which the AFM probe is scanned. The data is written out to a file after the entire image has been acquired; the pixel values are 16-bit integers, of which only the lower 12 bits are used, because the AT-MIO-16E-1 card has 12-bit analog-to-digital (A/D) converters.

2.1.3.2 FANSOM Data Acquisition

The basic idea of data acquisition in the FANSOM is to expose a modulation of an optical signal at a known frequency, namely the frequency of oscillation of the AFM probe. This modulation was originally detected by observing the output of a photodiode using a spectrum analyzer. From an imaging standpoint, however, a software-based spectrum analyzer would be much too slow to be practical. Another method had to be used to extract the modulation component from the optical signal.

Because of speed considerations, it was impossible to use the computer directly to extract the modulation component. It was thus necessary to use an external method of extracting the FANSOM data. At first, we used an analog demodulation circuit; later, we used an event counter controlled and polled by the computer. A description of each of these two methods follows.

Analog FANSOM The analog method of acquiring the FANSOM data was based on a simple AM-style demodulation circuit, with a low-pass filter on the output. This demodulation circuit had a twist: the use of the position signal of the AFM probe as the “local oscillator”. The reason for this is simple: the element that introduces the sought-after modulation into the optical signal is the probe itself, by virtue of its position. It is thus necessary to correlate the position of the probe with the optical signal in order to extract a probe-induced modulation.

Using the probe position signal would allow the detection of some modulation of the optical signal, but would also superimpose transient variations in the amplitude of the AFM probe’s oscillation. These transient variations are unavoidable: they are the error signal that our feedback system works to minimize—and a feedback circuit without any error is impossible. A phase-locked loop circuit (PLL) was thus added in order to translate the probe position into a signal of constant amplitude. A PLL always has a certain phase lag to the signal it chases, and that phase lag is a function of frequency. Given that the probe oscillates at a constant frequency during the acquisition of a given image, the PLL lag is a constant and therefore is not a problem.

It must be noted that using the driving signal directly from the function generator as the “local oscillator” would be incorrect. To do so would bypass the phase shifts that occur in the AFM’s oscil-

lation as its probe goes over obstacles and material changes. These phase shifts must be preserved because the optical modulation is a direct consequence of the probe's presence; as the modulating element's phase shifts relative to the driving signal, so does the phase of the modulation. The value of a phase-locked loop in this case is that it preserves the phase of the modulating element without introducing amplitude variations. Without the PLL, amplitude variations would carry right to the output of the demodulation circuit, and their effect would easily dwarf the contribution from the optical modulation we seek to collect.

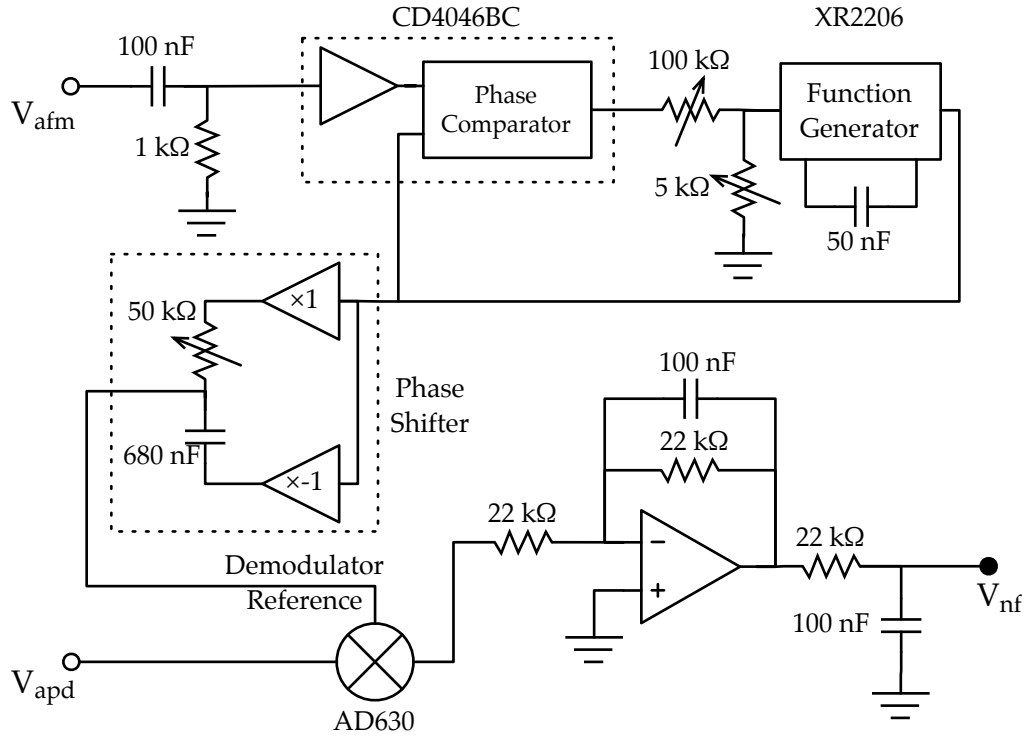


Figure 2.7: Diagram of the demodulation circuit for analog FANSOM operation. The top of the diagram is a phase-locked loop (PLL) that follows the phase of the AFM probe position signal. Its output is used as the reference of a demodulator circuit whose bandpass is 72 Hz. The capture band of the phase-locked loop was adjusted to range from 240 to 330 kHz, a band larger than the typical range of variation of TESP probe resonance frequencies.

A diagram of the analog demodulation circuit is shown in Figure 2.7. Three non-trivial integrated circuits were used in its construction. The CD4046BC is a digital phase-locked loop sold by Fairchild Semiconductor™. It includes its on-board voltage-controlled oscillator (VCO), but it outputs square waves only; we instead used the XR-2206 by Exar Corporation as a VCO. The XR-2206 is a function generator chip capable of producing sine waves with a distortion of 1% or less. The capture band of the VCO is determined by the 5 kΩ and 100 kΩ variable resistors in the circuit; they determine the lower bound and the width of the capture band, respectively. The phase of the

XR-2206 output is modified by through an adjustable phase-shifter circuit before being used as the demodulator reference.

The phase of the PLL output needs to be adjusted in order to obtain optimal response from the lock-in amplifier. The adjustment needs to be done once for every probe used, and it requires the presence of some FANSOM signal first; fortunately, the adjustment isn't large from probe to probe—a mere tweak. The reference and the input signal from the photon detector are multiplied together, with the effect that the probe-induced modulation in the photon detector output is frequency-shifted to the base band. The output of the mixer is then low-pass filtered; we chose a simple double-pole low-pass filter with a bandwidth of ~ 72 Hz.

The output of this circuit was read by the FANSOM driver through an A/D converter, using the same method as was used to acquire the AFM topography signal. The FANSOM data was also saved to images in a data format identical to the format used for AFM data.

Encouraging initial results were obtained with this scheme for FANSOM data acquisition, but its limitations proved difficult to overcome. Attempts at separating the FANSOM data from the background noise using analog electronics proved unreliable at best, but was a necessary step to go forward. The discovery of an event counter among equipment donated to the lab allowed us to make that step forward.

Digital FANSOM The digital method of FANSOM data acquisition was developed around the SR400 event counter by Stanford Research Systems. That instrument is equipped with a collection of edge detection inputs and two independently gated counters.

With appropriate adjustments to the gate timers, it was possible to make the counters detect input events during windows of 0.2 to 1 μ s, timed such that counter A would count photon events while the AFM probe is close to the surface, and counter B would count those events while the AFM probe is away from the surface. These windows are triggered by the output of the phase-locked loop portion of the analog FANSOM's demodulation circuit, and as such are synchronized with the cantilever motion. The binning of photon counts into pixels was in turn triggered by a pulse train, generated by the driver software using the digital outputs of the AT-MIO-16E-1 card.

The digital FANSOM data was saved very differently from the AFM topography. The SR400 event counter was programmed via the General-Purpose Interface Bus (GPIB), and the collected data was downloaded to the computer also via GPIB.

The SR400 event counter can memorize the values of up to 2000 bins per counter, which is substantially fewer than required for a high-resolution image. Therefore, it was necessary to extract these values from the counter at the end of each line.

At the end of an image acquisition, the images corresponding to the two channels ("near" and "far" images) were saved to disk. The file format of these images is similar to that of the AFM

images, except that the pixel data are 32-bit values.

2.1.3.3 Data Handling and Visualization

All the data produced by the homemade FANSOM was saved in binary files, and can be displayed using MATLAB®, using a collection of scripts written for this purpose. MATLAB® was mainly used for its flexibility, and the ease with which it can transition from being an everyday visualization tool to producing quality images or graphs.

The MATLAB® functions that read the AFM images and the optical images acquired by FANSOM driver are described and listed in Section D.1.1.

2.1.4 Alignment of the FANSOM

Due to the geometry of the FANSOM based on the home-built AFM, alignment of an AFM probe above the laser beam was relatively difficult. Since the laser was fixed, the probe had to be positioned as close as possible to the beam’s focal spot—an area of a few hundred square nanometers.

The alignment was done in a multi-stage process, the first of which is an alignment by hand. The probe was brought close to the location of the laser spot using visual clues, by turning the actuator screws of the AFM probe holder stage. With practice, we were able to repeatably bring the probe to within one micrometer of the center of the beam. The next stages consisted of repeated acquisition of simultaneous AFM and optical images. After imaging a sparse sample (3 to 5 objects per $25\text{ }\mu\text{m}^2$), the AFM and optical images were correlated by eye and, using the piezo-electric actuators, a correction was applied to the positioning of the AFM probe so as to make it coincide with the laser focal spot. The alignment was then verified, and corrections applied as necessary.

The image correlation step of the alignment procedure required samples that exhibit relatively tall topography, so that the AFM image would contain information that can be used for the correlation. Unfortunately, this means that it would be difficult or impossible to perform the alignment for samples with very little topography, such as single fluorophores.

2.2 Experimental Results

The samples used for the results discussed in this section were fluorescently dyed latex beads, with typical diameters of 100 nm or 60 nm. The beads, the preparation of the bead samples and the glass cleaning method are described in Appendix B.2.

The bead-covered cover glass was clamped down on top of the microscope objective, then imaged using the AFM and/or the optical microscope. The probes referred to below are described

in Appendix A. The results presented in this section have previously been published in a different form [34, 35].

2.2.1 Early Results

The earliest evidence of subwavelength imaging obtained from the FANSOM was obtained with a silicon probe over a relatively dense sample of dyed fluorescence beads. Unfortunately, FANSOM contrast proved very difficult to obtain by using silicon probes as modulation elements. The result was not reproduced for several weeks.

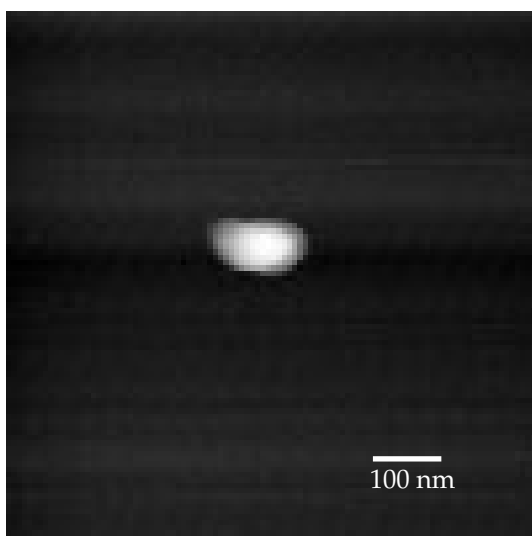
More easily reproducible results were obtained using gold-covered probes. These probes were fabricated by evaporating a thin layer of gold onto a standard silicon TESP probe. Film thicknesses from 10 to 30 nm were used.

With the analog version of the FANSOM described in Section 2.1.3.2, the near-field optical data is for all practical purposes a 150 Hz wide spectral band from the power spectrum of the photon detector output, centered at f_0 , the probe's oscillation frequency.

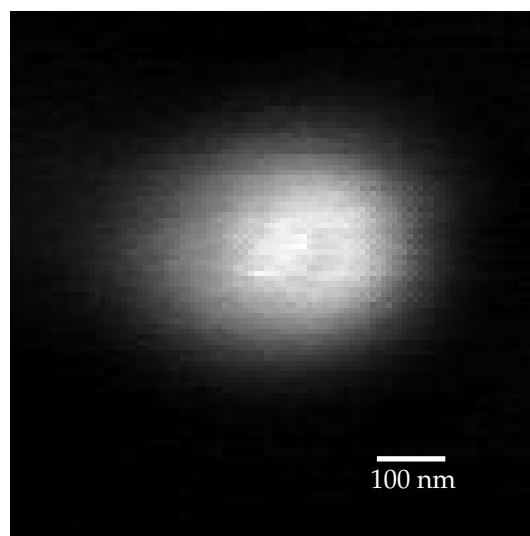
Consider that the output of the photon detector is a collection of short pulses (10 ns in duration) with random arrival times. This randomness means that, at frequencies lower than a few megahertz, some energy is spread over the entire power spectrum in an amount proportional to the total intensity. In other words, the energy measured at f_0 will comprise not only the modulated energy from a near-field interaction, but also a "background" intensity. This can be seen easily in Figure 2.8, which shows a FANSOM image along with simultaneously acquired AFM and far-field optical images.

Power spectra of the fluorescence were taken to determine whether photobleaching rates were affected by the near-field interactions. The power spectrum of the photon detector output was measured and saved at repeated intervals, and the height of the probe-induced peak was measured against the noise level (see Figure 2.10). The greater slope of the peak decay as compared to the background indicates that the decay rate of the modulated energy is greater than the decay rate of the unmodulated energy. This result suggests a localized increase in the photobleaching rate of the dye.

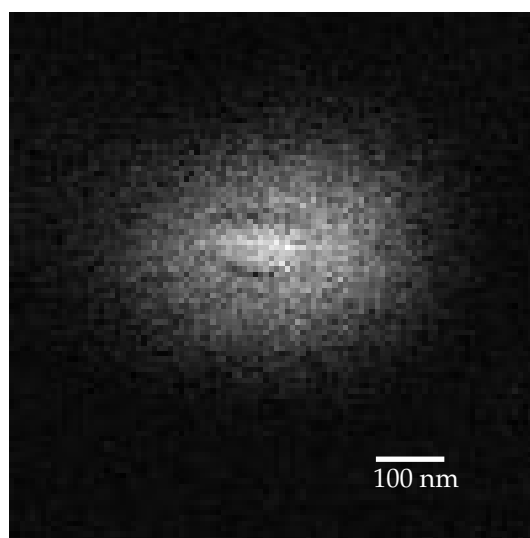
The presence of the background in the FANSOM image illustrates the shortcomings of the analog FANSOM data acquisition scheme. After an attempt to balance two lock-in amplifiers with diametrically opposed phases bore less fruit than expected, the FANSOM was reworked to use the SR400 event counter by Stanford Research Systems, as described in Section 2.1.3.2.



(a)

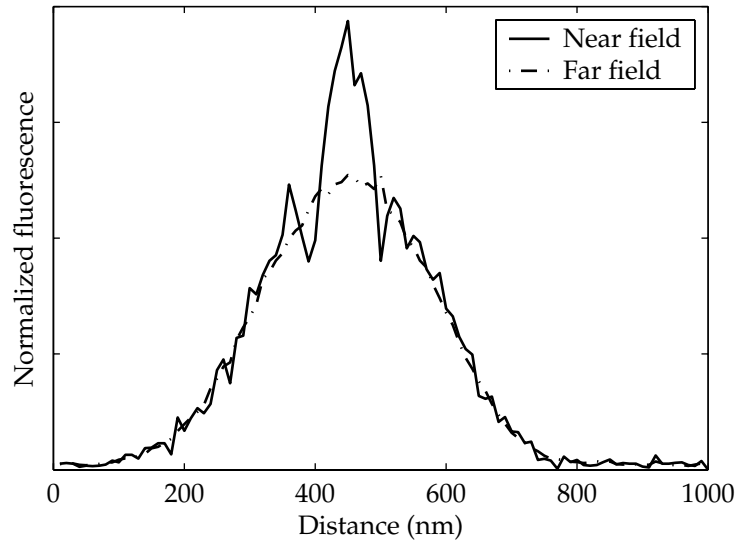


(b)

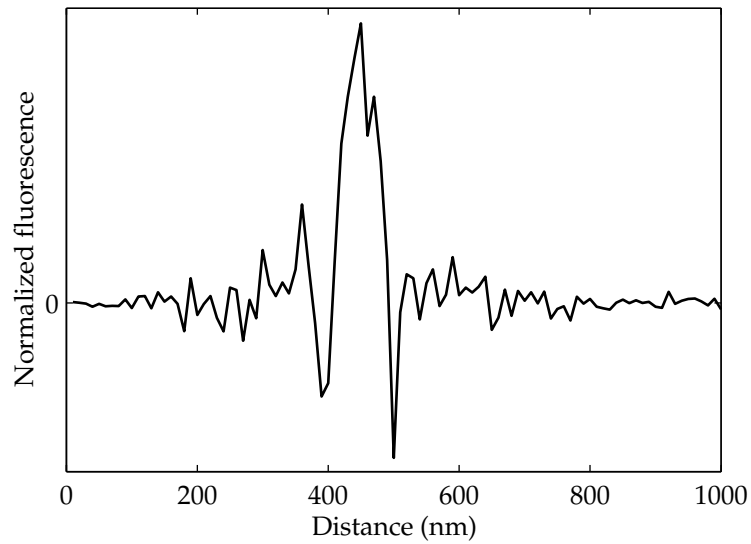


(c)

Figure 2.8: Images of a fluorescent 100 nm bead taken with the analog FANSOM, using a gold-coated TESP probe. The images represent $1\text{ }\mu\text{m}$ squares. (a) is the AFM image, (b) is the far-field optical image, (c) is the FANSOM image, which shows subwavelength optical contrast. The FANSOM contrast is seen as a hump on top of a background identical to the far-field image.



(a) Normalized comparison of vertical cross sections of the FANSOM and far-field images. The two curves follow each other closely until the central 100 nm. At the position of the observed bead, the near-field image exhibits a sharp local contrast.



(b) Difference between the two curves of 2.9(a). The difference is small until the central 100 nm, the position of the observed bead.

Figure 2.9: Comparison of vertical cross sections of the FANSOM and far-field images of Figure 2.8. The FANSOM image shows contrast at a resolution much better than the diffraction limit.

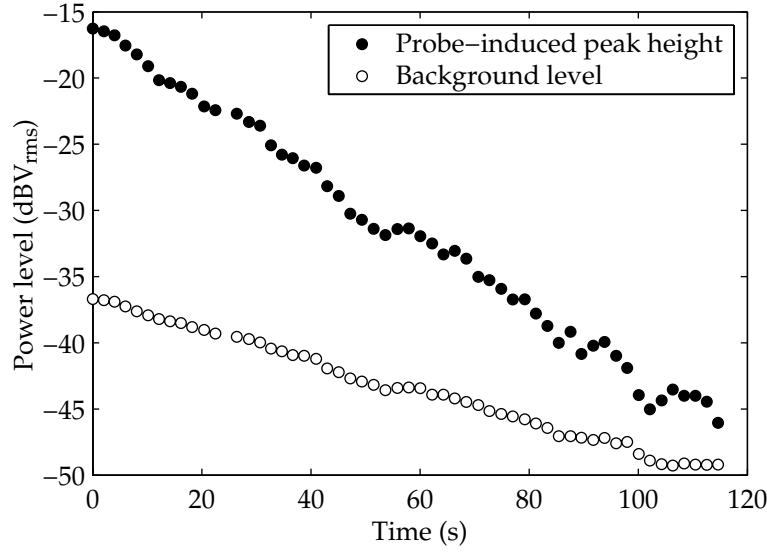


Figure 2.10: Decay of the modulation component in the fluorescence power spectrum as a function of time. A clear peak appears at the resonant frequency of the cantilever, and its decay over time has a rate greater than that of the background fluorescence.

2.2.2 FANSOM with Gated Photon Counting

The first major result obtained with the digital FANSOM scheme was that the contrast was obtained through a proximity-induced inhibition of fluorescence, rather than the enhancement we had at first expected to see. Comparing the pixel data from counter A (with the probe close to the sample) to the data from counter B (with the probe “far” from the sample), they had similar count levels except at pixels when the probe was over a bead; counter A would then count substantially fewer detected photons than counter B. We had subwavelength contrast and, by combining the A and B images, were able to add far-field rejection to the FANSOM (see Figure 2.11).

The real power of the method, however, proved to be in the ability to correlate photon arrival times with the probe position. The SR400 unit allowed the observation of samples through very narrow, scanned time windows.

By programming the event counter to scan along the phase of the probe oscillation once per line, it became possible to map the relationship between the observed fluorescence inhibition to the distance between the probe and the sample. Since the amplitude of the probe’s tapping motion was known, it was possible to map between time and the probe’s vertical displacement and obtain “approach curves” (see Figure 2.12).

We scanned the phase space in 200 ns increments with a gate width of 400 ns, for a variety of probe surfaces (bare silicon, gold) and bead diameters (100 nm, 60 nm). This showed that in the case

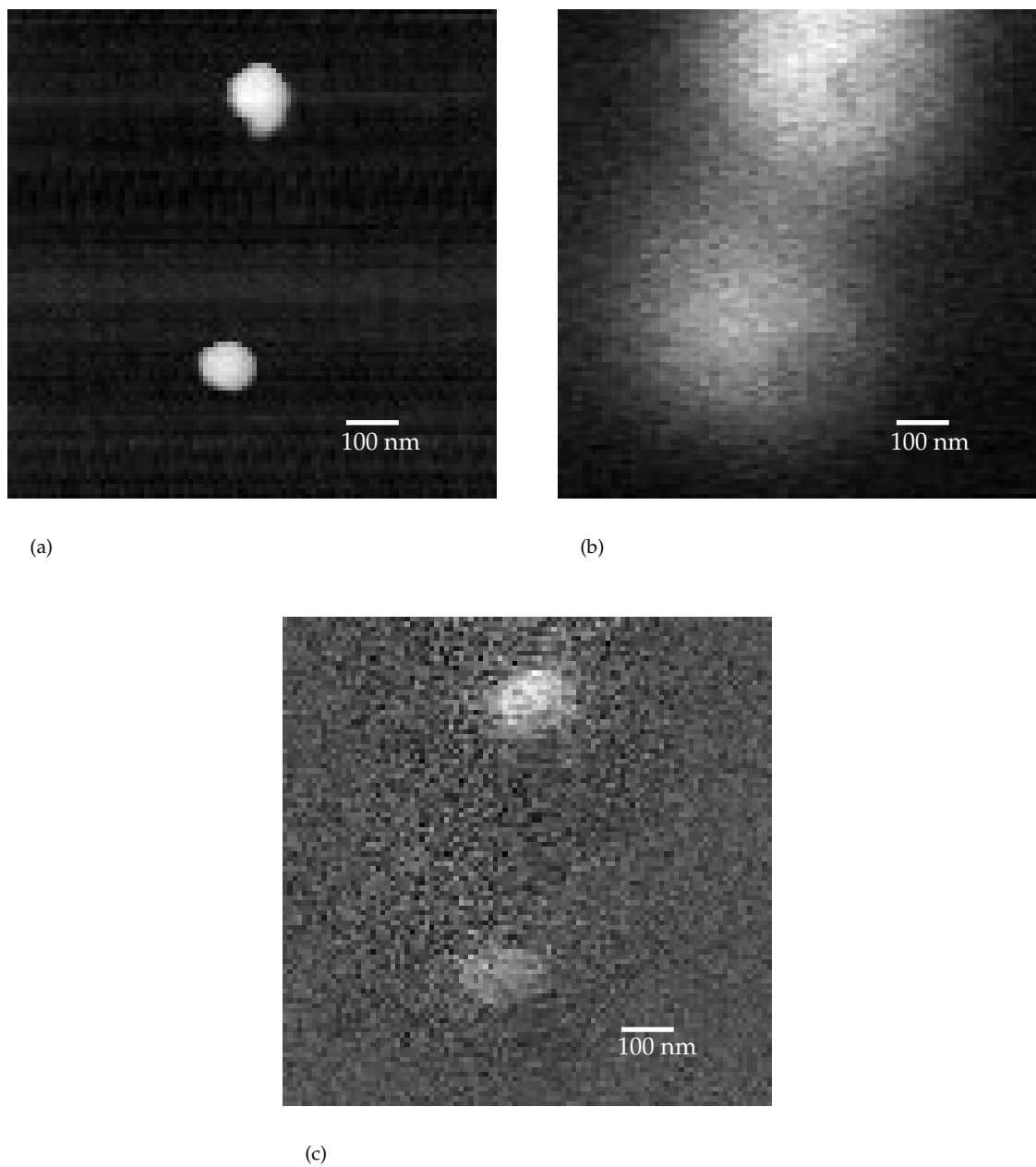
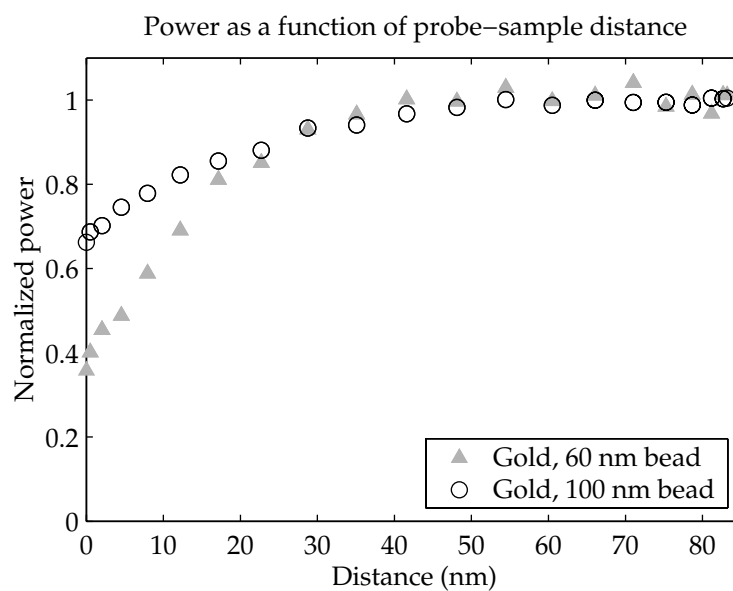
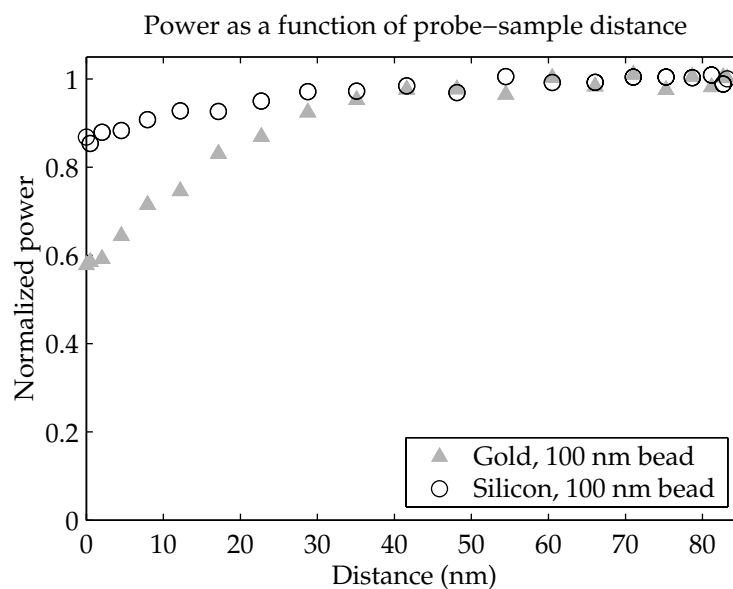


Figure 2.11: Images of two fluorescent 60 nm beads taken with the digital FANSOM. (a) is the AFM image, (b) is the far-field optical image, (c) is the FANSOM image.



(a) Comparison of approach curves for gold-coated probes above 100 nm and 60 nm beads.



(b) Comparison of approach curves for a gold-coated probe vs. a bare silicon probe.

Figure 2.12: Normalized fluorescence as a function of probe-sample distance. The fluorescence decreases as the probe approaches the sample.

of gold probes, the inhibition of fluorescence becomes very strong for vertical separations smaller than 20 nm. As expected, the relative fluorescence decrease is smaller for 100 nm beads than for 60 nm beads, but the amount of the decrease does not seem to scale straightforwardly with the bead volume: a 60 nm bead has 22% of the volume of a 100 nm bead. If the fluorescence decrease was purely a volume effect, the normalized minima for the two bead sizes would be much more different than those observed. This suggests that quenching of fluorescence by the metal may not be the only mechanism at work.

2.3 Limitations

Through the use of home-built hardware and software, we were able to obtain several datasets demonstrating subwavelength contrast with fluorescent samples. However, this home-built hardware had limitations that needed to be overcome in order to delve deeper into the physics of the problem. These limitations were mostly related to spatial stability and resolution, but also to a lesser extent to temporal resolution.

2.3.1 Atomic Force Microscope

Our AFM sample stage was driven by a National Instruments™ card, the AT-MIO-16E-1. This card's digital-to-analog converters are 12-bit chips, and this led to a fairly coarse positioning grid. Our piezoelectric stacks had a travel of 15 μm ; given the 2^{12} (4096) addressable positions, this meant that, at best, the addressable points were 3.7 nm apart. At first blush, this seems sufficient, but since we were able to obtain resolutions better than 30 nm in early attempts, the positioning accuracy would clearly have become a limitation.

Another limiting factor of the microscope was the amount of wear it imposed on probes. The probes rapidly got to the point where their sharpness was reduced to a radius of curvature greater than 20 or 30 nm. This was caused by noise both in the electrical circuits and noise in the environment. While the electrical noise could have been much improved, the large size of the microscope made it very difficult to isolate from the environment.

Because of these and other factors, the resolution of our AFM was limited to 20 nm at optimum operation, and it would routinely deteriorate to 40 nm or worse. In order to obtain an optical resolution of 10 nm with the FANSOM, a great improvement of the accuracy and resolution capability of the AFM was needed.

2.3.2 Alignment and Drift

The alignment procedure of the FANSOM was briefly described in Section 2.1.4. While the procedure sounds simple and straightforward, in practice it took too long and was too fickle.

As previously discussed, imaging repeatedly with the AFM resulted in probe wear; that alone made obtaining high-resolution results difficult. In addition, the large size of the whole apparatus made it especially susceptible to thermal drift, leading to situations where the alignment procedure would become a long chase at the end of which the probe was blunted, effort wasted, even before any meaningful data had been obtained.

It would be desirable to improve the method of alignment, and even more desirable to reduce the amount of thermal drift in the system.

2.3.3 Limited Phase Sensitivity

The phase sensitivity obtained thanks to the flexibility of the SR400 event counter was important in starting to understand the contrast mechanisms of the FANSOM, but it was clear that this flexibility paled in comparison to what would be feasible if event counting was done directly by the computer. This was not possible, however, because the AT-MIO-16E-1 card's counters and the SPCM-AQ 131 did not work well together, the latter's pulses being too narrow for detection by the former.

2.4 Conclusion

The early work on the FANSOM apparatus was sufficient to demonstrate that it is possible to obtain subwavelength fluorescence contrast with apertureless methods, and indeed implement a microscope with such a contrast mechanism. It also demonstrated that many improvements needed to be developed in order to obtain much higher resolutions. Chapter 3 describes a completely overhauled apparatus that produced improved results.

Chapter 3

Second-Generation FANSOM

In order to reach a resolution of 10 nm, a FANSOM apparatus would need to meet stricter performance criteria than the original apparatus in many aspects of its operation. One of those aspects is the ability of the contrast mechanism to perform at that resolution—we speculate that 10 nm resolution is possible, and until the exact performance of the contrast mechanism is known, the rest of the system must be designed as stringently as possible.

There are, as before, two microscopes to put together: an AFM and an inverted optical microscope. Both need to be stable and need to behave predictably when scanning an image. Any departure from perfection would detract from the result obtained by the apparatus, and would take away from our ability to objectively evaluate the contrast mechanism. A picture of the two microscopes put together, before the soundproofing and light proofing was done, appears in Figure 3.1.

3.1 Atomic Force Microscope

The AFM we chose to reimplement the FANSOM is a system built and sold by Digital Instruments (DI), composed of a BioScope with a Dimension scanner, with a NanoScope[®] III controller. This was later upgraded to a Dimension CL closed-loop scanner with a NanoScope[®] IV controller.

This scanner has a large scanning area (100 μm square). It is a probe-scanning system, i.e., the probe is scanned over a stationary sample. Note that if the option had been available, a sample-scanning microscope would surely have been preferred. However, in early 2001, the AFM market was not very competitive and the choices were between one established supplier (Digital Instruments) and several fledgling companies. DI did not (and still doesn't) offer a sample-scanning system, so the probe-scanning system was acquired.

The DI microscope solves most of the probe wear problems of the homemade AFM thanks to its sophisticated feedback electronics, and helps avoid some of the alignment difficulties thanks to the openness and compactness of the geometry of its scanning head. In Figure 3.1, the AFM scanning head is viewed from the side, and the AFM cantilever is pointed toward the camera; this

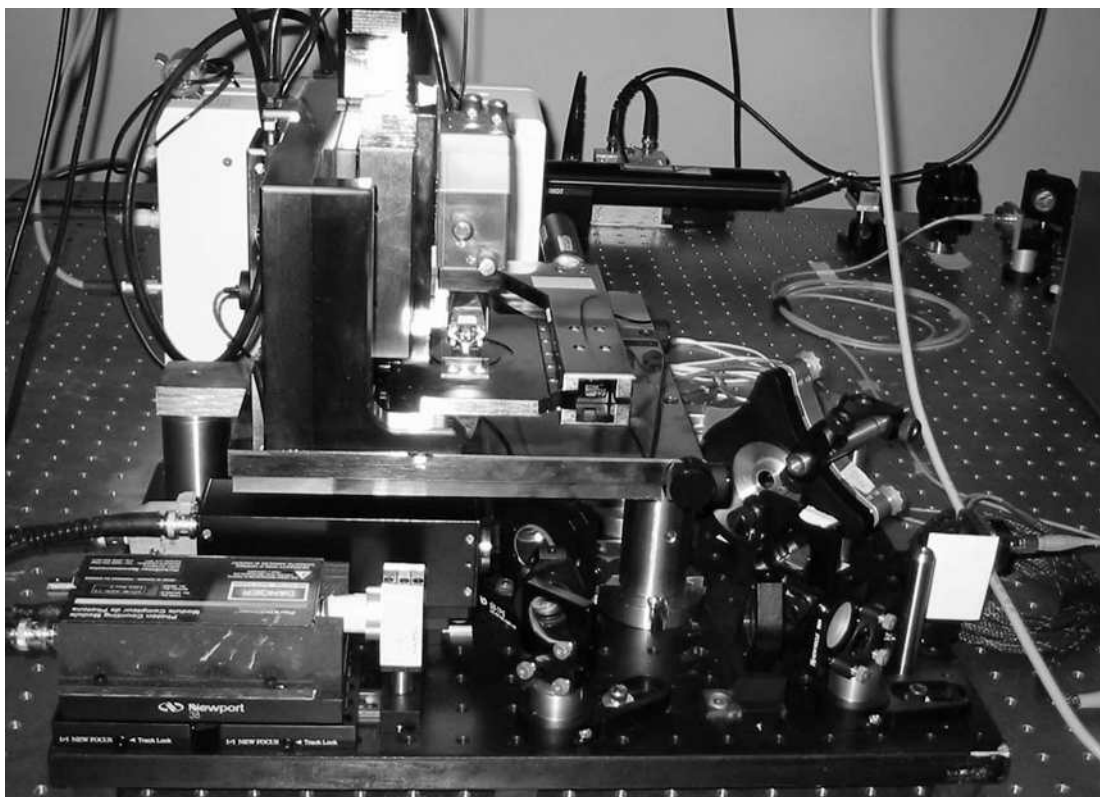


Figure 3.1: Picture of the second-generation FANSOM apparatus.

is an excellent angle to illuminate the AFM probe head-on, a trick that has proven very useful in aligning the AFM to the laser beam.

3.1.1 Issues with the DI System

The system sold by Digital Instruments is a closed system, and incorporating it in another instrument presents difficulties. Relatively few of the control signals are readily available from the system, even when using the “signal access module”. Certainly, information such as the dimensions of the area being scanned is not available at the time of data acquisition.

Two of the signals that are available, however, proved very useful: they are two sets of timing pulses signaling the start of each scan line and each frame. These pulses are called EOL and EOF, respectively, by the DI equipment. These names seem like misnomers, but that is a matter of perception: the NanoScope[®] systems scan continually, so the end of one line (or frame) is the start of another. In the FANSOM, we use the timing pulses both as start pulses and end pulses since we only deal with one scan direction during a given data acquisition run.

A third signal needed for the FANSOM is a marker of the probe position, to enable us to correlate detected photons with the position of the probe. The best candidate for this is the output from

the quadrant photodiode that part of the AFM scanning head for probe tracking. This signal is extracted from the BioScope™ circuit board at the earliest possible point: directly after the current-to-voltage conversion and two adding stages. The extracted signal is then transformed from an analog waveform to a pulse train by a high-speed comparator circuit. A pulse is output by the circuit every time the probe passes through the midpoint of its oscillation; the timing is not exact, but the error is a fixed delay of less than 100 ns, with no discernible dependency on the probe's resonance frequency in the range from 60 kHz to 350 kHz.

3.1.2 Vibration Isolation

In the homemade AFM, one of the main problems was the probe wear. One component that always affects the probe wear in atomic-force microscopy is environmental noise. Indeed, DI offers a range of enclosures to isolate their AFMs from noise.

In the case of the BioScope, this is typically accomplished by using a large enclosure that comprises a small optical table, on which an inverted microscope is mounted, and then the BioScope itself. Since we already had a very good optical table, we chose to simply build isolation around the system using specialty sound-insulating foams fabricated by EAR Specialty Composites. This also means that the AFM must somehow be insulated from the optical table itself, since a substantial amount of energy could be transmitted acoustically to the AFM through the surface of the optical table. The vibration isolation from the table is described as part of the optical microscope description.

3.1.3 Suppression of Amplified Spontaneous Emission

The Dimension and Dimension CL use a low-cost laser diode for probe tracking, just like our homemade AFM did. Similarly, it necessitated optical blocking away from the laser line in order to avoid drowning our signal in the ASE noise. This filter must be located between the laser and the AFM probe, which means that it must be placed inside the AFM head. The simultaneous small dimension requirements and optical blocking requirements on the filter itself presented a challenge that the engineers at Chroma Technology handled brilliantly. The logistics of inserting the filter inside the AFM head unfortunately proved to be the worst part of the challenge, requiring multiple trips to DI headquarters. Figure 3.2 shows the transmission spectrum of the clean-up filter we inserted in the AFM scanning head.

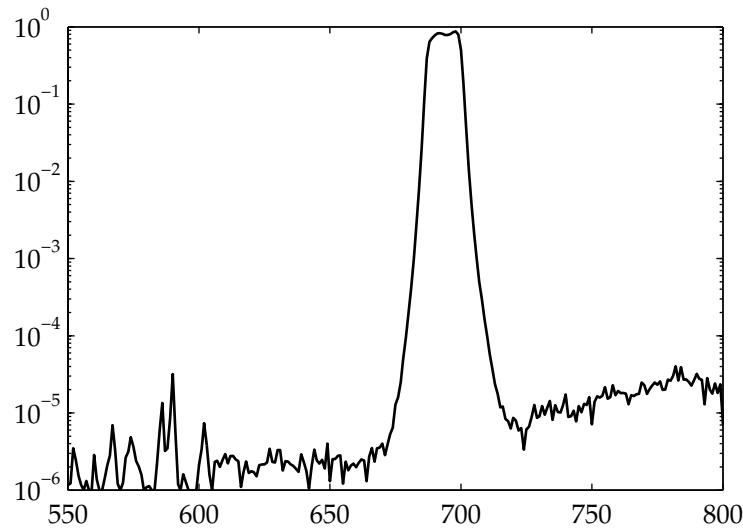


Figure 3.2: Transmission Spectrum of the optical filter (Chroma EB692/15, batch number 102494) used as a clean-up filter for the laser diode in the Dimension AFM scanning head. The energy present outside of the laser line, if left unblocked, would be sufficient to drown the signals of interest in the optical end of the FANSOM.

3.2 Inverted Optical Microscope

A custom optical microscope was built to interface to the BioScope system. In principle it is very similar to the one built for the original FANSOM, but a number of details were changed and improved; we also added new features such as vibration isolation, beam steering and beam shaping.

3.2.1 Vibration Isolation

As stated earlier, the AFM must be isolated from the optical table. However, the AFM is installed on top of our inverted optical microscope, so this requirement implies isolating the optical microscope from the table as well. We accomplished this isolation by using an optical fiber cable to decouple the illumination laser alignment from the microscope itself. The optical fiber used is single-mode at 543 nm, our illumination wavelength. A convenient side effect of the optical fiber is that it acts as a blocking filter against energy from the 633 nm resonance of the HeNe laser, eliminating the need to insert a filter specifically to block that wavelength.

The optical microscope is built on top of a relatively small breadboard (24 inches by 18 inches), which is isolated from the optical table itself by rubber pads. The pads are made from the specialty rubber Sorbothane[®], an effective material for shock damping and vibration isolation.

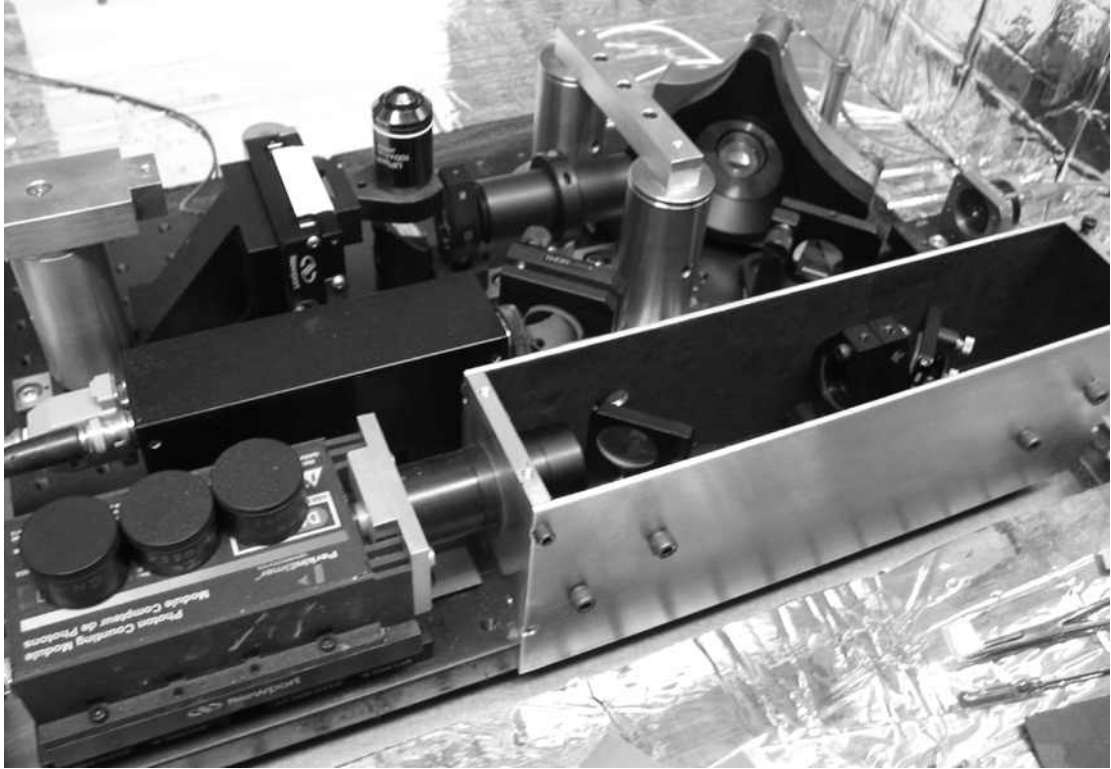


Figure 3.3: Picture of the optical microscope setup in the second-generation FANSOM. The excitation laser enters from the right, through an optical fiber, and is directed to the microscope through a 1:1 telescope. The fluorescence is detected by the Perkin-Elmer™ detector at the bottom left. The light from the microscope objective enters the dark box through a hole that is not visible on the picture.

3.2.2 Illumination and Beam Steering

The illumination beam is introduced onto the optical breadboard using an OFR FiberBench®, which allows for easy insertion of optical elements in the beam path. Thereafter, the beam is reflected towards the microscope objective by the dichroic mirror and the beam-steering mirror.

Beam steering is accomplished by taking advantage of a 1:1 telescope. The back aperture of the microscope objective is located one focal length away from one telescope lens and the beam-steering mirror one focal length away from the other telescope lens. Small deflections of the mirror result in a change in the angle of the beam at the back aperture of the objective, without changing the location of the beam. By changing the angle at which the beam enters the back aperture of the microscope objective, the location of the spot in the sample plane is changed. The scanning system we built allows the beam to be scanned over an area of $6.6\text{ }\mu\text{m}$ by $8.0\text{ }\mu\text{m}$; the driver software is limited to scanning a square area of $6.6\text{ }\mu\text{m}$ on a side, for simplicity.

The beam steering mirror itself is mounted on a two-axis tip-tilt actuator built by Mad City Labs, model number Nano-MTA2. Coarse positioning of the mirror is done using a Newport ULTIMA®

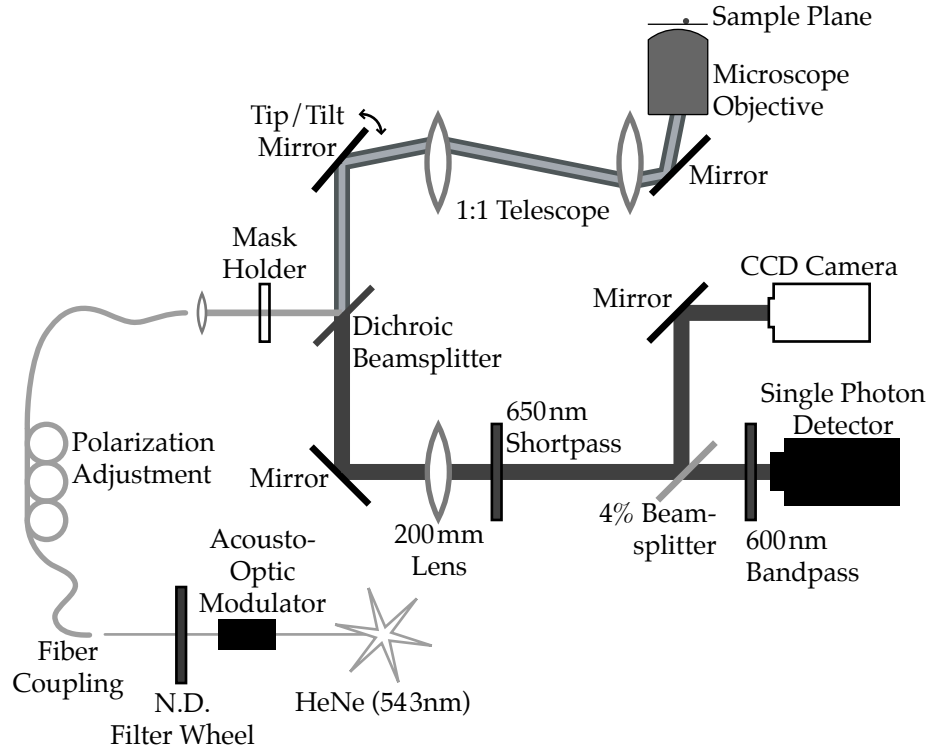


Figure 3.4: Diagram of the inverted optical microscope for the second-generation FANSOM. The tip/tilt mirror allows the laser spot to be scanned in the focal plane.

gimbal mount. The Nano-MTA2 actuator provides the fine positioning; it is a closed-loop actuator, allowing for reproducible positioning of the laser spot in a 2.2 mrad range of motion and 0.02 μ rad resolution. The coarse positioning is gimbal, but the fine positioning isn't—this is nearly imperceptible given the limited range of motion of the actuator.

Control of the tip-tilt mirror is done using a variation of the signal used to control stage actuation in the previous generation of the FANSOM. The National Instruments™ card used to output the signal is now model PCI 6052E, featuring 16-bit output capability. The granularity of the laser spot positioning using this combination is 1.2 Å—well below the level of other sources of noise, including the electronic noise in the mirror driver. This effectively eliminates the positioning granularity as a source of uncertainty in the system.

3.2.2.1 Beam Shaping

An important feature of this newer optical microscope is the ability to easily insert a mask in the path of the illumination beam, upstream of the dichroic filter. Inserting a mask does not affect the detection path, and allows for some shaping of the laser beam. We created some masks out of aluminum foil, and used them to perform a form of simple, high-loss beam shaping. This allowed

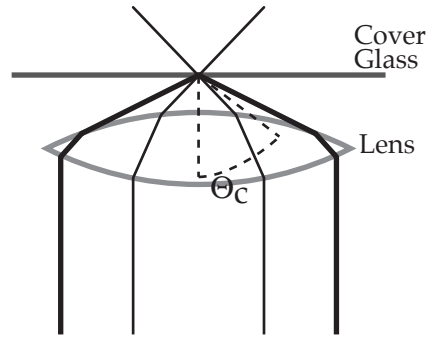


Figure 3.5: Full beam illumination at the sample plane. The volume between the lens and the cover glass is filled with immersion oil, and a portion of the light is incident upon the glass/air interface at an angle greater than the critical angle Θ_c .

us to isolate and study the effect of every polarization component of the illumination at the sample plane.

When the full beam is used, a portion of the illumination beam is incident upon the glass/air interface at an angle greater than the critical angle, due to the large numerical aperture of the microscope objective. This light is totally internally reflected at the sample interface, as illustrated in Figure 3.5. The totally internally reflected light makes it possible for a polarization component to be perpendicular to the sample plane and parallel to the optical axis. Figure 3.6 shows the behaviour of a totally internally reflected ray in two polarizations, to illustrate how the vertical polarization arises.

In the full beam, the contribution of the vertically polarized light is completely overpowered by the contribution of the light from the center of the beam, which is polarized parallel to the sample plane. We used masks inserted in the illumination path to isolate the various components.

The technique used to fabricate the masks out of aluminum foil is described in Appendix C. The masks were used in two different ways, depending on geometric constraints. In cases where it was sufficient to select only a part of the beam, e.g., to allow only the central part of the beam to go through, we used the foil directly. In the cases where we wanted to select a ring-shaped part of the beam, the foil was used as a mask to deposit an opaque gold pattern on an optical window.

The illumination modes that we have used in the FANSOM are the following:

- Direct illumination only: The mask is used to block the parts of the beam that would result in totally internally reflected rays. The spot at the sample plane has most of its energy polarized parallel to the sample plane.
- TIR illumination only: The reverse of the previous situation, a dot blocks the center of the beam. A ring-shaped beam reaches the back of the objective, and the spot at the sample plane has a complex shape involving destructive interference of vertically polarized components.

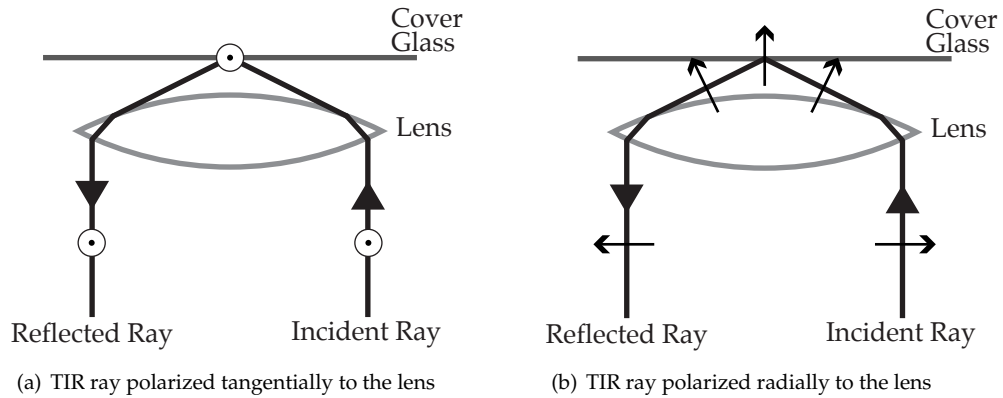


Figure 3.6: Polarization behaviour of a totally internally reflected ray. (a) shows an incident ray whose polarization is tangential to the edge of the lens (perpendicular to the page). At the sample plane, the polarization component is parallel to the sample plane. (b) shows an incident ray whose polarization is radial to the lens. At the sample plane, the polarization component is perpendicular to the sample plane. Note that two rays with the same polarization (along a diameter) entering from opposite edges of the lens will interfere destructively at the sample plane.

- Crescent illumination: Only a portion of the ring-shaped beam from the TIR illumination mode is allowed through. By rotating the polarization of the laser beam, the polarization at the sample plane can be selected to be perpendicular or parallel to the sample plane. The top two designs in Figure C.1 are used for crescent illumination.

3.2.3 Optical Detection System

The detection arm of the optical microscope is in principle identical to that of the earlier instrument. The beam steering system moves the focus of the detection system along with the illumination beam and as such is effectively invisible to the detector.

The avalanche photodiode detector was changed to a newer version of the same product, namely the SPCM-AQR-13 by Perkin-Elmer Optoelectronics. This detector's output pulses are longer and easier to detect, at the price of a slightly longer dead time between detected photons. Given the power levels involved in our optical observations, this presents no difficulty.

The optical detector was encased in a small metal box whose inside surfaces were covered with black felt. This helps reduce the background noise to the level of the detector's dark current. The set of filters in front of the detector was changed to a 600 nm bandpass filter (as before) and a 650 nm shortpass filter (also purchased from Chroma Technologies), as this resulted in a slightly better transmission of the fluorescence wavelengths. The blocking power of the shortpass filter from 650 to 700 nm is of approximately 6 optical densities. With the optical filters in place, the

quasi-totality of the pulses output by the detector corresponds to photons that we intend to detect, meaning that they come from the sample plane and are in the range of wavelengths of interest. Consequently, the acquired data can be assumed to be composed exclusively of photons we *want* to detect; the noise in the system is completely dominated by shot noise, because of the random arrival times of the detected photons.

3.3 Data Acquisition

Digital Instruments's BioScope system includes its own computer to control the microscope and to acquire and display data. It is a closed system, with no possibility of building or using "plug-in" modules. This means that, unlike the entirely home-built system, the second-generation FANSOM system cannot be run using a single console.

This having been said, there was no need to change the operation for the case of optical-only, far-field scanning, since optical scans do not need any information from the AFM.

3.3.1 Nature of the Data

In the case of FANSOM operation, the optical system must synchronize itself with the AFM's EOL and EOF pulses. In the first-generation FANSOM, the system worked from a single control panel, and all parts of the system used the same source of timing. Divorcing the AFM from the optical microscope controller introduces a difficulty with timing: we must be able to acquire information that is correctly and reliably synchronized with the AFM—otherwise we cannot have a precise instrument.

This issue was side-stepped by a fundamental change in the way the data is acquired. Previously, photons had been counted in pixel-sized bins. This was a compact method of saving images, with a predictable file size. However, the possibilities for data analysis on data acquired this way were limited. Since 1998, the price-per-megabyte of computer hard drives has dropped by nearly two orders of magnitude. A welcome side effect of that price drop is that acquiring and storing 1 GB of data in a single day is now affordable enough to be done routinely.

In the new system, a timestamp is saved for each photon detected by the avalanche photodiode. In addition to the photon timestamps, we also save a timestamp for each tap of the AFM probe, using the output of the fast comparator circuit described in Section 3.1.1.

3.3.2 Synchronization

The synchronization of the data acquisition with the AFM scanning is done by using the EOL and EOF pulses from the NanoScope[®] controller to alternately start and stop a 20 MHz clock. Interrup-

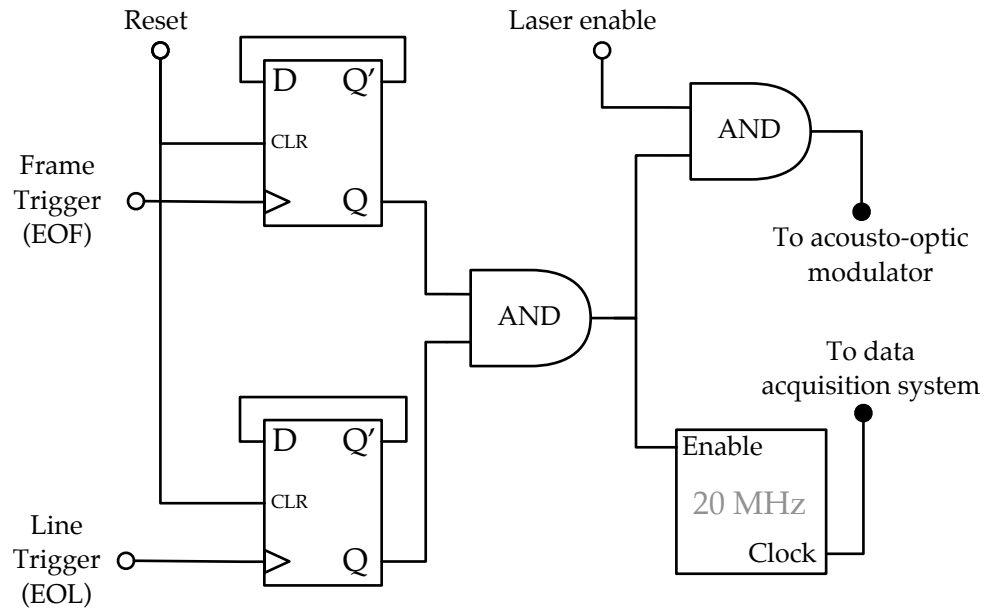


Figure 3.7: Diagram of the clock synchronization circuit. Open circles represent inputs, while filled circles represent outputs; all the signals are TTL logic. The line trigger (EOL) and frame trigger (EOF) cause the outputs (Q) of their D flip-flops to alternate, so that the laser and the clock are active only while data is acquired. The state of the outputs is reset by a software command before data acquisition begins.

tions of the clock's activity are detected and interpreted by the software as the end of a data line, upon which the line data is written to disk and preparations are made for the next line. Figure 3.7 shows a diagram of the circuit that interrupts the clock. The FANSOM software only needs to know the number of lines it is expected to acquire.

The timestamps saved for both the probe taps and the detected photons are the number of times the 20 MHz clock has ticked since data acquisition started for the current line. Two streams of timestamps are stored for each line, one for the taps and the other for the detected photons. Each data file is a monotonically increasing sequence of numbers.

3.3.3 Interpreting the Data

In essence, this version of the FANSOM is composed of two loosely synchronized instruments. The clock circuit's reaction to the AFM's synchronization pulses is very rapid and reliable; at less than 200 ns, it can be considered instantaneous.

The 20 MHz clock chip used in the synchronization circuit has an uncertainty of 10 ppm. The closed-loop version of the AFM head (Dimension[®] CL) has a nonlinearity of less than 1%, with a

positioning noise of less than 2 nm RMS. Given that the AFM probe is moved at a constant velocity, time and position can be very well correlated by a linear mapping of time to position. It is of note that the NanoScope[®] software itself assumes that the probe is moving at a constant velocity [36]. It therefore acquires its topographical data points at a constant data rate, as a result of which correlations between topographic and optical data are easily accomplished, even for images acquired using the open-loop scanning mode of the AFM scanner.

The sequence of timestamps is a somewhat compact way of storing what is in essence a 1-bit sampling of the photon detector's state: 1 if a photon is present, 0 otherwise, with a sample every 50 ns. Unless the total fluorescence is such that the count rate is in excess of 625 000 photons per second, storing the positive events as timestamps is more compact than storing 1-bit samples. If a time representation is desired, the transformation from the timestamp representation to the sample representation is easy in MATLAB[®]. An array of timestamps `d` can be transformed to a 1-bit sample representation with the following two steps:

```
td = zeros(d(end)-d(1)+1, 1);
td(d - (d(1)-1)) = 1;
```

The next chapter describes the methods used to extract data from the sequences of timestamps saved by the FANSOM driver.

Chapter 4

Mathematical Modeling and Data Processing

4.1 A Model for the FANSOM Data

The data measured by the FANSOM is a stream of photons emitted by a fluorescent source. We record each detected photon individually by its arrival time; the delay between two recorded photons is a random quantity that can be represented by the exponential random variable

$$f(\tau) = u(\tau) \cdot \lambda e^{-\lambda\tau} , \quad (4.1)$$

where λ represents the average rate of photon arrivals and

$$u(\tau) = \begin{cases} 0 & \tau < 0 \\ 1 & \tau \geq 0 . \end{cases}$$

Equation (4.1) represents a constant-rate case, with no modulation. The probability that the next photon will arrive at a time $X > \tau$ is given by

$$\begin{aligned} P[X > \tau] &= \int_{\tau}^{\infty} \lambda e^{-\lambda t} dt \\ &= e^{-\lambda\tau} . \end{aligned} \quad (4.2)$$

The exponential distribution of τ implies that it is a memoryless process, i.e., the distribution is the same regardless of the moment at which we start observing the system. The assumption that the process is memoryless is not strictly true for single fluorophores because of their photon antibunching behaviour; we do not stimulate them at a high enough rate for this to become an issue, however.

The exponential distribution can be thought of as a limiting form of the geometric random variable [37]. The geometric random variable models the number M of Bernoulli trials before the next successful trial. A Bernoulli trial has two outcomes: “failure” (0) and “success” (1). The probability of the “success” outcome is p .

The geometric random variable is defined as follows:

$$\begin{aligned} P[M = k] &= (1 - p)^{k-1} p \quad k = 0, 1, 2, \dots \\ P[M > k] &= (1 - p)^k \end{aligned}$$

We can use the average rate from Equation (4.1) to derive the exponential distribution from the geometric variable. We first divide a time interval of duration T into n time-slices, and calculate the probability of success per time-slice as $p = \lambda T/n$. The number of time-slices until the occurrence of a “success” event is a geometric random variable M , representing a time interval $X = M \cdot T/n$:

$$\begin{aligned} P[X > \tau] &= P\left[M > n \frac{\tau}{T}\right] = (1 - p)^{n\tau/T} \\ &= \left(\left(1 - \frac{\lambda T}{n}\right)^n\right)^{\tau/T} \end{aligned} \tag{4.3}$$

Equation (4.3) reduces to Equation (4.2) when n goes to infinity:

$$\lim_{n \rightarrow \infty} P[X > \tau] = e^{-\lambda \tau}$$

In practice, the granularity of the FANSOM data collection in time is finite, yet it is high enough that time delays are well modeled by a geometric random variable. Consider that we rarely see detection rates above 100000 s^{-1} : a time-slice of 50 ns is much shorter than the average wait encountered at that high rate. This approximation suggests that each individual time-slice in which observations are made can be modeled by a single Bernoulli trial in which $p = \lambda T/n$, where λ is the rate of events applicable at the current time interval.

In the FANSOM, the rate λ is periodically modulated by the probe; to represent this, we redefine λ as a function of time

$$\lambda(t) = \lambda_0 + \sum_{k=1}^K A_k \sin(k\omega_0 t + \theta_k), \tag{4.4}$$

where A_k are the coefficients of the probe modulation at its fundamental frequency f_0 and its harmonics, and θ_k are phase delays for each frequency. For convenience, since we are mostly concerned with the amplitude terms representing contrast, we assume that the θ_k are small or zero. Note that, by the same token, the probability of success p of the Bernoulli trials also becomes a function of time.

The problem we are faced with is estimating the modulation coefficients A_k in the expression $\lambda(t)$ from Equation (4.4). Once estimations are made for a set of pixels forming an image, a second problem is the determination of whether the values of a set of pixels represent an object. In the section that follows, we used a simplified version of the FANSOM data acquisition to calculate signal-to-noise ratios for realistic signal levels.

4.2 Time-Binned Data Acquisition Model

In this section, we establish the level of noise and uncertainty present in FANSOM measurements composed of paired bins, the method described in 2.1.3.2). A first bin's value is the number of photons counted while the AFM probe is interacting with the sample, and the second's value is the number of photons counted while the probe is outside of its interaction range with the sample. The signal we study in this section is the difference of those two bin values.

4.2.1 Modeling FANSOM Data with the Poisson Distribution

For one of these bins we count the number N of photon arrivals during a fixed time interval T . This situation is modeled by the Poisson random variable, expressed as

$$P[N = k] = \frac{\alpha^k}{k!} e^{-\alpha} \quad k = 0, 1, 2, \dots, \quad (4.5)$$

where α is a dimensionless rate. We can relate α to the rate λ from Equation (4.1) as follows: $\alpha = \lambda T$. The mean and variance of the Poisson variable are

$$\begin{aligned} E[P[k]] &= \alpha \quad \text{and} \\ \text{VAR}[P[k]] &= \alpha. \end{aligned} \quad (4.6)$$

If we consider the bin as a collection of discrete time-slices each modeled by a single Bernoulli trial, the Poisson variable is still a good approximation. Consider the probability of all permutations of k successes out of n Bernoulli trials:

$$\begin{aligned} P[N = k] &= \binom{n}{k} (1-p)^{n-k} p^k \\ \lim_{\substack{n \rightarrow \infty \\ \alpha = np}} P[N = k] &= \frac{\alpha^k}{k!} e^{-\alpha} \end{aligned}$$

Note that $\alpha = np$ remains a fixed value as $n \rightarrow \infty$. As was the case when taking the limit of Equation (4.3), the granularity of our time-slices is sufficiently high, since a typical bin covers a

duration on the order of one millisecond whereas the observation time-slice is, at 50 ns, less than 10^{-4} times as long.

During the data acquisition, we collect bin data by counting photons detections during a series of discontinuous periods of time. We can consider the sum of these periods of time as one continuous block, since by nature the underlying Bernoulli trials are independent from each other.

By the same token, we can also say that for a given $\alpha = K$, we are observing the sum of K different Poisson variables each having $\alpha = 1$. In fact, for $\alpha > 10$, we can use the central limit theorem and obtain an approximation of $P[k]$ from a normal distribution of mean and variance α , so that the parameters of the normal distribution are $\mu = \alpha$ and $\sigma = \sqrt{\alpha}$. The normal distribution is determined by its mean (μ) and standard deviation (σ), and is defined as

$$n(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}.$$

We go from the Poisson to the normal distribution with

$$\begin{aligned} P[N = k] &= \frac{\alpha^k}{k!} e^{-\alpha} \\ &\approx \frac{1}{\sqrt{2\pi\alpha}} \int_{k-1/2}^{k+1/2} e^{-(x-\alpha)^2/2\alpha} dx \\ &\approx \frac{e^{-(k-\alpha)^2/2\alpha}}{\sqrt{2\pi\alpha}} \\ &\approx n(k; \alpha, \sqrt{\alpha}), \end{aligned} \tag{4.7}$$

where we have approximated that the slope is constant over the unit-length integration range. We can then make a simple variable substitution in the cumulative distribution integral,

$$\begin{aligned} P[N > k] &\approx \frac{1}{\sqrt{2\pi\alpha}} \int_k^\infty e^{-(x-\alpha)^2/2\alpha} dx \\ &\approx \frac{1}{\sqrt{2\pi}} \int_{(k-\alpha)/\sqrt{\alpha}}^\infty e^{-t^2/2} dt \\ &\approx Q\left(\frac{k-\alpha}{\sqrt{\alpha}}\right), \end{aligned}$$

where we make use of the function $Q(x)$, whose results are widely tabulated:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt \tag{4.8}$$

In modeling this mode of FANSOM data acquisition, each of the pixels are composed of two bins for which the Poisson distributions have different rates. We use the rates α_u for the “probe up” bin and α_d for the “probe down” bin. α_d is related to α_u by a modulation factor q_0 that corresponds

to the amount of probe-induced quenching or enhancement present during the data collection:

$$\alpha_d = q_0 \alpha_u$$

The value of the modulation factor q_0 can represent three different situations.

$$q_0 \begin{cases} < 1 & \text{for the case of fluorescence quenching,} \\ = 1 & \text{in the absence of probe effect, and} \\ > 1 & \text{for the case of fluorescence enhancement.} \end{cases}$$

We assume that the probe's lateral movement is negligible while the data for each pixel is being acquired, i.e., α_u and α_d (and q_0) do not vary throughout a given bin. If we collect the data $k = k_u - k_d$, the probability distribution of k is given by the following distribution:

$$P[k] = \sum_{n=0}^{\infty} P_{k_u}[k_u = n] P_{k_d}[k_d = n - k] \quad (4.9)$$

The mean and variance of $P[k]$ are given by

$$E[P[k]] = E[P_{k_u}[k_u]] - E[P_{k_d}[k_d]] \quad \text{and} \quad (4.10)$$

$$\begin{aligned} \text{VAR}[P[k]] &= \text{VAR}[P_{k_u}[k_u]] + \text{VAR}[P_{k_d}[k_d]] - 2 \cdot \text{COV}[P_{k_u}[k_u] P_{k_d}[k_d]] \\ &= \text{VAR}[P_{k_u}[k_u]] + \text{VAR}[P_{k_d}[k_d]] , \end{aligned} \quad (4.11)$$

where the covariance is zero since we have independent random variables. Substituting from Equation (4.6) into the preceding two equations,

$$\begin{aligned} E[P[k]] &= \alpha_u - \alpha_d = \alpha_u(1 - q_0) \\ \text{VAR}[P[k]] &= \alpha_u + \alpha_d = \alpha_u(1 + q_0) \end{aligned}$$

In keeping with our use of the central limit theorem, sufficiently large values of α_u and α_d in Equation (4.9) lead to the following normal distribution:

$$P[k] \approx n\left(k; \alpha_u(1 - q_0), \sqrt{\alpha_u(1 + q_0)}\right) \quad (4.12)$$

4.2.2 Signal-to-Noise Ratio in Images

Equipped with an expression for the probability distribution of pixel values, we can move on to finding a reasonable expression for a signal-to-noise ratio. We compare the values of two sets of bins: the bins corresponding to the near-field signal of an object ($q_0 \neq 1$) and the bins corresponding

Table 4.1: Photon detection rates for various objects.

Object	Photon Detection Rate λ [s ⁻¹]
Single fluorophore	5000
20 nm bead	10000
>60 nm bead	20000
Quantum dot	50000

to the far-field signal of the same object ($q_0 = 1$, with $\alpha_u \gg 1$).

The two sets have normal distributions with parameters μ_1 , σ_1 , μ_2 and σ_2 . We can therefore consider the following quantity as a signal-to-noise ratio:

$$\frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}} \quad (4.13)$$

4.2.2.1 Typical Conditions of Illumination

Before we continue, we need to establish some parameters corresponding to typical conditions for the FANSOM signals we wish to model.

Detected Power Levels Table 4.1 lists photon detection rates as order-of-magnitude measurements¹, for various targets that have been used during the instrument development. For some of these, like the larger beads and quantum dots, it is easy to raise the count rates to higher values by simply increasing the illumination power, but photon count rates for single fluorophores and 20 nm beads stay close to those listed, limited mainly by the phenomenon of photobleaching.

The single fluorophores referred to in Table 4.1 are CyTM 3, obtained from Amersham Biosciences Inc. and TRITC (tetramethylrhodamine-5-(and-6)-isothiocyanate), obtained from Molecular Probes Inc. The beads are latex beads with embedded Nile Red fluorophores from Interfacial Dynamics Corporation; the beads are further described in Appendix B.2. Lastly, the quantum dots are ZnS-capped CdSe colloids with an emission wavelength of 580 nm, from the laboratory of Paul Alivisatos at the University of California at Berkeley.

Typical Observation Times The Poisson distributions use the non-dimensional rate $\alpha = \lambda T$, making the signal-to-noise figures obviously dependent on the observation time. FANSOM images are typically acquired at the rate of one line per second, where the acquisition time is half of this period since the probe travels at a constant speed and must “rescan” the width of the imaged area before the next line is acquired. The line period is then further divided into the number of pixels we wish

¹That is, if we consider octaves rather than decades.

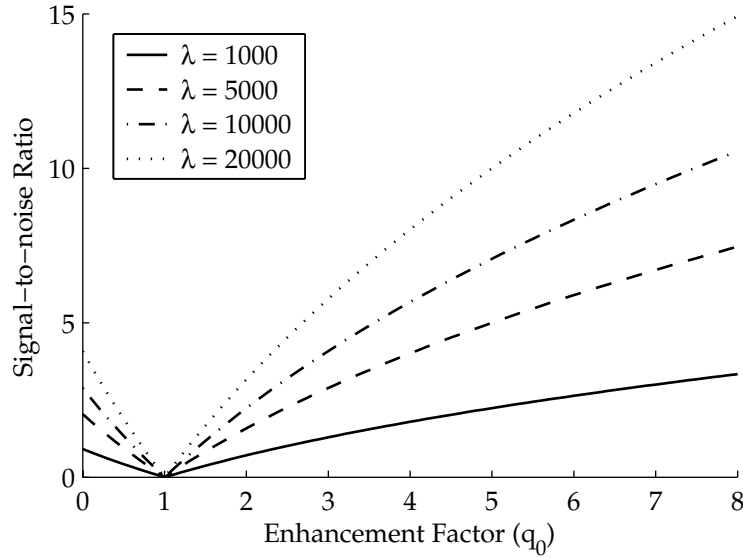


Figure 4.1: Variation of SNR as q_0 and λ vary, for a single illuminated object.

to use. The time interval for a typical pixel is approximately 0.01 second, although this can vary easily by an order of magnitude. The observations for each of the paired bins discussed in this section last for 25% of the total pixel interval rather 50%, since we aim to minimize the variation of the probe's effect throughout the observation.

4.2.2.2 Signal-to-Noise Ratio for a Single Object

For a single object, we obtain from Equation (4.13) an expression of the form

$$\frac{|\alpha_u(1 - q_0) - \alpha_u(1 - 1)|}{\sqrt{\alpha_u(1 + q_0) + \alpha_u(1 + 1)}} = \frac{|\alpha_u(1 - q_0)|}{\sqrt{\alpha_u(3 + q_0)}}.$$

If we were to plot this as a function of α_u and q_0 , we would obtain a surface whose value is proportional to the square root of α_u and to a function close to the absolute value of the square root of $(1 - q_0)$. Figure 4.1 shows a plot of computed signal-to-noise ratio as a function of q_0 for a few values of λ . The pixel interval used was 0.01 second, with the paired bins each covering 25% of the pixel interval.

These plots show, that for a sample consisting of a single fluorophore, modulation situations that involve a reduction of detected fluorescence ($q_0 < 1$) can only result in marginal signal-to-noise ratios. The enhancement factor would need to be quite high, possibly 10 or more, in order to get quality contrast on a single fluorophore.

4.2.2.3 Signal-to-Noise Ratio for Multiple Objects

The area of effect of the FANSOM probe is much smaller than a diffraction-limited laser spot, so it is possible to illuminate multiple targets while the probe modulates the fluorescence of only one of them. In such a situation, all the modulation is provided by the single modulated target while the rest contribute only noise.

The values of μ_1 and μ_2 in Equation (4.13) are unchanged by this situation, but the variances σ_1^2 and σ_2^2 are changed. Here we assume the worst case, namely that all n targets are excited with the same laser intensity. The unmodulated objects will thus produce photons at the rate of α_u , the same as the rate for the “far” bin of the modulated object:

$$\begin{aligned}\sigma_1^2 &= \alpha_u(1 + q_0) + 2\alpha_u(n - 1) \\ &= \alpha_u((2n - 1) + q_0) \\ \sigma_2^2 &= \alpha_u(2n)\end{aligned}$$

We substitute these back into Equation (4.13) to obtain

$$\frac{|\alpha_u(1 - q_0)|}{\sqrt{\alpha_u((4n - 1) + q_0)}}.$$

The modified expression now has an inverse proportionality to the square root of the number of illuminated objects n , in addition to the aforementioned dependencies on α_u and q_0 . Figures 4.2 and 4.3 show the effect of the presence of multiple objects in the illumination beam.

4.2.3 Photon Budget and Signal-to-Noise

Fluorescence is fleeting by nature. It exhibits a number of traits that make it difficult to observe. The worst of these is photobleaching, another, more benign, behaviour is blinking. Either of these, however, mean that any detection must occur as rapidly as possible—detection may become impossible after the source has gone dark.

Sources that contain multiple fluorophores, such as fluorescent latex beads, have a long lifetime and tend to go dark in an exponential decay. Single fluorescent molecules go dark suddenly and irrevocably as they photobleach. Quantum dots go dark just as suddenly and, although they do not photobleach, the moment of their reappearance is highly uncertain [38].

Experience shows that single fluorophores at the glass-air interface can typically be observed for 15 to 20 seconds before they photobleach, allowing us to collect between 40 000 and 60 000 photons from the molecule. This is consistent with the collection efficiency of our optical system (approximately 8%) and the conventional wisdom that fluorescent molecules emit up to an average

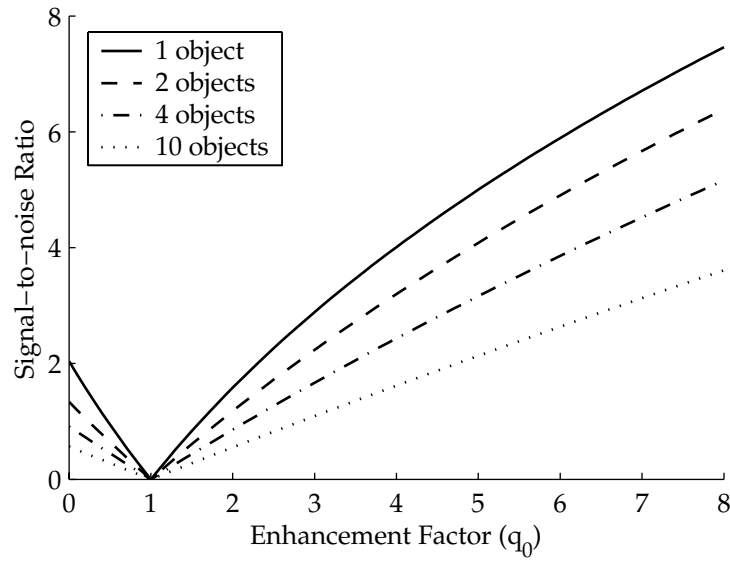


Figure 4.2: SNR curves for various numbers of illuminated objects. The plotted data was computed for a photon detection rate of $\lambda = 5000$.

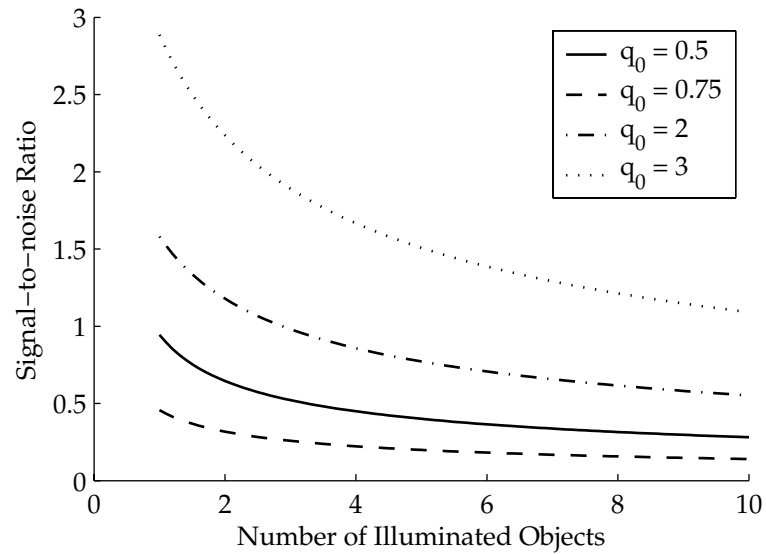


Figure 4.3: Variation of the SNR as the number of illuminated objects grows, for multiple values of q_0 . The plotted data was computed for a photon detection rate of $\lambda = 5000$.

of 10^6 photons before photobleaching.

If a single molecule were located under the AFM probe before we started illuminating it, we would be able to collect an ample amount of data on it by the time it photobleached. In practice, however, a single molecule is effectively invisible to the AFM and must be located optically.

In order to get FANSOM data from a molecule, it has to be scanned by the AFM probe. However, it is illuminated by the much larger laser spot for a long time before the probe even reaches it. We assume that the sample is scanned relative to the probe and laser (either by a sample scanner or synchronized laser/probe scanning). In the resulting measurement, we can reasonably determine that a fluorophore exists at a given location by using the results of a few contiguous pixels, repeated on consecutive scan lines. We set the figure at four pixels repeated on four consecutive lines.

Since the AFM probe is at the center of the laser beam, approximately half of the laser beam area is scanned over the sample before detection becomes possible. We use the pixel rate cited above under “typical observation times”, i.e., our pixel scan rate is 100 s^{-1} .

Finally, we need to assume a certain “size” for our sample, that is, determine an area that corresponds to the 4×4 pixels mentioned above. We assume that our lateral interaction length is approximately 10 nm, a number that is reasonable given current results.

Photon budget in direct illumination Direct illumination with a Gaussian beam results in a localized reduction of the detected fluorescence. With our setup, the spot at the sample plane has a full-width half-maximum (FWHM) diameter of approximately 400 nm. We are interested in evaluating the amount of energy that hits our sample more than in the shape of the beam, so we will use the “top hat” approximation of a Gaussian beam (i.e., flat intensity equal to the Gaussian peak, with a circular beam; see Siegman [39], Equation (17-7)). We can relate the “top hat” area to the FWHM diameter in the following way:

$$A_{TH} = \frac{\pi D_{FWHM}^2}{4 \ln 2} \approx 0.18 \mu\text{m}^2$$

If we scan at a speed of 2.5 nm per 0.01 second, or $0.25 \mu\text{m/s}$, scanning an entire square micrometer requires 400 lines at 4 seconds per line. Thus, when scanning our sample with the “top hat” laser beam, the sample is illuminated for

$$\Delta t_{\text{scan}} = 0.18 \cdot 1600 \text{ s} = 288 \text{ s} .$$

Half of the laser spot area is scanned over the sample before the probe gets the chance to interact with it; the sample will have been illuminated for approximately 150 s by the time our target has

been scanned for all the pixels required for detection.

In these conditions, a single fluorophore will definitely be photobleached by the time we have scanned it with the FANSOM. Constant illumination is clearly not an option, but we can calculate in reverse and determine how much illumination we can afford.

As previously stated, we can illuminate the molecule at peak intensity for approximately 15 seconds before photobleaching occurs, enough to acquire 1500 pixels—a 100 nm square, under the scanning conditions calculated above.

We can conclude from this that a single-molecule detection is difficult but possible using these conditions. The molecule could be located approximately using a coarse (and fast) optical scan, then a 80-100 nm square scanned near the center of the located peak. This would result in a sufficient amount of data for a single molecule detection.

Fortunately, it is possible to extend the life of a single fluorescent molecule by immersing it in a nitrogen atmosphere or a liquid, or by embedding it in a thin polymer film. The first and the third of these possibilities are obviously compatible with FANSOM imaging. It may be possible to do FANSOM imaging in liquids, but it will require further research and probe development.

4.2.4 Limitations of the Opposed Bins Method

The estimation of modulation depth through binning seems to not be as powerful as it should be for our purposes. First of all, it doesn't make use of all available data, since it must throw away a part of the oscillation period in order to get bins with average powers as different as possible. Secondly, the phase offsets of the opposed bins must be preprogrammed, since there is no way to automatically find out the optimum phase offset. If the type of probe used is always the same, then there is little need to change the phase offset, but different probe and cantilever types have different responses to the driving force, and the phase offset can change.

Also, the opposed bins method cannot find modulation parameters for harmonics of the excitation frequency. Given that the FANSOM modulation is strongly nonlinear with respect to the distance between sample and probe, the harmonics of the probe frequency could contain useful information.

4.3 Fourier Analysis

Unlike the method described in the previous section, Fourier analysis can make use of the entirety of the photon timestamp data, and could potentially provide information about higher harmonics.

The Fourier transform is defined as

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt \\ &= \int_{-\infty}^{\infty} f(t) [\cos(\omega t) + i \sin(\omega t)] . \end{aligned} \quad (4.14)$$

The FANSOM data can be converted from timestamps to a time series with a sampling rate equal to the data acquisition clock, where the sampled values are binary: either “no photon detected” (0) or “photon detected” (1). The Fast Fourier Transform (FFT) algorithm can extract a sampled frequency spectrum from the resulting time series. By separating timestamps in subintervals of the observation time and applying an FFT to each subintervals, “time-resolved” spectral information can be obtained. A comparison of the spectral line of interest, i.e., at the probe’s resonance frequency to the level of background noise provides a quantity usable as pixel data.

If a random signal is observed for a very long time, the power spectrum of the observed signal is the Fourier transform of the signal’s autocorrelation [40]. For a sufficiently long observation, the Fourier coefficients are (nearly) independent random variables following normal distributions, such that

$$E[F_k] = \text{FFT}[E[f_i]] .$$

That is, the average FFT output is the FFT of the average input sample sequence. Unfortunately, the variances of these normal distributions are not bounded, even for an infinite period of observation; smoothing of the spectrum is necessary. In the case of our very sparse data, we cannot reasonably do this by calculating spectra for multiple subintervals of a pixel (as suggested by Leon-Garcia [37], Section 7.1) because, in most situations, there isn’t enough data for that. This means that performing spectral smoothing will require either two FFTs and one inverse FFT or two FFTs and a convolution (see [41], Section 12-2).

Consider that the FFT algorithm has an execution time proportional to $N \log N$, where N is the number of input samples. Our data has a number of samples per line equal to the observation time divided by 50 ns. Building images out of unsmoothed FFTs for a 64-line dataset takes approximately 10 minutes of computing with a 2 GHz Pentium® 4 processor; using smoothed spectra takes at least twice as long. The code used by MATLAB® to perform the calculations is known to be highly optimized, so a substantial speedup cannot be expected. Even by using code optimized for single-precision floating point numbers (rather than MATLAB®’s double-precision code), at most a speedup of two or three could be hoped for.

The result of FFT analysis would be desirable, but its computation is simply too time consuming for everyday use; the required delay would be acceptable if the instrument were its own end, but not if we want to use it as a tool. A faster method is preferable.

4.4 Correlation Techniques

A result similar to the output of a Fourier transform can be obtained by using the correlation function,

$$R(\Delta) = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau f(t) g(t + \Delta) dt . \quad (4.15)$$

We do not have an infinite amount of data to analyze, let alone infinite time. Thus, we must stop the integration at some finite value of τ . In the case that interests us, we also happen to be looking for the harmonic response to a harmonic excitation of the system:

$$f(t) = a \sin(\omega t)$$

$$g(t) = b \sin(\omega t + \Delta)$$

Substituting these into Equation (4.15), the correlation becomes a function of τ ,

$$R_1(\tau, \Delta) = \frac{ab}{\tau} \int_0^\tau \sin(\omega t) \sin(\omega t + \Delta) dt . \quad (4.16)$$

Interestingly, Equation (4.16) is very similar to the imaginary part of Equation (4.14) if $f(t)$ is taken to be $\sin(\omega t + \Delta)$. In fact, the result differs from the imaginary part of the Fourier transform of a switched sine wave by only a multiplicative factor.

The correlation function has a filter behaviour. When performing a correlation on data for a 0.01 second pixel with an oscillation frequency of 75 kHz, we obtain a bandwidth of 45 Hz. This bandwidth narrows with increasing τ .

The correlation is also phase sensitive. In order to maximize the result, the reference signal should have the same phase as that of the data being analyzed. Unfortunately, in general we do not know the correct phase in advance. However, there is a way to calculate it from the data itself.

We pointed out earlier that Equation (4.16) is a mere multiplicative factor away from the imaginary component of the Fourier transform at ω . We can also replace $f(t)$ in (4.15) by a cosine, obtaining

$$R_2(\tau, \Delta) = \frac{ab}{\tau} \int_0^\tau \cos(\omega t) \sin(\omega t + \Delta) dt . \quad (4.17)$$

This is similar to the real component of the Fourier transform at ω .

If we solve (4.16) and (4.17) for τ equal to an integer number of periods, we get

$$\begin{aligned} R_1(\Delta) &= \pm \frac{ab}{2} \cos(\Delta) \quad \text{and} \\ R_2(\Delta) &= \pm \frac{ab}{2} \sin(\Delta) . \end{aligned} \quad (4.18)$$

Our two values can be added together as polar coordinates, and we obtain

$$\begin{aligned}
 R_{out} &= \sqrt{R_1^2 + R_2^2} \\
 &= \sqrt{\left(\frac{ab}{2}\right)^2 [\cos^2(\Delta) + \sin^2(\Delta)]} \quad \text{where} \\
 \Delta &= \tan^{-1}\left(\frac{R_2}{R_1}\right).
 \end{aligned} \tag{4.19}$$

This shows that we can deduce the modulation phase from the modulated data.

A correlation-based method of data analysis is both more flexible and more realistic than the opposed bins method used earlier. We can even use a harmonic frequency of the cantilever resonance as the reference signal to study the contribution of such harmonics to the signal.

4.4.1 Implementation of Correlation Analysis

The computation of images by correlation analysis is well suited to the data we acquire with the FANSOM. The first step is to compute the time elapsed between each photon and the previous tap of the AFM probe. This is accomplished quickly (if not efficiently) by a double loop implemented in C that is called from MATLAB[®], called `convert2phases`. The code for the MATLAB[®] and the C versions of this function are shown in Listings D.8 and D.9, respectively. After this step is done, we have two vectors containing integer values corresponding to the detected photons; the original has the timestamps, while the other has time offsets. The values at a given index in the two vectors correspond to the same detected photon.

After obtaining the time offsets, the line of data is then separated into a number of bins corresponding to the number of pixels we want to obtain. The photon timestamps are used to determine the boundary points, and the range of vector indices corresponding to each bin is found.

Note here that in MATLAB[®], any integer can be used as an index to dereference a vector. We prepare two vectors, a sine and a cosine of identical lengths, corresponding to one period of the AFM probe oscillation. The vector of phase offsets only contains values that are smaller than the length of our sine and cosine vectors.

We can use the range of indices found earlier, along with the vector of phase offsets, to select values out of the sine and cosine vectors. This trick is mathematically equivalent to having expanded the sine and cosine to the full time interval of the pixel, and multiplied the expanded sine or cosine vector with a corresponding vector of 1-bit samples of the photon detector output, i.e., multiplying by one if there was a photon detected, and multiplying by zero otherwise. Adding the selected values together effectively performs the integration from Equation (4.15).

In the code execution, the actions of the previous three paragraphs are interleaved in order to economize memory; execution time is often adversely affected by large memory allocations, and

this implementation avoids that. The scripts that are run to perform this analysis are included in Section D.2.2.

The performance of this implementation of the correlation analysis is excellent. The most costly step (determining the time offsets) is performed by optimized code, and the rest of the steps are implemented in a way that makes full use of MATLAB®'s strength, vector operations. Timing experiments have shown that this analysis method routinely executes 100 times faster than a FFT analysis of the same data, while giving an output that is slightly less noisy.

4.5 Probe/Sample Approach Curves

In Chapter 2, we showed some approach curves that illustrated the variation of the detected fluorescence as a function of the probe-sample separation. With the old apparatus, substantial changes were required to acquire approach curves, but with the timestamp data of the newer apparatus, obtaining approach curves is simply a matter of re-analyzing the data with the proper script.

We start by using the `convert2phases` script to convert photon timestamps to time offsets. The resulting vector contains, for each photon, its time offset to the most recent tapping timestamp. That time offset can be converted to a phase by comparing it to the tapping period obtained from the tap timestamp data.

We can easily obtain a histogram of the distribution of the detected photons with respect to the probe phase. We obtain a vector of time offsets by using `convert2phases`, and then issue the `hist` MATLAB® command as part of a construct such as:

```
h = hist(d , (0:max(d)));
```

This produces a histogram that has enough unit-width bins to hold all the values present in the phase offsets vector `d`. Once enough phase offsets have been binned in this way, we have a meaningful reading of the influence of the probe throughout its oscillation period. The script `timestamphist` shown in Listing D.14 implements histogram generation from sets of timestamps.

4.5.1 Computation of Approach Curves

With a photons vs. phases histogram in hand, we are still a step away from an approach curve. We need to map the phase space to a probe-sample separation. The method we have used is to find the offset of either the minimum or maximum of the histogram, and use that offset as the point where the probe-sample separation is the smallest. In order to find this minimum or maximum, we calculate a few Fourier series coefficients to get a fit of the histogram, then use the calculated coefficients to obtain the minimum (or maximum) of the curve they represent. This works well

for data that is neither excessively noisy nor excessively flat—in other words, it works well for interesting data.

Once the minimum of the probe oscillation is identified, the whole phase space can be converted to a vertical position, as long as the oscillation amplitude is known. The oscillation amplitude can be obtained by performing a calibration in the NanoScope[®] software. Approach curve generation is implemented in the script `acfromhist` shown in Listing D.15. The phase-to-distance mapping is done assuming a perfectly sinusoidal trajectory of the AFM probe. This is a reasonable approximation, since the harmonic distortion of the probe position signal has been measured to be approximately 1% for both the first and the second harmonic of the driving frequency. This is not due to filtering by the electronics of the Dimension[™] head; the harmonic content is similar whether the probe is driven at 75 kHz or at 300 kHz.

In some data sets, the last one or two bins of the histogram are clearly outliers, having a much lower count than any of the other bins. Those bins represent the highest time offsets obtainable, just before a new tap gets recorded. The oscillators we use (the 20 MHz clock and the function generator of the NanoScope[®]) are good, but not perfect, and there is inevitably some jitter. Also, the frequency of the data acquisition clock is not an even multiple of the oscillator driving the AFM probe, so the last bin of the histogram is bound to be only partially filled, resulting in a lower value.

In the next chapter, we present data obtained with the FANSOM and analyzed with the methods we have just discussed.

Chapter 5

Experimental Results

The AFM probes used to generate the data described in this chapter were silicon FESP and TESP probes, and platinum/iridium-coated PIT probes. More details about these probes are given in Appendix A. The samples are 20 nm fluorescent latex beads, prepared as described in Appendix B.2.

5.1 Sample Illumination

Results in this chapter were acquired with the illumination modes described in Section 3.2.2. We first show far-field images of beads under the various illumination modes, to illustrate the energy distributions involved.

5.1.1 Beads under Illumination

Fluorescent samples are sensitive to local intensity. The intensity of the beam illuminating the sample can be mapped by imaging it with a fluorescent sample. These images were produced by scanning the laser beam over the stationary samples.

5.1.1.1 Direct Illumination

Figure 5.1 shows the image of a single 20 nm bead when illuminated by our tightly focused laser beam. This is the same kind of illumination used for the investigations with the original FANSOM apparatus (Chapter 2). The image is a convolution of the imaged bead with the laser mode (nearly Gaussian) that is used to illuminate it. The full-width half-max diameter of the signal shown in Figure 5.1 is 375 nm.

5.1.1.2 Annular Total-Internal-Reflection Illumination

When we illuminate a 20 nm bead with a polarized, annular, totally-internally-reflected beam, we obtain a pattern such as the one shown at Figure 5.2. Here, all of the radiation is incident on the

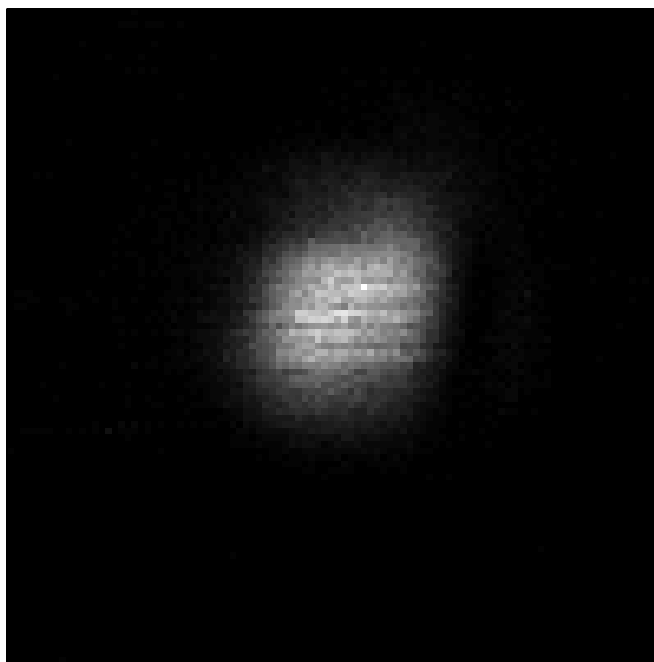


Figure 5.1: Image of a 20 nm fluorescent bead with direct illumination. The field of view is a $2\text{ }\mu\text{m}$ square.

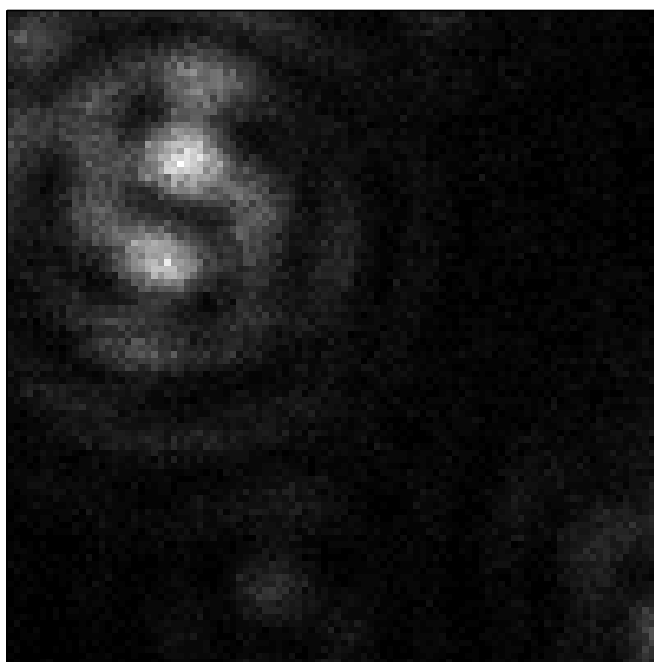


Figure 5.2: Image of a 20 nm fluorescent bead illuminated by a TIR ring. The field of view is a $2\text{ }\mu\text{m}$ square.

glass-air interface at an angle greater than the critical angle. The central portion of the illuminated area is not the brightest part of the pattern, as portions of the beam interfere destructively at that location. This image seems consistent with the measurements of Sick et al. [42, 43] about single fluorophores probed with annular illumination. The field distributions they calculate are 5 nm below the interface rather than in a range of 0-20 nm above the interface as is our case.

We can interpret the image of Figure 5.2 as a function of the polarization of the illuminating beam. The green photons entering from the sides of the image have a polarization component that is vertical, perpendicular to the sample plane; however, the vertical components are antiparallel, resulting in destructive interference near the center. Meanwhile, the photons coming from the top and bottom have parallel polarization components; they interfere constructively, producing the two bright lobes near the center.

5.1.1.3 Crescent Illumination (Directional-TIR)

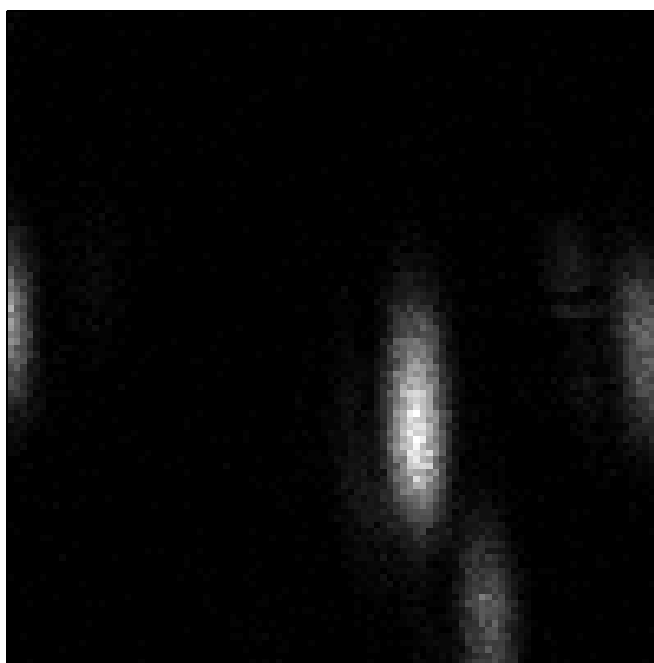


Figure 5.3: Image of multiple 20 nm fluorescent beads illuminated by TIR. The field of view is a $6.6 \mu\text{m}$ square.

An image of a bead illuminated with the single-polarization TIR illumination is shown at Figure 5.3. The laser beam no longer fills the back aperture of the microscope objective, and as a result the illumination at the sample plane is far from being a tightly focused spot. Fluorescence from a single bead is detected over a large area shaped like an ellipse 400 nm by 1300 nm; this area is much larger than if the bead were illuminated with a tight spot.

5.1.2 AFM Probes: Imaged Backscatter

There is a legitimate question of whether the AFM probes significantly affect the illumination field when they are above the sample. We attempted to determine this by observing backscattered energy from the probe, keeping in mind that it might be possible to locate the position of the AFM probe by analyzing the pattern of the backscattered photons.

We observed the backscatter by changing the bandpass filter in front of the photon counter for a bandpass filter that selects the green wavelengths from 530-550 nm. A neutral density filter of density 2 is also added, in order to avoid saturating the counter.

We measured the backscatter while scanning a tightly focused laser beam across the tip region of a stationary metal-coated AFM probe oscillating at the glass-air interface. Results for two different polarizations of the laser beam are shown at Figure 5.4.

The probe causes a variation of the backscattered power that is about 10% of the total. If we look at the modulation of the backscattered energy at the probe's oscillation frequency, we find that there is a modulation by the probe, but it has no discernible spatial component at any scale smaller than a few micrometers. This indicates that the modulation of the backscattered energy has little to do with the probe itself; it is probably due to reflections from the cantilever. This fact allows us to rule out the effect of reflected energy from the probe/cantilever system in producing a modulation of sample fluorescence at high spatial frequencies.

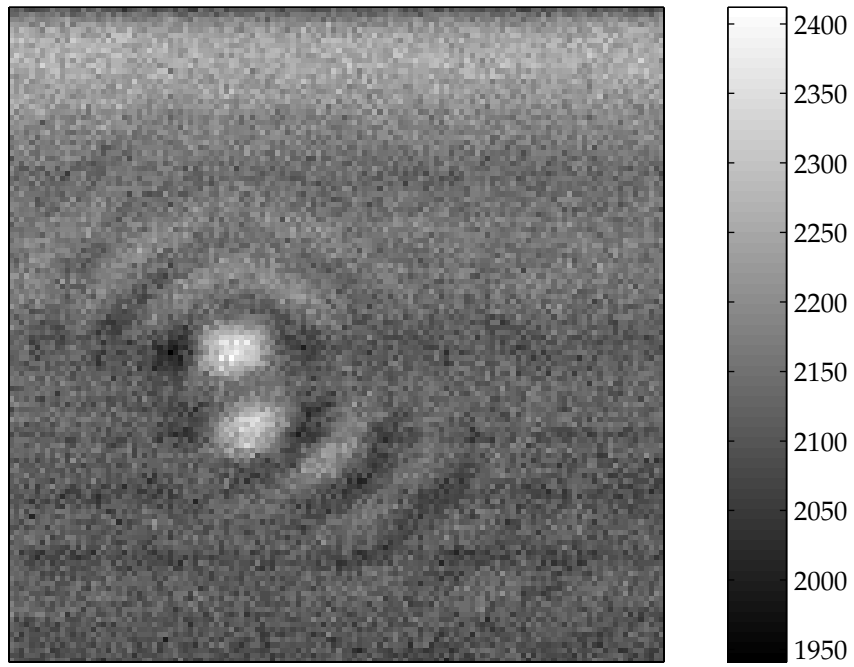
5.1.3 Fluorescence Pattern and AFM Probes

Since the presence of a probe affects (modestly) the shape of the backscattered energy, its presence might also affect the pattern of fluorescence from a bead. Figure 5.5 shows the fluorescence of a bead illuminated with an annular TIR beam, while a PIT probe oscillates above it.

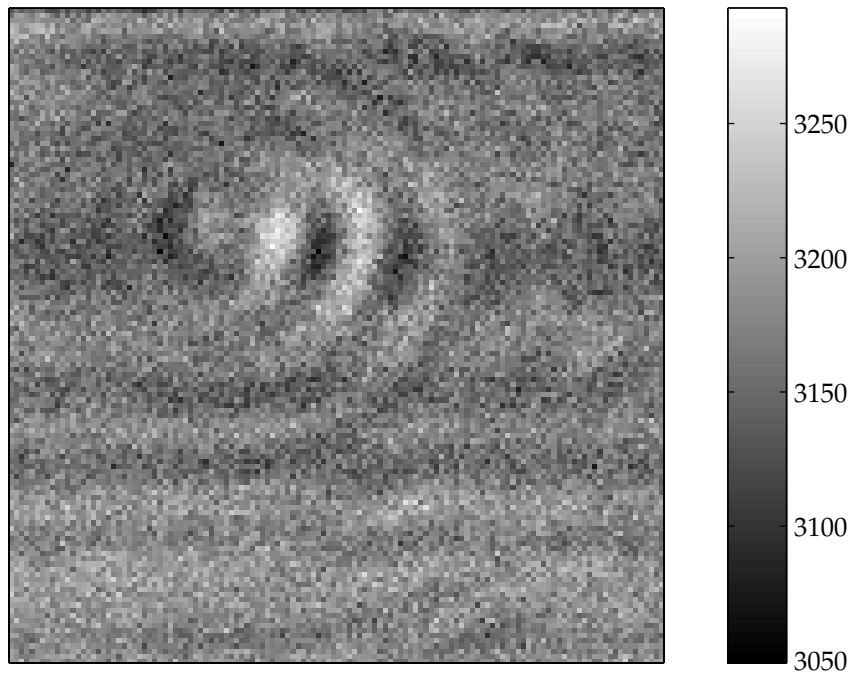
There is no significant change in the *shape* of the fluorescence from a bead when an AFM probe is near it, as we can see by comparing Figure 5.5 to Figure 5.2. The rotation of the pattern is accounted for by the fact that the polarization of the illumination was rotated by 90 degrees between the two images. Without making use of synchronous detection, the difference between the two images is not visible.

5.2 Fields, Probes and Fluorescence

The nature of the FANSOM contrast can be evaluated by correlating the position of the AFM probe with the arrival times of detected photons. The illumination being constant (i.e., uncorrelated with the probe), any correlation between the probe movement and the photon arrival times is the result of a perturbation caused by the probe.



(a) Backscattering with laser polarization parallel to the AFM cantilever.



(b) Backscattering with laser polarization perpendicular to the AFM cantilever.

Figure 5.4: Image of backscattered photons from the glass-air interface, perturbed by a PIT probe. The surface is illuminated with a tightly focused laser beam, and the fields of view are $2\mu\text{m}$ squares.

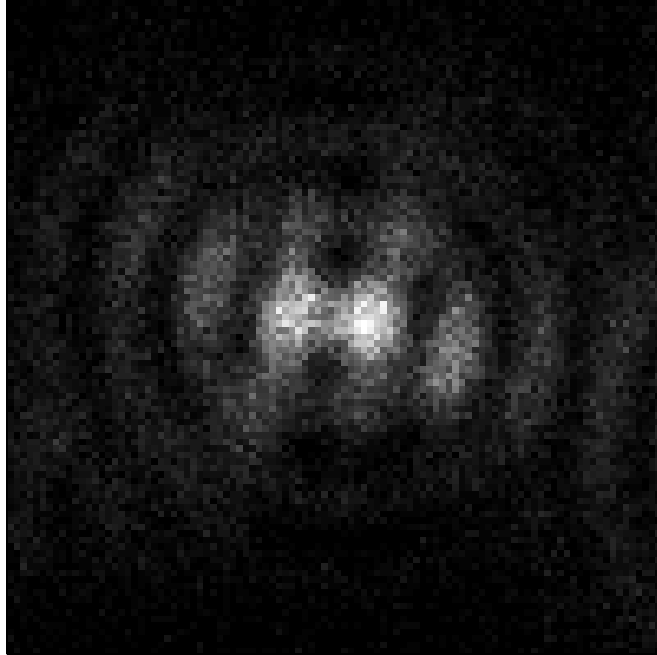


Figure 5.5: Image of a 20 nm fluorescent bead illuminated by an annular TIR beam, in the presence of a platinum/iridium-coated AFM probe.

The correlation is visualized by producing histograms where detected photons are binned as a function of the difference between their arrival timestamp and the most recent probe timestamp. In the absence of any correlation, the phase-offset histogram is flat (with the exception of noise). An example of a flat histogram is shown in Figure 5.6. When a significant correlation exists, the histogram bins can be mapped to a specific phase of the probe oscillation, and then to a probe-sample separation.

5.2.1 Fluorescence Inhibition

In all the modes of illumination we have used, metallic AFM probes introduce a modulation of the detected fluorescence intensity. An effect we always see with such probes is a reduction of the amount of detected fluorescence that correlates with the moments when the probe-sample separation is smallest.

Figures 5.7, 5.8 and 5.9 show phase-offset histograms and approach curves of 20 nm fluorescent beads modulated by PIT probes. These approach curve profiles are typical examples, and were repeated several times with different probes, producing similar results. Note that the approach curves of Figures 5.8 and 5.9 have nearly the same shape. For the measurement of Figure 5.8, we placed the bead on one of the two lobes where maximum intensity is located (see Figure 5.2). The polarization at that point is mostly in the plane of the glass-air interface, similar to the illumination

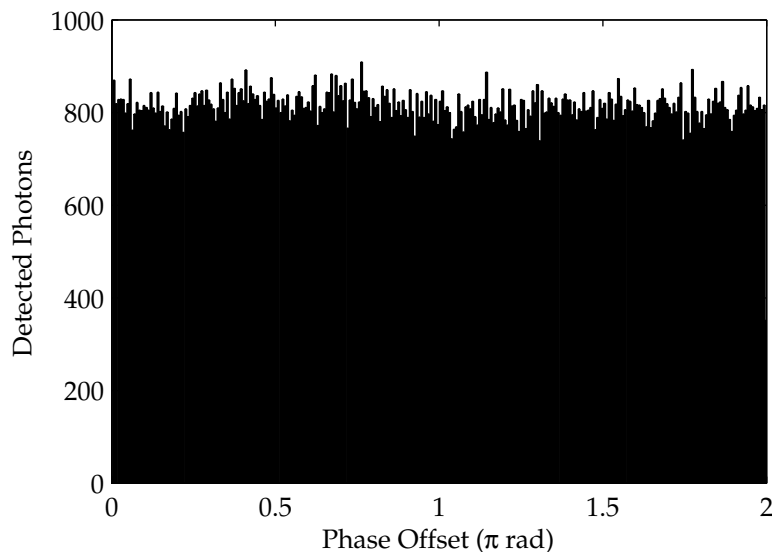


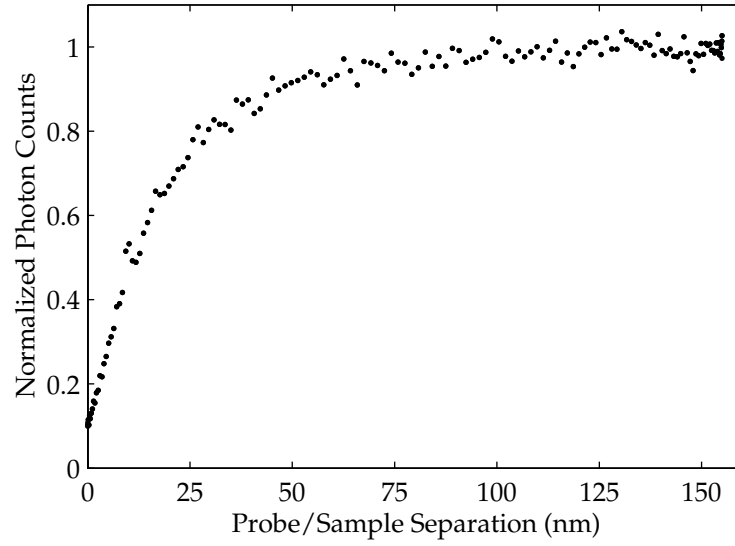
Figure 5.6: Phase offset histogram for a 20 nm bead illuminated by a tightly focused beam, modulated by a silicon probe. This probe had a resonant frequency of 70.8 kHz and an oscillation amplitude of 67 nm. The flatness of the histogram shows that the modulation caused by the probe is either nonexistent or too small to be detected. A flat histogram is also obtained in the absence of a probe, e.g., if it is raised far above the surface.

for Figure 5.9. This explains the similarity of the two figures.

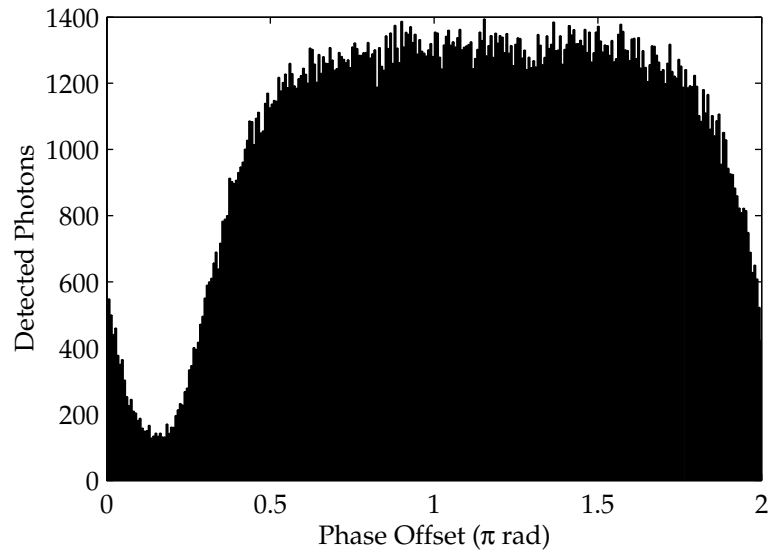
In all three cases, we observe a reduction of the detected fluorescence as the probe approaches the sample. The interaction range appears to be shorter for the direct illumination than for the in-plane TIR illumination, but nevertheless some fluorescence is somehow inhibited from reaching the detector. There are two possible mechanisms that could cause a reduction of detected fluorescence: fluorescence quenching and fluorescence redirection.

In fluorescence quenching [44], a non-radiative path for the molecular de-excitation is favored over the radiative path of fluorescence emission. Quenching denotes a class of processes that manifest themselves as a reduction of the excited-state lifetime of a fluorophore [22]. The quenching of interest here is a non-radiative energy transfer from an excited fluorophore to a nearby metallic surface, where the energy is channeled to the metal's surface plasmon modes [45]. The range that we observe for the fluorescence inhibition (~ 20 nm) is larger than what would be expected for this type of quenching.

The other mechanism is a redirection of the molecular emission pattern in a manner that would reduce the energy coupled into the microscope objective. While fluorescence redirection has been demonstrated at the single-molecule level by Gersen et al. [46, 47], it seems unlikely from those results that redirection could have such a dramatic effect. The results described therein shows that one half-space is favored three-to-one over the other; that is not enough to explain our reduction

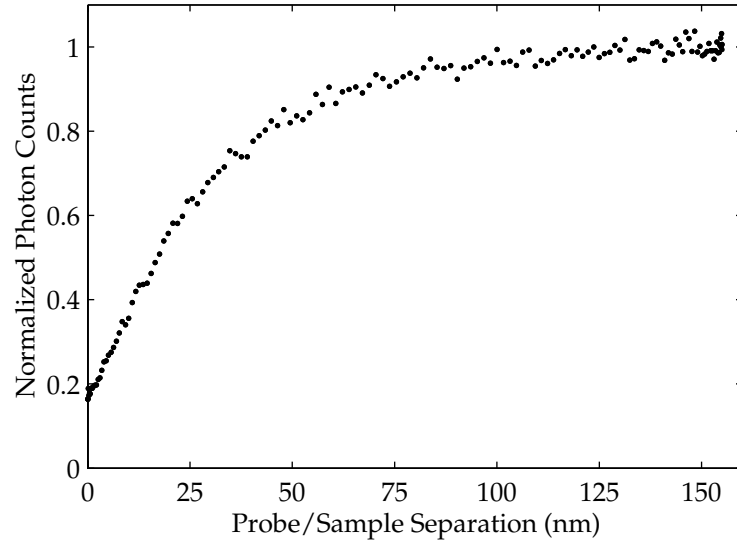


(a) Detected energy as a PIT probe approaches the sample.

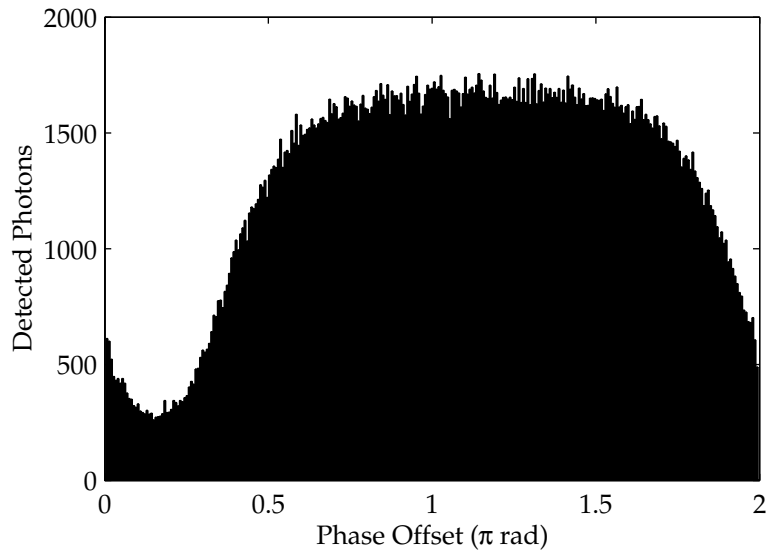


(b) Counted photons binned by phase offset.

Figure 5.7: Effect of a PIT probe on the fluorescence of a 20 nm bead excited by Gaussian laser illumination. This probe had an oscillation frequency of 68.3 kHz and an amplitude of 155 nm.

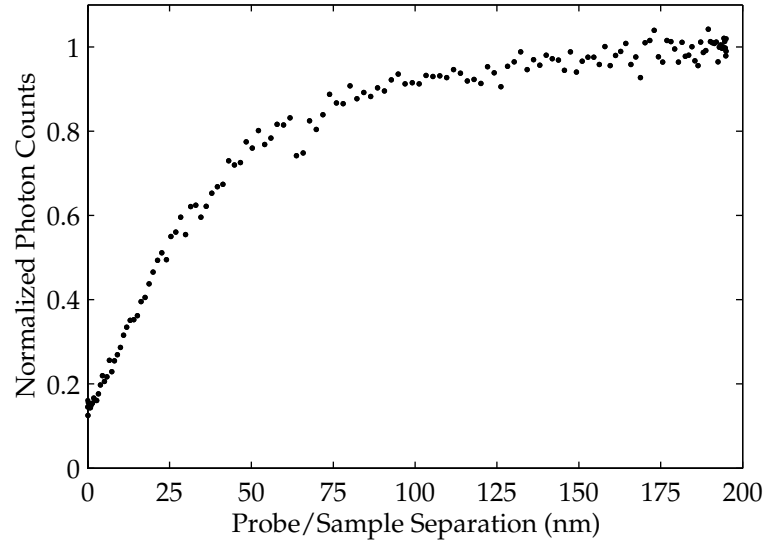


(a) Detected energy as a PIT probe approaches the sample.

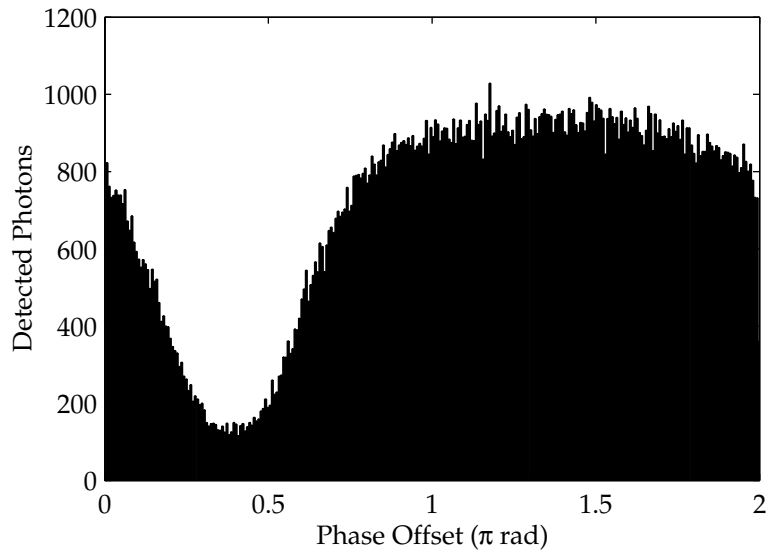


(b) Counted photons binned by phase offset.

Figure 5.8: Effect of a PIT probe on the fluorescence of a 20 nm bead excited by annular TIR illumination. This probe had an oscillation frequency of 68.3 kHz and an amplitude of 155 nm.



(a) Detected energy as a PIT probe approaches the sample.



(b) Counted photons binned by phase offset.

Figure 5.9: Effect of a PIT probe on the fluorescence of a 20 nm bead excited by a TIR illumination with its polarization parallel to the sample plane. This probe had an oscillation frequency of 69.3 kHz and an amplitude of 195 nm.

(nine to one). They also show that, in their system, fluorophores oriented along the optical axis redirect their fluorescence in the opposite direction from fluorophores perpendicular to the optical axis. Since we probe the behaviour of an ensemble of randomly-oriented fluorophores, this dependence on orientation should obscure the effect of redirection, making it contribute even less to the contrast.

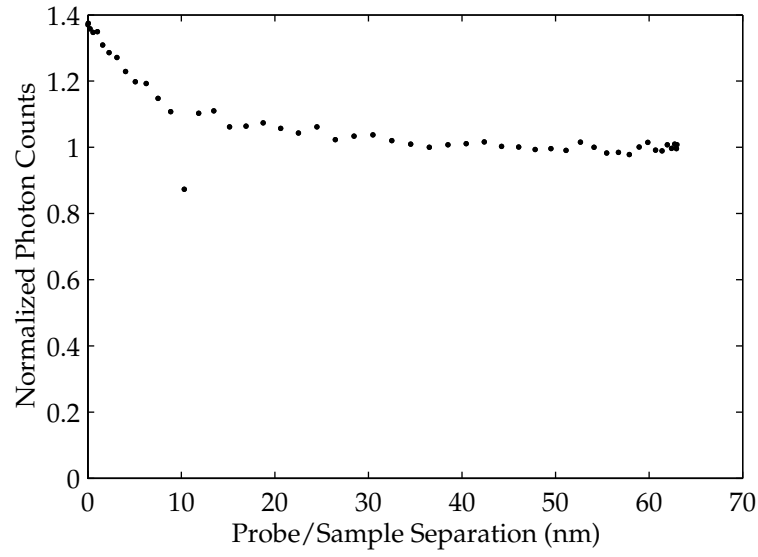
Some experiments can be proposed to differentiate between the two effects. Fluorescence quenching could be identified by taking fluorescence lifetime measurements using a pulsed laser and measuring the delay between a laser pulse and the fluorescent photons emitted by the sample as a result of that pulse. The redirection of fluorescence could be investigated by studying the changes in the angular distribution of the photons that travel to the single-photon detector. A method similar to the one used by Gersen et al. [47], i.e., the addition of a second single-photon detector, and a spatial separation of the photons directed to them. A significant difference in the shape of the approach curves obtained from each detector would suggest redirection. A simpler experiment might simply be to test objectives with a variety of numerical apertures. If the approach curves obtained from a 1.3 N.A. objective are systematically deeper than those obtained from a 1.4 N.A. objective, fluorescence redirection could be an explanation.

5.2.2 Field Enhancement

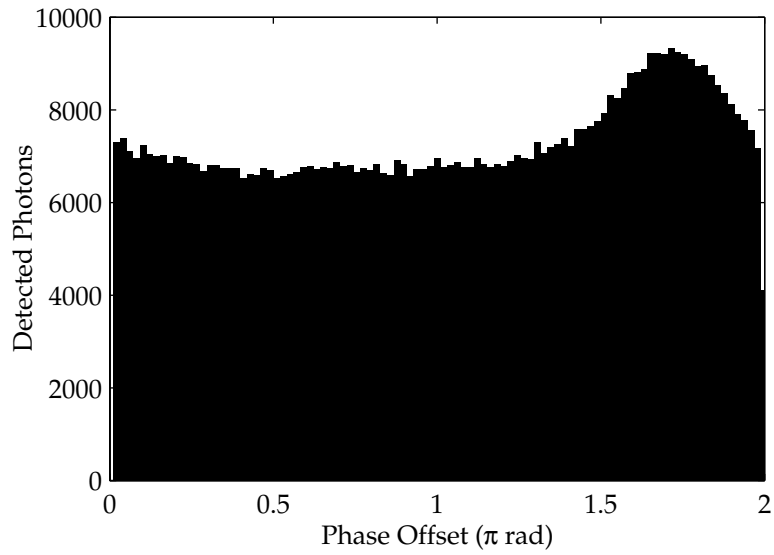
A frequently predicted result used to argue in favor of apertureless near-field microscopy is field enhancement at the tip of AFM probes [27, 48, 49]. Bohn, Nesbitt and Gallagher [48] calculate field enhancements of up to a factor of 20 by modeling a probe as an ellipsoid, while Martin, Hamann and Wickramasinghe [49] calculate factors of up to 6 by modeling probes with a more realistic probe geometry composed of a hemisphere-tipped cone. The fluorescence enhancement obtained should be proportional to the square of the field enhancement, since absorption is proportional to intensity—which is proportional to the square of the electric field.

A significant component of the incident field polarization must be parallel to the probe axis for field enhancement to occur. We can produce an illumination field with this characteristic by illuminating the sample with total internal reflection, using a beam that has a polarization that is radial compared to the back aperture of the objective. At the sample plane, such a beam will result in a polarization component parallel to the optical axis, and close to parallel with the probe axis.

We illuminated 20 nm beads in this manner, and measured approach curves using silicon probes and PIT probes. Figures 5.10 and 5.11 show enhancement in the presence of silicon probes. We used silicon probes with resonance frequencies of ~ 300 kHz (TESP) and ~ 75 kHz (FESP) to similar effect. We obtained lower enhancement factors using probes with the higher resonance frequency, probably as a result of tip sharpness. The TESP probes tend to tap the surface harder than the FESP probes do, and as a result get blunted faster. Since theoretical calculations show a strong

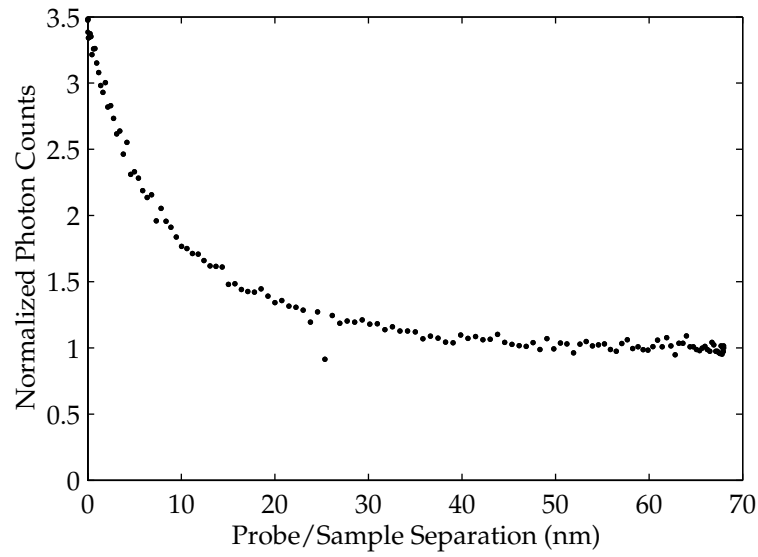


(a) Detected energy as a silicon probe approaches the sample.

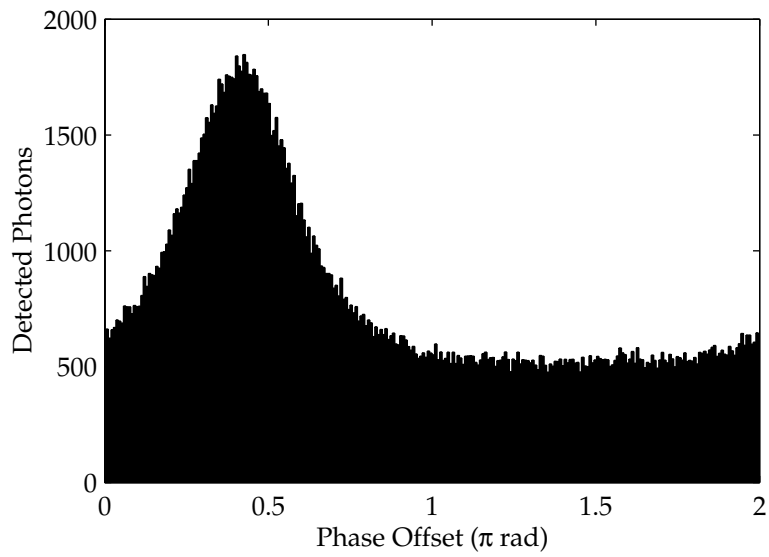


(b) Counted photons binned by phase offset.

Figure 5.10: Effect of a silicon probe on the fluorescence of a 20 nm bead excited by a TIR illumination with its polarization perpendicular to the sample plane—parallel to the probe axis. This probe had an oscillation frequency of 205 kHz and an amplitude of 63 nm.



(a) Detected energy as a silicon probe approaches the sample.



(b) Counted photons binned by phase offset.

Figure 5.11: Effect of a silicon probe on the fluorescence of a 20 nm bead excited by a TIR illumination with its polarization perpendicular to the sample plane—parallel to the probe axis. This probe had an oscillation frequency of 76.5 kHz and an amplitude of 68 nm.

dependence of enhancement on sharpness [48, 49] and the starting geometry of the probes is the same, the blunting of the probe is a plausible explanation.

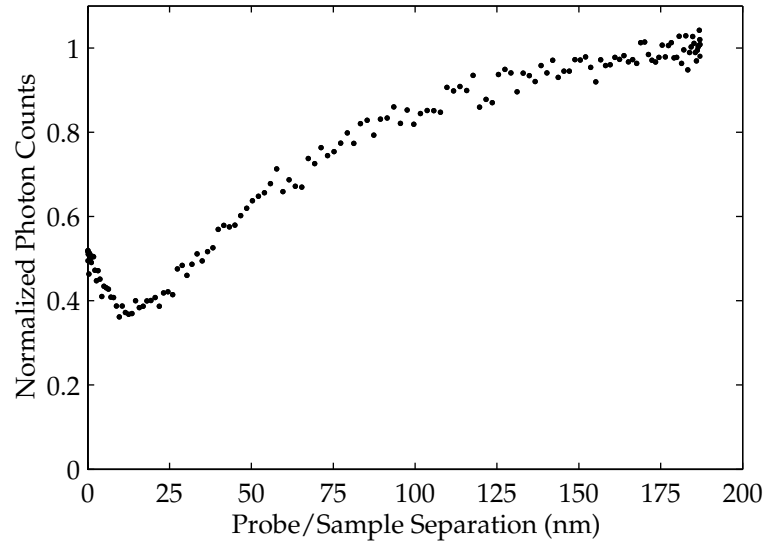
The maximum fluorescence enhancement we obtained from silicon probes on 20 nm beads was 4.5 (the result shown is a more typical 3.5). The range of field enhancement in the vertical direction should be very short according to Bohn et al. [48], with enhancement factors falling sharply between zero separation and 10 nm—the tip end radius of curvature of the probe. The range of the enhancement as calculated by Bohn et al. is qualitatively supported by our observation, although it is not possible to precisely extract the range of the enhancement from these measurements. The size of the beads is greater than the length scale of the enhancement, which makes an evaluation of the enhancement range and enhancement factor difficult at best.

Field enhancement should also be obtainable from metallic probes. We measured approach curves in the same illumination conditions as for the silicon probes using PIT probes, with the results shown in Figure 5.12. As can be seen from the approach curve, some enhancement occurs, but it is obscured by fluorescence inhibition as previously shown in Section 5.2.1. Here, the enhancement factor is even more difficult to evaluate than for the silicon probes, since it is superimposed on the fluorescence inhibition effect.

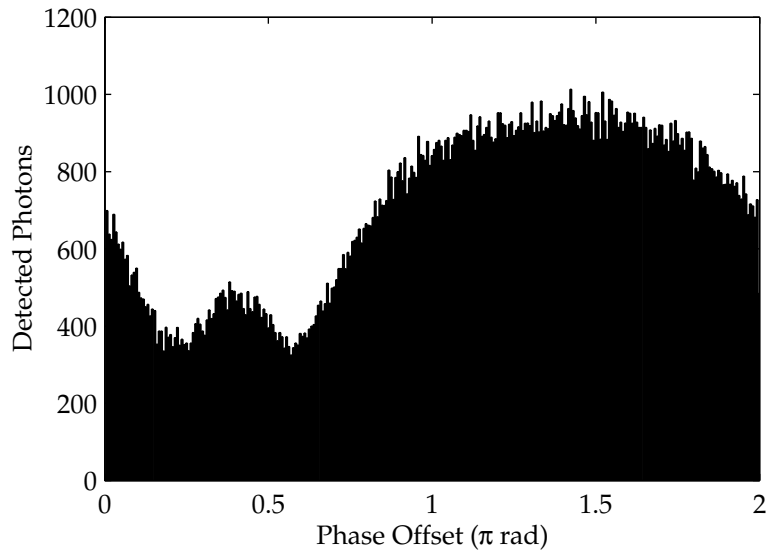
5.3 FANSOM Images

We have acquired images with the goal of evaluating the lateral resolution of the FANSOM. The laser beam was kept stationary as the AFM probe was scanned over a 20 nm bead. In this way, the background fluorescence does not change over the field of view, except for photobleaching—a slow effect compared to the acquisition time of one of these images. For comparison purposes, the contrast in the images shown here is displayed as a positive effect, regardless of whether it was obtained through fluorescence enhancement or inhibition.

Figure 5.13 shows a FANSOM image acquired using a PIT probe with direct illumination (no TIR), resulting in contrast generated in fluorescence inhibition mode. The FWHM of the imaged feature is approximately 78 nm, compared with 48 nm with the AFM image, as shown in the cross-section profiles of Figure 5.14. This difference in image width suggests that the contrast generation mechanism for the FANSOM image had a greater lateral extent than the probe itself. Let us imagine a contrast mechanism whose effect is isotropic, as quenching would be (it depends only on the distance from metal to fluorophore). This mechanism would start being felt when the probe is at a certain distance from the bead, regardless of which part of the probe is in proximity—the side or the tip. With such a mechanism, the interaction distance would effectively increase the apparent size of the probe. We know from the approach curves obtained with PIT probes that the interaction distance vertically is approximately 20 nm (see Figure 5.7). If the same interaction

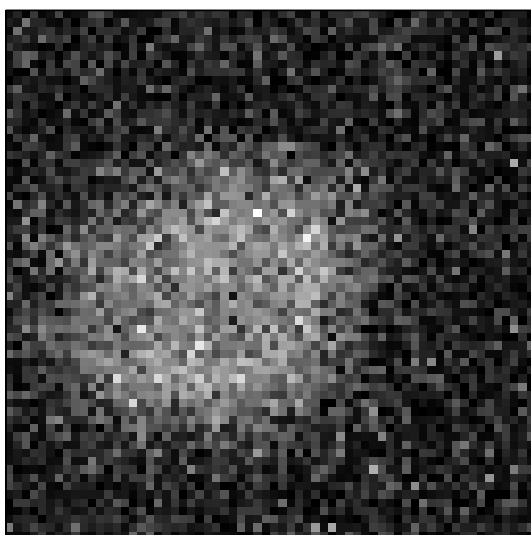


(a) Detected energy as a PIT probe approaches the sample.

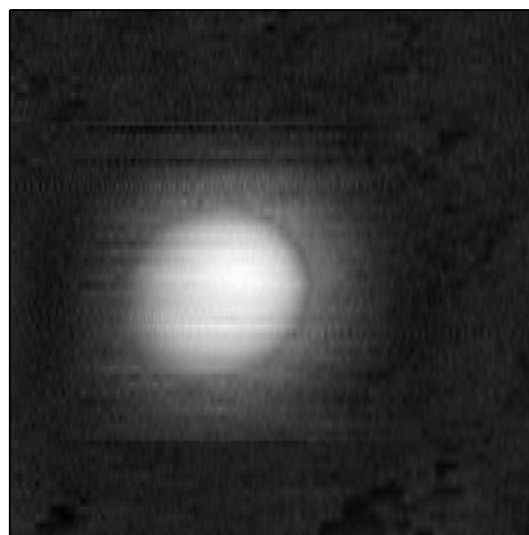


(b) Counted photons binned by phase offset.

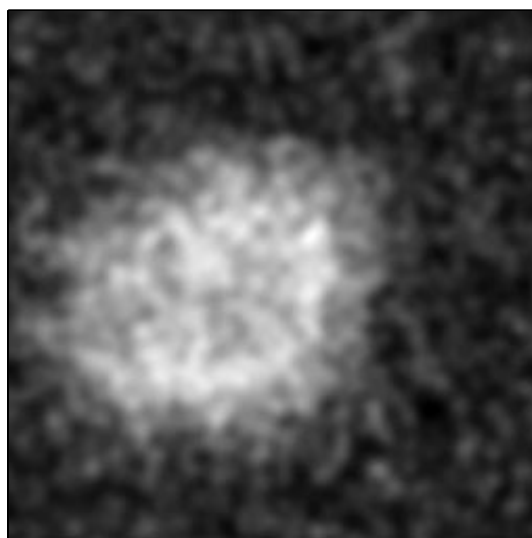
Figure 5.12: Effect of a PIT probe on the fluorescence of a 20 nm bead excited by a TIR illumination with its polarization perpendicular to the sample plane—parallel to the probe axis. This probe had an oscillation frequency of 69.5 kHz and an amplitude of 187 nm.



(a) FANSOM image by correlation analysis (raw).

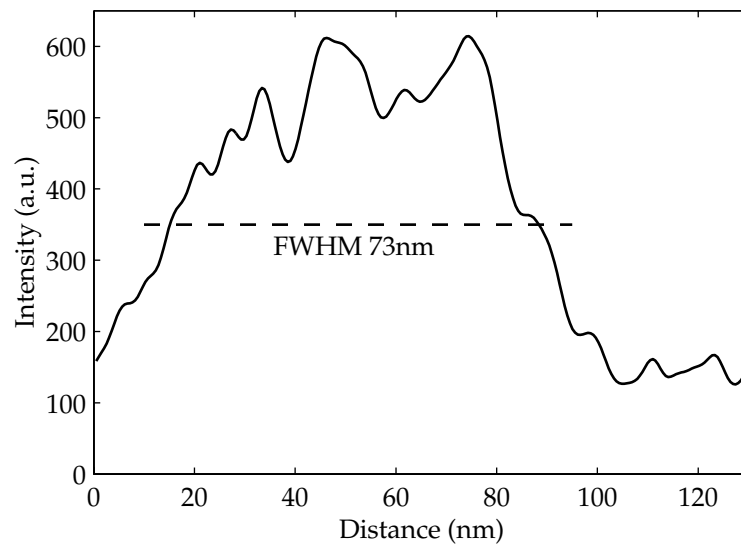


(b) Simultaneously acquired AFM image.

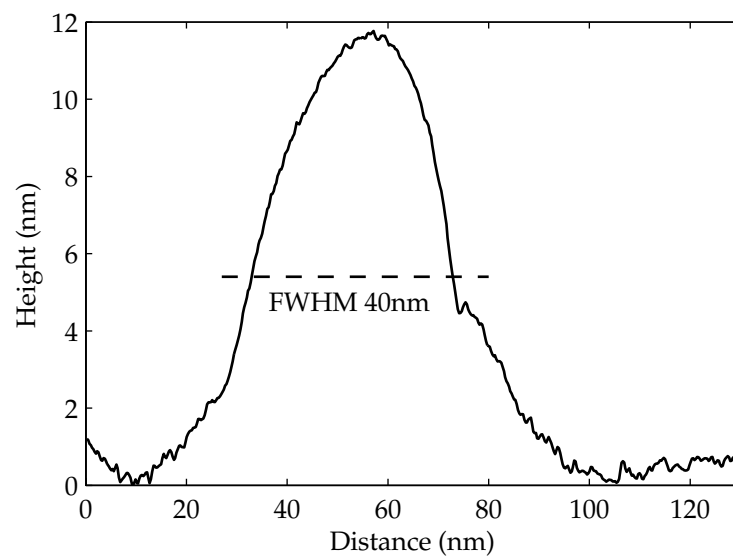


(c) FANSOM image by correlation analysis (filtered).

Figure 5.13: FANSOM image of a 20 nm bead with contrast generated in the fluorescence inhibition mode with a PIT probe. The laser was held stationary while the probe was scanning. The scanned field of view is a 130 nm square. The filtering applied to 5.13(c) was a 0.7 pixel Gaussian blur, applied after an interpolation of the raw image data.

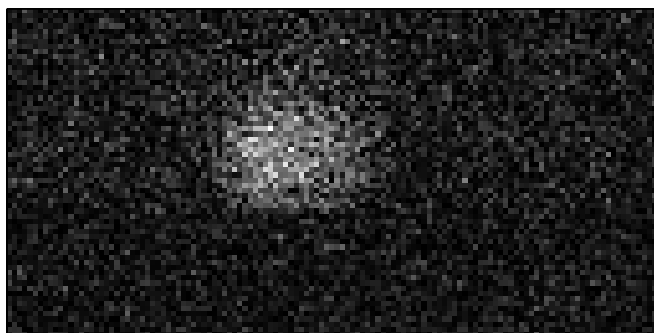


(a) Profile of the FANSOM image.

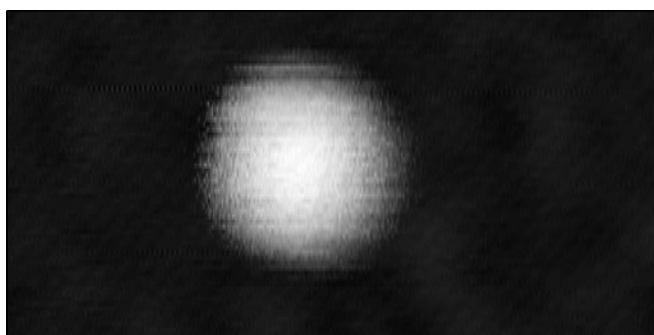


(b) Profile of the AFM image.

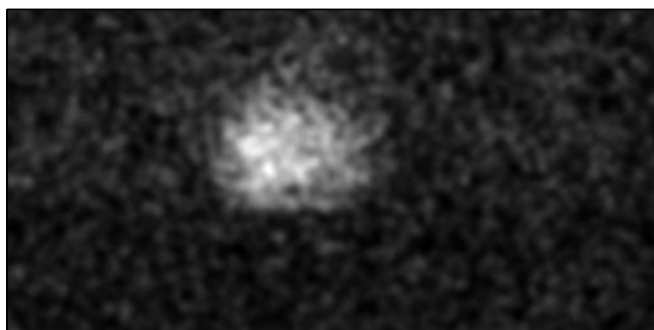
Figure 5.14: Cross sections of the images from Figure 5.13.



(a) FANSOM image by correlation analysis (raw).

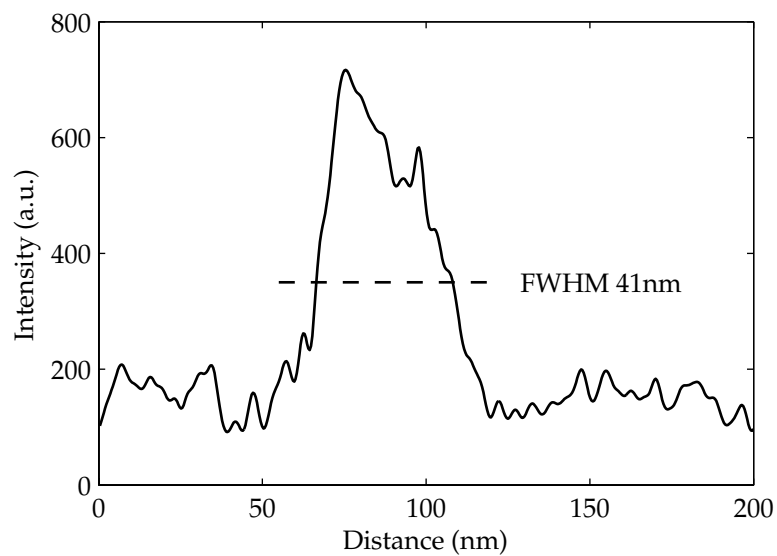


(b) Simultaneously acquired AFM image.

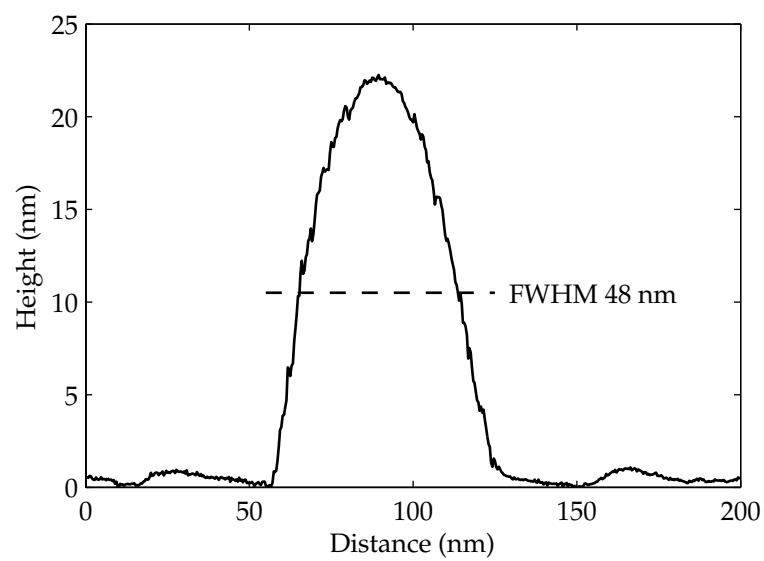


(c) FANSOM image by correlation analysis (filtered).

Figure 5.15: FANSOM image of a 20 nm bead with contrast generated in fluorescence enhancement mode with a silicon (FESP) probe. The laser was held stationary while the probe was scanning. The scanned field of view is 100 nm by 200 nm. The filtering applied to 5.15(c) was a 0.7 pixel Gaussian blur, applied after an interpolation of the raw image data.



(a) Profile of the FANSOM image.



(b) Profile of the AFM image.

Figure 5.16: Cross sections of the images from Figure 5.15.

were to occur laterally, it is logical to expect an optical feature approximately 40 nm larger than the topographical feature. The set of images in Figure 5.13 supports this interpretation, as the optical feature is 30 nm larger than the topographical feature, corresponding to an interaction length of approximately 15 nm.

Figure 5.15 shows a FANSOM image acquired using a silicon probe (FESP type). The full-width at half-maximum of the optical feature is approximately 41 nm, which is similar to the FWHM of the topographical feature. This similarity implies that the field enhancement effect is weak unless the sample is largely below the probe. This agrees well with the calculations of Martin et al. [49], who show enhancement profiles with a full-width at half-maximum of less than 20 nm.

Chapter 6

Conclusion

6.1 Summary

We have demonstrated an apertureless near-field microscope capable of imaging fluorescent features with a lateral resolution better than 20 nm. Along the way, we have demonstrated that it is possible to use the probe of an atomic-force microscope to cause a detectable modulation of the fluorescence of a nanometer-scale fluorescent sample.

The processes that cause this modulation act for very small probe-sample separations, namely less than 20 nanometers. We have demonstrated contrast using two completely different effects; one is fluorescence quenching, the other is fluorescence enhancement through electric field enhancement. Those two effects have allowed us to obtain contrast ratios of 1.90 to 1 and 5 to 1, respectively.

The design of the data acquisition system allows us to obtain the most precise mappings of signal intensity to probe-sample separation that have been achieved yet in an apertureless near-field microscopy system. It has allowed us to obtain repeatable approach curves that differ clearly from one another when the type of illumination or probe is changed.

6.2 Outlook and Future Directions

The resolution capabilities of the FANSOM are far from having been completely exploited. The work presented in this thesis shows FANSOM resolutions that depend directly on sample sizes because of the conical shape of the probes we used. There are two directions in which work must proceed in order to discover the ultimate limit of FANSOM resolution. First, smaller samples must be examined. The leap to single fluorophores is difficult because of photobleaching-related concerns, but another type of “single fluorophores” can be used: fluorescent quantum dots. Their small size would provide not only a better measurement of the lateral resolution obtainable through field enhancement, but also a better measurement of the vertical profile of the electric field obtained

from approach curves.

Another possible investigation into greater FANSOM resolution is the use of intrinsically sharper probes, such as single-walled nanotube probes. The Caltech team of Larry Wade, Pat Collier, Ziyang Ma and Ian Shapiro has already made advances in the production of such probes; it is hoped that carbon nanotube AFM probes can be used to good effect in investigations of FANSOM contrast.

Our method of measuring approach curves provides an excellent sampling of the space between the surface and the probe's oscillation maximum, especially with the FESP class of probes that have an oscillation period of 12 microseconds. With a data acquisition clock of 20 Mhz, a gap-dependent quantity can be sampled 240 times during a period, or 120 sampling points in one swing of the probe. At the point of greatest probe velocity, the sampling points will be spaced at 1/60th of the oscillation amplitude. Considering that it is easy to make an tapping probe oscillate with an amplitude of 60 nm, we have a tool capable of sampling a gap-dependent phenomenon with a 1 nm sampling grid. Such a tool could become valuable for the validation and verification of theories regarding interactions of fields, probes and samples at the nanometer scale.

Appendix A

AFM Probes

A variety of AFM probes were used during the development of the FANSOM. We used silicon and metal-coated probes, where the metal-coated probes were modified versions of the silicon probes with the coatings applied either by us or by vendors.

The silicon probes we have used are listed in Table A.1, along with their nominal resonance frequencies and the names used for them elsewhere in this document. The probes are composed of a pyramid 10 to 15 μm tall, etched out from crystalline silicon; the full angle of the pyramid is approximately 35 degrees along both axes. The tip of the pyramid has a radius of curvature of approximately 10 nm.

Table A.1: Characteristics of silicon probes.

Product Name	Vendor	Force Constant	Resonance ^a	Nickname ^b
PointProbe [®] NCH	NanoSensors ^c	42 N/m	330 kHz	TESP
Tap300	NanoDevices	40 N/m	300 kHz	TESP
PointProbe [®] FM	NanoSensors ^c	2.8 N/m	75 kHz	FESP
Multi75	NanoDevices	3 N/m	75 kHz	FESP

^a The actual resonance frequency of an individual probe varies by $\pm 15\%$ compared to the nominal value.

^b The “nickname” is the name used for the probe type elsewhere in this document. Different probe types that were found to perform identically were used as replacements and did not get new nicknames.

^c NanoSensors was acquired by NanoWorld AG in November 2002.

We used several kinds of metal-coated probes as well. We first applied metallic coatings to probes for measurements using the first embodiment of the FANSOM. The coatings were applied to probes using an evaporator. With this method, gold and silver coatings with thicknesses of 5 to 30 nm were applied to TESP probes.

Later, we used commercially available metal-coated probes with the BioScope-based embodiment of the FANSOM. We used cobalt-coated and platinum/iridium-coated probes, both on the PointProbe[®] FM probes from NanoSensors. The cobalt coating is added by Veeco, applied to wafers

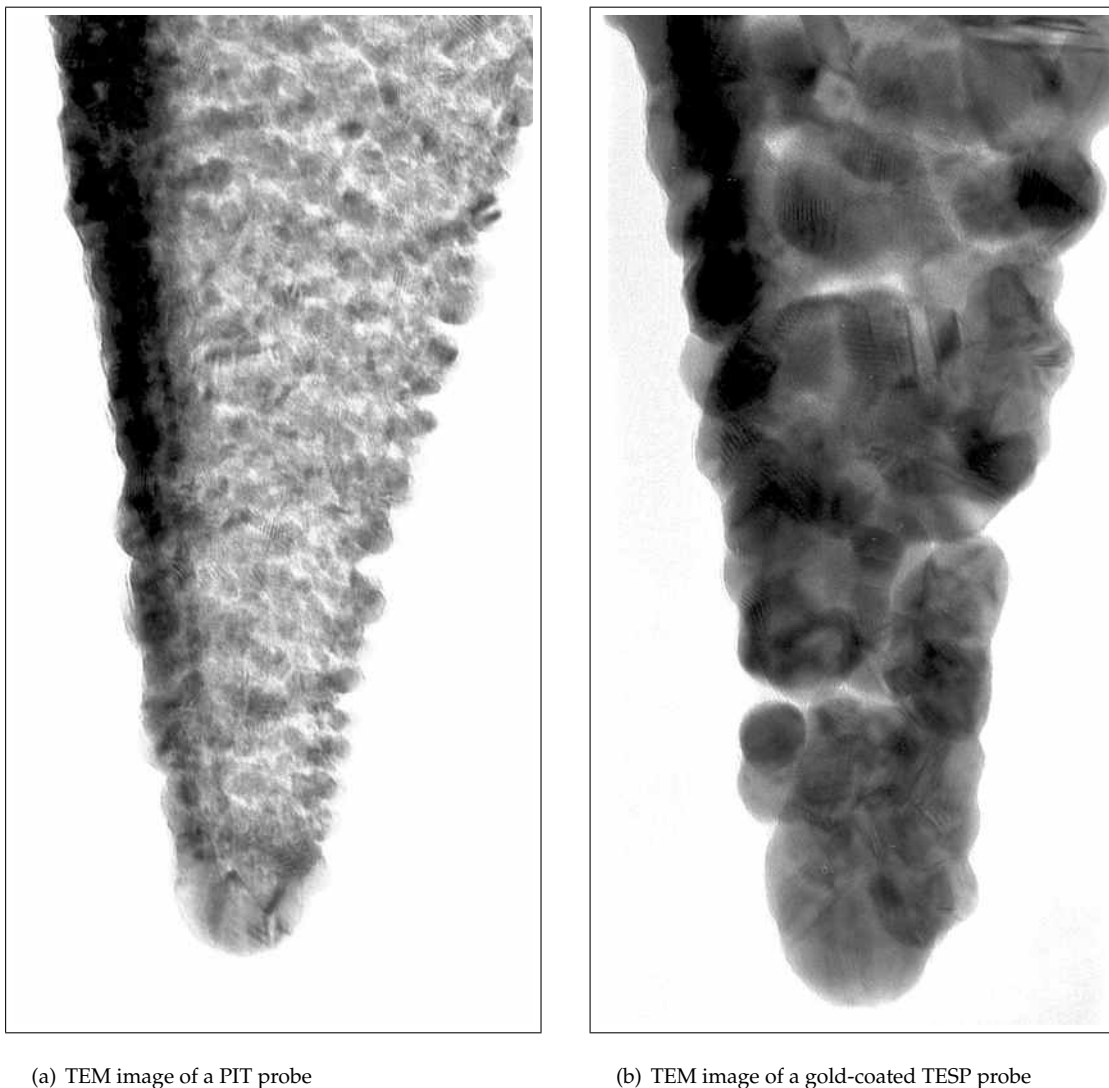


Figure A.1: Transmission electron micrograph (TEM) of (a) a PIT probe, (b) a gold-coated FESP probe. The micrographs were taken at $260\,000\times$ magnification. The width of represented region in each picture is approximately $5\,\mu\text{m}$ – the magnification of the printed images is $\sim 14\,000\times$.

of PointProbe[®] FM probes; these modified probes were too blunt to provide good results. The platinum/iridium coating is added by NanoSensors themselves. It is a 20 nm coating with a sharpened end; the tip-end radius is unspecified, but the half-angle is reported as 20 degrees. We refer to the platinum/iridium-coated probes elsewhere in the document as PIT probes—their part number in the Veeco catalog. A transmission electron micrograph (TEM) of a PIT probe can be seen in Figure A.1(a).

A TEM image of a homemade gold-coated FESP probe can be seen at Figure A.1(b). A gold coating of a nominal thickness of $100\,\text{\AA}$ was applied onto the probe by evaporation. It is clear from the

comparison with Figure A.1(a) that the coating is substantially worse than the commercial coating, despite our best efforts. The tip-end radius of curvature of the coated probe was too large for work on 20 nm beads, although we used similar probes on larger samples to good effect, as shown in Chapter 2. We believe that evaporating the metal onto a heated cantilever/probe assembly would result in a much better coating.

The change from home-coated probes to commercial coated probes was made in light of this difficulty in producing gold-coated probes, since we could obtain satisfactory results using the PIT probes.

Appendix B

Sample Preparation Methods

The samples we have used during the course of most of the FANSOM development were thin cover glasses, sparsely covered with fluorescent latex beads. The preparation of a good sample requires cleaned cover glasses; two techniques for this is described in the first section of this appendix. The fluorescent beads are described in the second section, along with the simple recipe used to apply them onto the glass slide.

B.1 Glass Cleaning Methods

Commercially available glass cover slips have two major problems with respect to the FANSOM experiment: they are very dirty, and their surface is hydrophobic. It is mandatory to clean them before use, especially with very dim sources such as single molecules.

The glass cleaning procedure that has been used until now for the FANSOM experiment has been passed on from person to person in the Quake lab, each time including some new tweaks that have been made here and there. This first procedure is a variant of the so-called Standard Clean 1 (SC1), the first step of the “RCA clean” process, developed by Kern and Puotinen [50]. The relative concentrations of the reagents are modified; in particular, the concentration of ammonium hydroxide is much greater than in SC1.

This difference poses the question of whether we used the best possible recipe. Our testing of the method focused only on determining whether our cover slip cleaning procedure was adequate for FANSOM imaging, and did not go any further.

A second method, based on HCl and NaOH, was attempted and refined by Matthias Scholz while he was summer intern with our group. This newer method results in cover glasses with a slightly better smoothness, although it requires larger amounts of more expensive reagents than the RCA method. It is included here for completeness.

B.1.1 The RCA Cleaning Procedure, or SC1

This procedure solves both of the problems encountered with commercial glass cover slips: it reacts with organic contaminants on the glass surface, altering them and washing them away, and leaves the surface covered with hydroxyl (OH) groups, which make the surface hydrophilic. VWR cover slips (thickness #1) cleaned using this procedure have a surface roughness of 0.5 to 0.7 nm RMS.

The RCA cleaning process was developed at the RCA laboratories by Kern and Puotinen [50], as a cleaning method for silicon wafers. The full procedure is designed to burn organic contaminants and to remove a thin layer of silicon dioxide from the wafers. The full RCA procedure consists of three steps, of which we apply only the first, called “standard clean 1” or SC1. The SC1 step is effective to remove organic contaminants. We use a concentration of ammonium hydroxide four times larger than suggested by Kern. A higher concentration of ammonium hydroxide raises the pH, resulting in an increase of the reactivity of the hydrogen peroxide [50].

The procedure requires at least one 125 × 65 mm crystallization dish, although using more than one can expedite the procedure during the rinsing stages. Preferably, the dishes are very clean at the start of the procedure; performing the RCA procedure on the empty dishes is effective, although probably excessive. Commercial cover slip holders made out of Teflon are sufficient to hold the slips during the procedure and to store them until use.

1. Inside a 125 × 65 mm crystallization dish, put stock cover slips into a cover slip holder.
2. Rinse with high-purity water (HP H₂O).
3. Add 10 ml of “Micro-90” cleaning agent, fill the dish with HP H₂O, then put it in the ultrasonic bath for 20 minutes. Using the ultrasonic bath’s heater is not necessary.
4. Rinse and flush with HP H₂O at least three times.
5. In the beaker, put 300 ml HP H₂O; move the dish to the hood, then add the following reagents:
 - 50 ml Hydrogen Peroxide (H₂O₂, 30% concentration)
 - 200 ml Ammonium Hydroxide (NH₄OH, 28-30% concentration)
6. On a hot plate, boil gently with a stir bar for 1 hour. A final temperature between 62°C and 68°C is appropriate. At higher temperatures, excessive bubbling will adversely affect the quality of the glass surface, probably as a result of cavitation at the surface.
7. Extract the cover glass holder from the mixture and rinse with HP H₂O. Rinse at least three times. This step benefits greatly from a second dish, as vaporized ammonia is rising from the dish—taking it out of the hood before rinsing means having to breathe this vapor.
8. Store the cover glasses in HP H₂O until they are needed.

B.1.2 HCl/NaOH Cleaning Procedure

This method was investigated with the hope that the resulting cover slips would have a better surface roughness than if they were cleaned by the RCA method. After cleaning, we obtained surface roughnesses between 0.25 and 0.5 nm RMS using FISHERfinest™ brand cover glasses from Fisher Scientific.

As in previous procedure, we use one 125 × 65 mm crystallization dish. Commercial cover slip holders made of Teflon hold the slips during the procedure and store them until use.

1. Inside a 125 × 65 mm crystallization dish, put stock cover slips into a cover slip holder.
2. Rinse with high-purity water (HP H₂O)
3. Add 10 ml of “Micro-90”, fill the dish with HP H₂O, then put it in the ultrasonic bath for 20 minutes. Using the ultrasonic bath’s heater is not necessary.
4. Rinse and flush with HP H₂O at least three times.
5. Put a solution of NaOH (30% by weight) in the dish. Sonicate for 1 hour at room temperature.
6. Rinse and flush until a pH of 6.5 is reached.
7. Put a solution of HCl (1 Molar) in the dish. Sonicate for 1 hour at room temperature.
8. Rinse and flush with HP H₂O until a pH of 6.5 is reached.
9. Store the cover glasses in HP H₂O until they are needed.

B.2 Fluorescent Bead Sample Preparation

The beads we used in the development of the FANSOM are fluorescent latex beads sold by Interfacial Dynamics Corporation (IDC). The fluorophores in the IDC beads are scattered throughout the volume of the bead, rather than only at the surface as is the case with similar products from other vendors.

We have used beads colored with the fluorescent dye “Nile Red”. It has an absorption peak at 580 nm and an emission peak at 600 nm; it is easily distinguishable from the 670 nm energy from the AFM probe-tracking laser. It is compatible with affordable green lasers such as the green HeNe or the frequency-doubled YAG. Its emission peak is broader than the emission peak of rhodamine 6G, a common red fluorophore. We used beads made with a sulfate-stabilized latex.

The beads from IDC are delivered in a suspension containing approximately 2% solids. This concentration is much too high for our purposes, and it is necessary to dilute the beads in high-purity water before depositing them onto a glass slide. The amount of dilution necessary varies

according to the size of the beads, since the volume fraction is constant across bead sizes. Table B.1 shows the nominal dimensions of the beads, along with the dilution ratios we used.

Table B.1: Nominal characteristics of IDC latex beads.

Diameter (nm)	C. V. (%)	Vol. Fraction (%)	Dilution
20	15	2	10^{-5}
60	15	2	10^{-4}
100	5	2	10^{-3}

The coefficient of variance (C. V.) is the ratio of the standard deviation to the mean. Since the number of fluorophores is proportional to the volume of the beads, there is substantial variability in the brightness of the beads; it is not uncommon to see a bead that is 50% brighter than its neighbor.

The beads are deposited on a cover glass that has been cleaned according to the method described in Section B.1.1. Approximately 20 μl of the diluted bead suspension is deposited onto the clean glass slide; after a few seconds, the cover glass is blow-dried in a stream of nitrogen. The cover glass is then ready to be used as a sample. It is generally easy to find regions with areal concentrations of 1 to 5 beads/ μm^2 on the sample.

Appendix C

Fabrication of Aluminum Masks

The laser beam that illuminates the FANSOM's sample arrives to the sample plane by way of a high numerical aperture (N. A.) microscope objective. The objective's N. A. is greater than the critical angle of the glass-air interface at which the sample is located. This results in an optical field composed of both traveling and evanescent wave components (see Figure 3.5). These components have different polarization properties (see Figure 3.6), and separating them is necessary in order to elucidate the processes responsible for contrast in the FANSOM.

We separate the components of the illumination by inserting masks in the path of the laser beam. Each mask is either a layer of gold patterned on the surface of an optical window or a piece of aluminum foil with a patterned opening.

In either case, the fabrication involves etching aluminum foil in patterns transferred by a lithographic technique. The technique was selected for its flexibility and seemed to require an amount of work similar to machining, the other option we found viable.

C.1 Mask Design

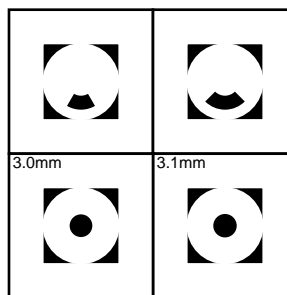


Figure C.1: A mask design for illumination beam shaping. The openings in the mask (in black) are a 60° crescent, a 90° crescent, a 3.0 mm dot and a 3.1 mm dot.

The designs are printed on a transparency using a high-resolution imagesetter. The printed pattern must be opaque where the mask should be opaque. Figure C.1 shows some examples

of mask designs that have been fabricated. They were designed using Adobe® Illustrator® and printed in negative.

The important parts of the masks in Figure C.1 are the 0.4 inch disks; each is attached to a square frame, in order to facilitate the manipulation of the small pieces of foil obtained as a result of the fabrication procedure. The frame can be removed when mounting the mask for utilization.

C.2 Wafer Preparation

Silicon wafers are used as substrates for the procedure, serving as rigid supports. They were chosen for their ready availability in the clean room.

First, a silicon wafer is covered with photoresist. We used Microposit SJR 5740 photoresist from Shipley, spin-coated at 2000 revolutions per minute (r.p.m.) for 60 seconds. Then, a piece of aluminum foil is laid on top of the wafer, covering it entirely. It is trimmed to the wafer's size using a blade. The aluminum foil is then covered with a negative photoresist (5740 is such a photoresist), using the same parameters as previously used for the spin-coating step.

If the photoresist coating is sufficiently even, the wafer is baked on a hot plate for 2 minutes at 90°C. Uneven coatings do not produce good results in the etching step.

C.3 Lithographic Transfer

The transfer of the pattern from the transparency to the photoresist is made during a 30 second exposure to ultraviolet light (UV) The wafer is then immersed in a developer solution composed of four parts water and one part developer. We used Microposit 2401 developer from Shipley, as recommended for the 5740 photoresist. When the UV-exposed parts of the developer have dissolved away, the wafer is dried using a jet of nitrogen.

The wafer is baked again, this time for 20 minutes at 90°C, in order to harden the remaining photoresist prior to etching.

C.4 Aluminum Etching

The aluminum etching is conducted using a variant of the so-called PAN etching procedure. The etchant solution is composed of 8 parts phosphoric acid, 1 part acetic acid and 1 part nitric acid. Various sources quote slightly different proportions.

The etchant solution is heated to a temperature around 40°C and agitated using a Teflon stir bar. Given that the goal of the etching is to completely remove the aluminum layer where it is exposed, little attention has been paid to the actual etch rate; this particular etchant solution at

the specified temperature will etch all the way through a 75 μm -thick of aluminum foil in about 20 minutes. Using a higher temperature results in faster etch rates, but results in jagged edges around the pattern—an undesirable outcome.

Visual inspection of the wafer in its etchant bath is sufficient to determine when the etching process is complete: there is a visible reduction of the amount of bubbling and the dominant color becomes the dark red-brown of the photoresist, rather than that of the aluminum. The reaction is allowed to continue for approximately one minute, after which the wafer is extracted from its acid bath.

After a thorough rinsing of the wafer in a water bath, the remaining photoresist can be removed by squirting acetone on the wafer.

C.5 Mask Removal

In some cases, the small aluminum pieces remain stuck to the underlying photoresist, and separating them from the wafer without destroying them is a delicate task. Immersing the wafer in acetone with light heating (30°C) can help in separating the masks from the wafer. Once the masks have been thoroughly rinsed with acetone, they are ready for use.

Appendix D

Data Formats and Data Analysis

This appendix describes the data formats used by the FANSOM driver, along with the C++ and MATLAB[®] code used to respectively write and read the data. Section D.2 contains the data analysis scripts used to generate the results shown in Chapter 5.

D.1 Data File Formats

Two main types of data files are used to save data from the FANSOM driver. They are image files and timestamp files. The image files have been used since the beginning, and contain topography data from the homemade AFM, far-field optical data or FANSOM data from the analog version of the instrument. Timestamp files are the outputs from the second-generation instrument.

D.1.1 AFM and Optical Images

The AFM images saved by the FANSOM software are represented as sequences of 16-bit values. For simplicity, the software always acquires a square number of pixels (i.e., as many pixels per lines as there are lines in the image). The images from non-FANSOM optical scans have a similar format, but with 32-bit values.

The following method of class `LDaq` is the method used to write image files. The parameter `pixelWidth` contains the “width” in bytes of the pixel data, i.e., 2 bytes for 16-bit data and 4 for 32-bit data. The method writes the data from the array `image` containing `lines` lines of `cols` pixels to a new file whose name is given by the string `fname`.

Listing D.1: Method `LDAQ::WriteImage()`.

```

void LDAQ::WriteImage(const short *image, const char *fname,
                      long lines, long cols, long pixelWidth)
{
    DWORD linebytes = cols*pixelWidth;
5   DWORD written;
    HANDLE fout;
    BOOL okay;

    const short *outbytes;

10   // Ask Win32 for a file handle
    fout = CreateFile(fname, GENERIC_WRITE, FILE_SHARE_READ,
                      NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);

15   // lines is the number of lines in the image.
    for(long i=0; i<lines; i++)
    {
        // We *could* write the whole file in one huge write operation,
        // but it's not that good an idea. Besides, this will more
20   // straightforwardly be extensible to arbitrary-length images...

        // cols is the array-length of one row; let's do pointer arithmetic
        outbytes = (image + i*cols);

25   // Tell Win32 to write data to file
        okay = WriteFile(fout, outbytes, linebytes, &written, NULL);
        if ((written != linebytes) || (okay == false))
        {
            break;
30   }
    }

    // Clean up, whether we were successful or not.
    CloseHandle(fout);
35 }

```

Images files saved with the method `LDAQ::WriteImage()` from Listing D.1 are read by the function `getrect` in MATLAB®. This function is used by the aptly-named higher-level functions `getafm` and `getoptical` to read image data with respectively 16-bit values and 32-bit values.

Listing D.2: Function `getrect`.

```

function data = getrect(fname, datatype, aspectratio)

if nargin < 3
    aspectratio = 1;
5 end;

mine = fopen(fname, 'r', 'ieee-le');

dat = fread(mine, datatype);
10 frewind(mine);

c = length(dat);

```

```

    s = floor(sqrt(c/(2*aspectratio)));
    if (2*s^2 == c)
15    Dim = [2*aspectratio*s,s];
    else
        s = sqrt(c/aspectratio);
        Dim = [aspectratio*s,s];
    end
20
    [raw,count] = fread(mine, Dim, datatype);
    fclose(mine);

    if nargout == 1
25        % transposition
        data = raw';
    end;

```

The result of the `getrect` function is a two-dimensional array that can be displayed with the standard MATLAB[®] functions `image` and `imagesc`. Note that some of the very old AFM image files contain rectangular arrays with a 2:1 aspect ratio, rather than square arrays – thus the `aspectratio` argument to `getrect`.

Optical images acquired using the FANSOM driver since the summer of 2002 are saved with an auxiliary file that contains some parameters of the data acquisition. The most notable parameters are the coordinates of the image's corners in the scannable area of the optical microscope.

Listing D.3: Function `vimeta`.

```

function [meta] = vimeta(fname)

try
    fileid = fopen([fname '.rmd'], 'r');
5
    dat = fread(fileid, 'char');
    frewind(fileid);

    % Amplification factor (range)
10    meta(1) = fread(fileid, 1, 'double');

    % Number of pixels
    meta(2) = fread(fileid, 1, 'unsigned_long');

15    % Acquisition frequency (of pixels)
    meta(3) = fread(fileid, 1, 'double');

    if (length(dat) > 40)

20        % Skewness factor
        meta(4) = fread(fileid, 1, 'double');

        % Stretch factor
        meta(5) = fread(fileid, 1, 'double');

25        % Lower X bound
        meta(6) = fread(fileid, 1, 'double');

```

```

    % Lower Y bound
30    meta(7) = fread(fileid, 1, 'double');

    end

    fclose(fileid);
35
catch
    meta(1) = -1;
end

```

The data read by `vimeta` is written out by the `WriteOutput()` method of class `UserData` found in the FANSOM driver code (the `UserData` class is not shown here). The following example illustrates the use of the output from `vimeta`:

Listing D.4: An example of the use of image metadata.

```

function display(filename)

% DISPLAY
%
5 % Display the image contained in the file pointed to by FILENAME,
% using its corresponding metadata.

imdata = getoptical(filename);

10 meta = vimeta(filename);

if (meta(1) ~= -1)
    if (length(meta) > 5)

15        % If META is longer than 5, then it contains the coordinates of
        % the four corners of the image

        rangex = [meta(6) (meta(6)+meta(1))];
        rangey = [meta(7) (meta(7)+meta(1))];
20
    else

        % Otherwise, it only contains the length of a size of the image

25        rangex = [0 meta(1)];
        rangey = rangex;

    end

30    imagesc(rangex, rangey, imdata);

else

    % Since there is no meta data, display without meaningful axes
35    imagesc(imdata);

end

```

D.1.2 Timestamp Data Files

The timestamp data from the second-generation FANSOM (Chapter 3) are saved line by line. Each individual file is labelled

`<basefilename>-<NNN>-<C>d.raw,`

where `<basefilename>` is the base file name as entered in the `./dcafm` control panel, `<NNN>` is the line number indexed from zero in a 3-digit field, and `<C>` is the channel number. In general, the scripts that handle this kind of file automatically take care of the channel number, line number and file extension.

The utility function written to open a file from a named dataset for a given line and channel number is `stripttfile`; it is listed here along with the methods it calls. The data saved in the file may contain some extraneous samples. The function `timestampstrip` cleans up the data, by removing the duplicate values that are inserted at the beginning and end of the file by one of the versions of the timestamp acquisition software, and then removing any impossible values from the array, i.e., values that violate the physics of the data acquisition. The impossible values are inserted in the array as a result of National Instruments™ hardware and/or software defects. An impossible value is characterized by being larger than the value that follows it or by being smaller than the value that precedes it; such values are impossible since the counter is only counting up, and it is the value of the counter that is copied to memory.

Listing D.5: Function `stripttfile`.

```

function data = stripttfile(base, line, channel)

% STRIPTTFILE utility script to open a FANSOM timestamp file
% Usage: stripttfile(base, line, channel)
5 % BASE is the base filename for the dataset we're looking at.
% LINE is the line number in the dataset.
% CHANNEL is the channel number for which we want to get the
% timestamps.

10 % Get the full filename
    f = [base '-' num2str(line, '%03d') '-' num2str(channel, '%01d') 'd.raw'];

    % Get the data from the file
    d = getlin(f, 'int32');

15 % Clean it up
    data = timestampstrip(d);

```

Listing D.6: Function timestampstrip.

```

function [r] = timestampstrip(data)

% TIMESTAMPSTRIP
% Strips any duplicates of the first and last values of a vector,
5 % detects instances of a 24-bit counter rolling over, and
% removes any data which is not monotonically increasing. The
% cleaned-up vector is output as the return value.
%
% (C) 2002,2003 Guillaume Lessard
10 % All rights reserved.

% Detect instances of a 24-bit counter rolling over, and fix them.

dd = diff(data);
15 indices = find(dd < -1e7);
for i=1:length(indices),
    a = indices(i)+1;
    if i<length(indices),
        b = indices(i+1);
20    else
        b = length(data);
    end
    data(a:b) = data(a:b) + i*2^24;
end
25
% Strip duplicate values, and at the same time remove any out-of-order
% time stamps that got inserted by the bugs of the National
% Instruments hardware and/or software.

30 % duplicate values and high outliers.

dd = diff(data);
data(dd<=0) = 0;
data = nonzeros(data);
35
% low outliers get zapped here.

dd = diff(data);
indices = find(dd <= 0);
40 data(indices + 1) = 0;
data = nonzeros(data);

r = [ 0; data ];

```

Listing D.7: Function getlin.

```

function out = getlin(fname, datatype)

% GETLIN wrapper to get a vector of values of a given type
% Usage: getline(fname, datatype)
5 % FNAME is the name of the file to open in order to get the vector
% DATATYPE is the type of data to extract from the file.
%
% Data from the FANSOM project is 'int16', 'int32' or 'uint32',
% depending on the nature of the measurement. Note that 'int32'
10 % was used for only a short time in 1999.
%
% Type HELP FREAD for a list of applicable data types.
% Type HELP FOPEN for more info on endianness in files.

15 mine = fopen(fname, 'r', 'ieee-le');
raw0 = fread(mine, datatype);
fclose(mine);

out = raw0;

```

D.2 FANSOM Data Analysis Scripts

D.2.1 Photon Phase Offset Calculation

Listing D.8: Function convert2phases.

```

function phases = convert2phases(stamps, taps)

%CONVERT2PHASES Convert photon timestamps to corresponding phases
% Usage: convert2phases(stamps, taps)
5 % Each timestamp in the vector "stamps" will be replaced in
% the output by its distance in ticks to the (then) most recent
% probe timestamp.
%
% (C) 2002,2003 Guillaume Lessard
10 % All rights reserved.

phases = zeros(size(stamps));

j=1;
15 for i=1:length(stamps),
    try
        while (stamps(i) >= taps(j+1)),
            j = j+1;
        end
20 catch
    end

    phases(i) = stamps(i) - taps(j);
end

```

The code of function `convert2phases` is relatively slow when interpreted by MATLAB®. To remedy this, a native C version was written; the following C code is intended to be compiled with the MATLAB® `mex` command, and runs nearly 20 times faster than the interpreted script on MATLAB® version 6.5. The speedup is nearly 50 times compared to version 6.1, which does not have on-the-fly compilation of scripts.

Listing D.9: Mex Function `convert2phases`.

```

/**
 * convert2phases
 *
 * For each photon timestamp in the input vector, this finds the time
5 * offset between its timestamp and the previous AFM probe tap.
 *
 * The above makes sense in the context of FANSOM data acquisition.
 *
 * The two arguments are the photon timestamp vector and the probe
10 * taps timestamp vector, respectively. The output argument is a
 * vector of timestamp offsets.
 *
 * This is a MEX-file for MATLAB.
 * (C) 2003 Guillaume Lessard
15 * All rights reserved.
 **/

/* $Revision: 1.3 $ */

20 #include "mex.h"

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
{
    /* Examine input (right-hand-side) arguments. */
25 if (nrhs != 2)
    mexErrMsgTxt("convert2phases_requires_two_inputs.\n");

    if (nlhs != 1)
    mexErrMsgTxt("convert2phases_requires_one_output.\n");
30
    int mrows = mxGetM(prhs[0]);
    int ncols = mxGetN(prhs[0]);
    if (!mxIsDouble(prhs[0]) || mxIsComplex(prhs[0]) ||
        !mxIsDouble(prhs[1]) || mxIsComplex(prhs[1]))
35 {
        mexErrMsgTxt("Inputs_must_be_vectors_of_real_double_values.\n");
    }

    int nd = mxGetNumberOfElements(prhs[0]);
40 int nt = mxGetNumberOfElements(prhs[1]);

    /* Create matrix for the return argument. */
    plhs[0] = mxCreateDoubleMatrix(mrows, ncols, mxREAL);

45 /* Get pointers to the data we're processing */
    double* stamps = mxGetPr(prhs[0]);
    double* taps = mxGetPr(prhs[1]);

```

```

double* phases = mxGetPr(plhs[0]);

50  int i,j;
    for (i=0, j=0; i<nd; i++)
    {
        while (((j+1)<nt) && (stamps[i] >= taps[j+1]))
        {
55      j++;
        }
        phases[i] = stamps[i] - taps[j];
    }
}

```

D.2.2 Image Generation Functions

FANSOM images are generated line by line from the timestamp files. The script `corrline` listed here can be used to obtain the correlations of a single line to a sine and a cosine. Once all the data for a scan has been processed this way, the lines are put that together. The function `corrimage` can then analyze the correlation to the sine and cosine in order to find the “best” correlation angle, and will project its inputs onto that angle.

Listing D.10: Function `corrline`.

```

function [sinbins, cosbins] = corrline(base, line, bins)

% CORRLINE correlate a line of timestamps with a sine and a cosine
%
5  % (C) 2003 Guillaume Lessard
  % All rights reserved.

% Get the timestamp data out of the files first. Photons first...
try
10  d = stripttfile(base, line, 1);
catch
    d = stripttfile(base, line, 2);
end;

15  % ...and then taps.
    t = stripttfile(base, line, 0);

% Prepare sinewave and cosinewave of period PER and order K
per = mean(diff(t(2:end-1)));
20  k = 1;
    sinw = sin((2*pi*k/per)*(1:ceil(1.1*per)));
    cosw = cos((2*pi*k/per)*(1:ceil(1.1*per)));

% The value we will use as the highest possible timestamp
25  linend = t(end)+1;

% Get the phase data
phasedata = convert2phases(d,t);

```

```

sn = zeros(bins, 1);
cs = zeros(bins, 1);

selend = 0;
35 selstart = 1;
for i = 1:bins,
    % Identify the boundaries of the current pixel
    b = floor((i-1)*linend/bins);
    e = floor(i*linend/bins);
40
    % Identify the indices of the photons closest to the pixel boundaries
    selend = gettimestampselection(d, selstart, b, e);

    if nargout > 1,
45
        % Use the indices we found to select a portion of the phasadata vector
        phases = phasedata(selstart+1:selend);
        phases = phases(phases<length(sinw));

50
        % Integrate
        cs(i) = sum(cosw(phases+1));
        sn(i) = sum(sinw(phases+1));

    end
55
    selstart = selend;
end

if nargout >= 2,
60
    sinbins = sn;
    cosbins = cs;
else
    sinbins = sn;
end

```

The `corrline` function relies on `gettimestampselection` to find the boundary points of its bins. It finds the boundary points by using the bisection method, which is a lot faster than the linear search we would get by using MATLAB[®] primitives, because they don't know anything about the structure of the data—we do.

Listing D.11: Function `gettimestampselection`.

```

function f = gettimestampselection(d, s, b, e)

% GETTIMESTAMPSELECTION
% Usage: gettimestampselection(D, S, B, E)
5 % D a vector of monotonically increasing values
% S a starting point, an index in D
% B the low value of the current bin
% E the high value of the current bin
%
10 % This will find the largest value in D that is smaller than E,
% using the Bisection Method.
% This is the equivalent of the following call:
%     max(D(D<E))
%

```

```

15 % ...except that this function is much, much faster on big data,
% because it runs in logarithmic time rather than linear time.
% This is possible because a crucial element of the structure
% of D is known: it's monotonically increasing.
%
20 % The search will start at index S, for further time-saving.
%
%
% (C) 2003 Guillaume Lessard
% All rights reserved.
25
% Find a power-of-two increment based on the number of times
% the value (E-B) is contained in the value D(end).
increment = ceil(2^(2+nextpow2(length(d)/floor((d(end)+1)/(e-b)))));

30 i = s+increment;
increment = floor(increment/2);

while increment > 0,
    if (i<1)
35         i = s;
    end
    if (i > length(d)),
        i = length(d);
    end
40
    if d(i)>e,
        i = i - increment;
    elseif (d(i) == d(end)),
        increment = 0;
45    else
        i = i + increment;
    end

    increment = floor(increment/2);
50 end

% This sometimes ends up on the wrong side of E. Fix it.
if d(i)>e,
    i = i-1;
55 end

f = i;

```

Listing D.12: Function corrimage.

```

function out = corrimage(imsin, imcos);

% CORRIMAGE
%
5 % Use the result of correlation with a sine and a cosine
% to find the optimal phase for the correlation.
%
% (C) 2003 Guillaume Lessard
% All rights reserved.
10
% Get magnitude of correlation (pixel by pixel)

```

```

    immag = sqrt(imsin.^2 + imcos.^2);

15 % Find a number of pixels whose values are significantly bigger
    % than the average. Presumably, those are the significant ones as
    % far as our signal goes...

    mags = reshape(immag, prod(size(immag)), 1);
20 criterion = mean(mags) + 0.5*std(mags);
    selection = (immag>criterion);

    % For the pixels we found, get an average correlation phase

25 selsin = mean(imsin(selection));
    selcos = mean(imcos(selection));

    selarg = atan2(selsin, selcos);

30 % Remembering the rules for a dot product, we should be able to get
    % a projection of our input matrices onto the angle we just found,
    % simply by using its sine and cosine.

    unitselsin = sin(selarg);
35 unitselcos = cos(selarg);

    % This should have the best correlation
    projected = imsin*unitselsin + imcos*unitselcos;

40 % This should have the worst correlation
    eradicated = imcos*unitselsin + imsin*unitselcos;

    if nargout > 0,
        out = projected;
45 end

```

The `linebinfft` function is an implementation of FANSOM data analysis by FFT. It is included for completeness, since it was superceded by the much faster function `corrline`.

Listing D.13: Function `linebinfft`.

```

function [fp, rp] = linebinfft(fn, bins)

% LINEBINFFT
%
5 % FANSOM analysis by FFT
%
% (C) 2002,2003 Guillaume Lessard, Jordan G. Gerton
% All rights reserved

10 try
    d = timestampstrip(getopticalline([fn '-2d.raw']));
catch
    d = timestampstrip(getopticalline([fn '-1d.raw']));
end;

15 t = timestampstrip(getopticalline([fn '-0d.raw']));
    freq = mean(diff(t(2:end-1)));

```

```

m = max(t)+1;

20 peakbins = zeros(bins, 1);
   meanbins = zeros(bins, 1);

   td = zeros(d(end)-d(1)+1, 1);
   td(d - (d(1)-1)) = 1;
25
   for i = 1:bins,
       b = floor((i-1)*m/bins);
       e = floor(i*m/bins);

30   ft = fft(td(b+1:e), 2^nextpow2(e-b));
       peakindexapprox = round(length(ft)/freq;
       range = (peakindexapprox:peakindexapprox+2);

       fp(i) = sum(abs(ft(range)));

35   if (fp(i) > 0),
       morn = ft(peakindexapprox-round(length(ft)/500):peakindexapprox-10);
       norm = mean(abs(morn));

40   rp(i) = fp(i)/norm;
       else
       rp(i) = 0;
       end
   end
end

```

D.2.3 Approach Curve Generation Functions

The two scripts that follow generate an approach curve from a FANSOM dataset. The required pieces of information are a filename and the oscillation amplitude of the AFM probe at the time of data acquisition.

Listing D.14: Function timestamphist.

```

function ph = timestamphist(base, lines, bins)

% TIMESTAMPHIST
% Usage: timestamphist(base, lines, bins)
5 % BASE is the base file name of the dataset
% LINES is the number of lines of data to accumulate
% BINS is the width of the histogram bin we want
%
% Omitting the BINS parameter is equivalent to setting it to 1.
10 %
% (C) 2002,2003 Guillaume Lessard
% All rights reserved.

% Set BINS to 1 if we didn't get it as a parameter
15 if nargin < 3,
    bins = 1,
end

```



```

phases = [];
20 for i=0:(lines-1),
    % First, we open the files...
    try
        phot = stripttfile(base, line, 1);
25    catch
        phot = stripttfile(base, line, 2);
    end;
    taps = stripttfile(base, line, 0);

30    % Get time offsets
    data = convert2phases(phot(1:newend), taps);

    % Put the time offsets in a histogram
    h = hist(data, (0:bin:max(data)));
35
    % H and PHASES may not have the same length, so we make an effort
    % to make them the same length before adding them together...

    hlen = length(h);
40    plen = length(phases);
    ldif = hlen - plen;

    if (ldif <= 0),
        phases(1:hlen) = phases(1:hlen) + h;
45    else
        phases((end+1):hlen) = zeros(1, ldif);
        phases = phases + h;
    end
end
50 ph = phases;

```

Listing D.15: Function acfromhist.

```

function = acfromhist(phases, amplitude)

% ACFROMHIST
% Usage: acfromhist(phases, amplitude)
5 %
% Get an approach curve from a phases histogram and the oscillation
% amplitude (in nanometers) of the AFM probe that generated the
% contrast.
%
10 % (C) 2002,2003 Guillaume Lessard
% All rights reserved.

% Find the boundary

15 boundary = fourierapproxmin(phases, 4);

% Recenter the histogram on the trajectory's extrema.

newphases = symmetrize(phases, boundary);
20
% Calculate the Fourier coefficients for the recentered histogram.

```

```

    fouriercoefficients = fourierexpansion(newphases, order);

25 % Get the waveform with no 'sine' components.

    fouriercoefficients{3} = [];
    fourierfit = fourierapproximation(length(phases), fouriercoefficients);

30 period = length(fourierfit);
    pos = amplitude*(0.5-0.5*cos((1:period)*(2*pi/period)));

    % Renormalize the phases histogram we got as a parameter relative to
35 % its average from the portion of the phase space where the AFM
    % probe is far from the surface (according to our calculations)

    range = (ceil(0.45*length(fourierfit)):floor(0.55*length(fourierfit)));
    normalisation_value = max(fourierfit(range));
40
    modphases = newphases/normalisation_value;

    % Plot an approach curve!

45 fig = figure;
    h = line(pos, fourierfit/normalisation_value);
    set(h, 'Color', [1 0 0]);
    set(h, 'Linewidth', 2);

50 h = line(pos, modphases);
    set(h, 'Marker', '.');
    set(h, 'LineStyle', 'none');

    ylims = get(gca, 'YLim');
55 set(gca, 'YLim', [0 0.1*ceil(10*max(modphases))]);

    xlabel('tip/sample_separation_(nm)');
    ylabel('normalized_count_rate');

60 set(fig, 'Visible', 'on');

    title(['Boundary:_' num2str(boundary)]);

```

Appendix E

Data Acquisition Code

This appendix contains sizeable samples of the data acquisition code used by the FANSOM apparatus. The code that acquires the old AFM topographic data is included, since that class has been the interface for data acquisition since the early days of the FANSOM. The other code listings of this appendix show current FANSOM driver code as of May 2003.

E.1 AFM and Optical Data Acquisition

The AFM data acquisition of the old embodiment of the FANSOM was performed by code in class `LDaq`. That class is also the superclass of and entry point for all the data acquisition code that was later written, so it has never gone away.

The entry point of data acquisition is the method `LDaq::ExecuteMeasurement()`. It calls a succession of other methods that break up the measurement into a series of simpler operations, so that only a subset of the methods needs to be overridden by any given subclass of `LDaq`. Listing E.2 omits some trivial portions of the code of `LDaq`.

Listing E.1: Class Header `ldaq.h`.

```

#ifndef __LDAQ__
#define __LDAQ__

// Needed header files
5
#include "userdata.h"
#include "stagepos.h"

/**** Line DAQ
10
*
* LDaq is the master class. It does line-by-line analog data
* acquisition, internally synchronized with the output scans using
* pulses generated by one of the counters.
*
15
* (C) 1997-2003 Guillaume Lessard
* All rights reserved.
```

```

    */

class MainWindow;

20 class LDaq
    {
        friend class MainWindow;

25     long    linelen;

    protected:
        short **image;
        short *sampleBuffer;

30     short aiDeviceNumber; // analog input device number
        short dioDeviceNumber; // digital I/O device number

        short overSamplingFactor;
35     short imageFormat;

        short          samptimebase;
        unsigned short sampint;
        unsigned long  sampCount; // total num. of samples per line, per channel

40     StagePos* stagePos;
        UserData* userData;

45     void LDaqConstants();

        virtual void PrepareBuffers();
        virtual void DeallocateBuffers();

50     virtual void CalculateParameters();

        virtual void InitializeInstrument();
        virtual void FinalizeInstrument();

55     virtual long LoopLines();

        virtual long TakeSingleLine(long linenum);
        virtual long SetupScan(long linenum);
60     virtual long DoScan(long linenum);
        virtual long DAQComplete(long linenum);
        virtual long LineScanComplete(long linenum);
        virtual void FinalizeLineScan(long linenum);
        virtual void PostProcessLineData(long linenum);

65     virtual long AcquirePixelsForLine(long linenum);
        virtual long GetPixel(long linenum, long rownum);

        virtual void SavePixel(long channel, long linenum,
70                             long rownum, short sample);

        virtual void WriteOutput();
        virtual void WriteImage(const short *image, const char *fname,
                                long lines, long cols, long pixelWidth);

```

```

75
    public:
        LDaq(StagePos* stagePos, UserData* theData);

80    virtual void initialize();

        virtual void ExecuteMeasurement();
        virtual void Clear();

85    virtual short isValid();

        virtual ~LDaq();
    };

90 #endif

```

Listing E.2: Class Definition ldaq.cpp.

```

#include <windows.h>
#include <stdlib.h>
#include <stdio.h>

5  #include "nidaq.h"
    #include "nidaqcns.h"
    #include "nidaqerr.h"

    #include "userdata.h"
10  #include "stagepos.h"
    #include "ldaq.h"

    /** LDaq, the Implementation
    *
15  * (C) 1997-2003 Guillaume Lessard
    * All rights reserved.
    */

    void LDaq::initialize()
20  {
        linelen = (userData->pixCount)*imageFormat;
        sampCount = linelen*overSamplingFactor;

        Set_DAQ_Device_Info(aiDeviceNumber,
25        ND_DATA_XFER_MODE_AI,
        ND_INTERRUPTS);

        stagePos->SetOrigin();
        stagePos->SetScanSizes(userData->amp, userData->pixCount);
30  stagePos->SetScanSpeed(userData->pixRate);

        stagePos->SetScanMode(StagePos::ScanRescan);
    }

35  // LDaq::ExecuteMeasurement()
    // This function performs all the necessary steps to take an
    // image with our setup.

```

```

void LDaq::ExecuteMeasurement()
40 {
    HCURSOR hNewCur, // new (hourglass) cursor
        hOldCur; // old (arrow) cursor

    // Change the cursor to the hourglass
45 hNewCur = LoadCursor(NULL, IDC_WAIT);
    hOldCur = SetCursor(hNewCur);

    // Allocate the buffers and fill them as appropriate
    PrepareBuffers();

50 // Calculate measurement and initialization parameters
    CalculateParameters();

    // Initialize daq board properly
55 InitializeInstrument();

    // Do the measurement!
    LoopLines();

60 WriteOutput();

    DeallocateBuffers();

    FinalizeInstrument();

65 // Change the cursor back to the previous value
    SetCursor(hOldCur);
}

70 // LDaq::WriteOutput()
    // Write out all the relevant data.

void
LDaq::WriteOutput()
75 {
    long c;
    char fname[30];

    userData->WriteOutput();

80 for(c=0; c<(userData->numChan); c++)
    {
        sprintf(fname, "%s%s%d%s",
            (userData->bname), "-", (userData->ch)[c], ".raw");
85 WriteImage(image[c], fname, userData->pixCount,
            imageFormat*userData->pixCount, sizeof(short));
    }
}

90 // LDaq::Clear
    // Clear the board. Useful at quit time.

void LDaq::Clear()
95 {
    DAQ_Clear(aiDeviceNumber);

```

```

    }

    // LDaq::CalculateParameters
100 // Calculate (and possibly negotiate) some of the parameters
    // necessary for the measurement.
    void LDaq::CalculateParameters()
    {
        volatile long err = 0;
105
        if(userData->numChan > 0)
        {
            DAQ_Rate((userData->pixRate)*overSamplingFactor*(userData->numChan),
                    0, &samptimebase, &sampint);
110
            // Set up the analog data acquisition sweep
            do
            {
                err = SCAN_Setup(aiDeviceNumber, (userData->numChan),
115                               (userData->ch), (userData->gain));
                if(err) throw err;

                err = SCAN_Start(aiDeviceNumber, sampleBuffer,
                                sampCount*(userData->numChan),
120                                samptimebase, sampint,
                                samptimebase, 0);
                if(err == rateNotSupportedError)
                {
                    sampint = (unsigned short)(sampint - 1);
125                }
                else if(err != 0)
                {
                    throw err;
                }
130            }
            while(err == rateNotSupportedError);
        }
        else
        {
135 // We at least need valid values for samptimebase and sampint...
            DAQ_Rate((userData->pixRate), 0, &samptimebase, &sampint);
        }
    }

140 // LDaq::InitializeInstrument
    // Initialize the data-acquisition part of the board.

    void LDaq::InitializeInstrument()
    {
145     volatile long err;

        DAQ_Clear(aiDeviceNumber);

        // Configure all channels: differential, ignored range, bipolar, ignored
150 err = AI_Configure(aiDeviceNumber, -1, 0, 20, 0, 1);
        if(err != 0)
            throw;

        DAQ_Config(aiDeviceNumber, 0, 0);

```

```

155      // now we select the input trigger signal as we really want it.
      // The input trigger will be the gating output of counter 0.
      err = Select_Signal(aiDeviceNumber, ND_IN_START_TRIGGER,
                          ND_GPCTR0_OUTPUT, ND_LOW_TO_HIGH);
160      if(err != 0) throw;

      stagePos->SetScanSpeed(userData->pixRate);
      if (userData->numChan > 0)
          stagePos->SetXAxisScanSyncToInput();
165      else
          stagePos->SetXAxisScanSyncToOutputTrigger();

      DIG_Line_Config(dioDeviceNumber, 0, 0, 1);
  }
170  void LDaq::FinalizeInstrument()
  {
  }

175  // LDaq::Looplines()
  // Take lines of data and increment the slow axis output

  long LDaq::LoopLines()
180  {
      long i, err;

      stagePos->RampToScanPosition(0,0);

185      for(i=0; i<(userData->pixCount); i++) // loop over lines of image
      {
          // Write out y axis value (slow axis sweep)
          stagePos->RampToScanPosition(0,i);

190          // Get the line of data
          err = TakeSingleLine(i);

          // Make sure we return to the scan line's origin...
          stagePos->RampToScanPosition(0,i);
195          if(err != 0)
              break;
      }

200      stagePos->RampToScanPosition(0,0);

      return err;
  }

205  long LDaq::TakeSingleLine(long linenum)
  {
      if (stagePos->GetScanMethod() == StagePos::LineScan)
      {
170          // Prepare the scanning and acquisition operation
          if(SetupScan(linenum) == 0)
          {

```



```

// TURN AOM ON
DIG_Out_Line(dioDeviceNumber, 0, 0, 1);
215
// Fire.
DoScan(linenum);

while (DAQComplete(linenum) == 0)
220 {
    Sleep(20);
}

// TURN AOM OFF
225 DIG_Out_Line(dioDeviceNumber, 0, 0, 0);

// Save the line data into the image array
PostProcessLineData(linenum);

230 // Wait for the scan to end!
while (LineScanComplete(linenum) == 0)
{
    Sleep(20);
}
235 }

// Do any cleanup
FinalizeLineScan(linenum);
}
240
return 0;
}

245 long LDaq::SetupScan(long linenum)
{
    if (userData->numChan > 0)
    {
        volatile long e1, e2;
250
        // Here, we will set up the asynchronous analog operations

        // Set up the analog data acquisition sweep
        do
255 {
            e1 = SCAN_Setup(aiDeviceNumber, (userData->numChan),
                           (userData->ch), (userData->gain));

            if (e1 != 0)
                return e1;
260

            e2 = SCAN_Start(aiDeviceNumber, sampleBuffer,
                           sampCount*(userData->numChan),
                           samptimebase, sampint, samptimebase, 0);

            if (e2 == rateNotSupportedError)
265 {
                sampint = (unsigned short)(sampint - 1);
            }
            else if (e2 != 0)
            {
270                return(e2);
            }
        }
    }
}

```

```

    }
    }
    while(e2 == rateNotSupportedError);
}
275     return stagePos->PrepareXAxisScan(linenum);
}

// LDaq::DoScan()
280 // Send some signal to actually start scanning
long LDaq::DoScan(long linenum)
{
    long e0 = 0, numPeriods = 1000;

285     e0 = GPCTR_Control(aiDeviceNumber, ND_COUNTER_0, ND_RESET);
    e0 = GPCTR_Set_Application(aiDeviceNumber,
                               ND_COUNTER_0, ND_SINGLE_PULSE_GNR);

    e0 = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_0,
290                               ND_COUNT_1, numPeriods);
    e0 = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_0,
                               ND_COUNT_2, numPeriods);

    e0 = Select_Signal(aiDeviceNumber, ND_GPCTR0_OUTPUT,
295                          ND_GPCTR0_OUTPUT, ND_LOW_TO_HIGH);

    e0 = GPCTR_Control(aiDeviceNumber, ND_COUNTER_0, ND_PROGRAM);

    return e0;
300 }

long LDaq::DAQComplete(long linenum)
{
305     short          daqStopped = 0;
    unsigned long    retr, err;

    if (userData->numChan > 0)
    {
310         err = DAQ_Check(aiDeviceNumber, &daqStopped, &retr);
        if (err)
            return err;

        return daqStopped;
315     }
    else
    {
        return 1;
    }
320 }

// Is the scan of the current line completely finished?
325 long LDaq::LineScanComplete(long linenum)
{
    return (DAQComplete(linenum) && stagePos->XAxisScanComplete());
}

```

```

330
    // Make sure to completely clear the scan data before moving on...

    void LDaq::FinalizeLineScan(long linenum)
335 {
    stagePos->Stop();

    DAQ_Clear(aiDeviceNumber);
    }

340

    // Save the data that we measured in the
    // proper place in the image matrices
345
    void LDaq::PostProcessLineData(long linenum)
    {
        if(userData->numChan > 0)
        {
350     long j, c, s, index, err, value;

        // Preliminary unspaghettization of the data.
        err = SCAN_Demux(sampleBuffer,
                        sampCount*(userData->numChan),
355                        userData->numChan, 0);

        // Copy data into image array
        // The data is spaghetti -- unjumbling it is important
        for(j=0; j<(imageFormat*(userData->pixCount)); j++)
360     {
        short newsamp = 0;
        for(c=0; c<(userData->numChan); c++)
        {
            value = 0;
365     for(s=0; s<overSamplingFactor; s++)
            {
                index = c*sampCount + j*overSamplingFactor + s;
                newsamp = (short)((sampleBuffer[index]*16)/overSamplingFactor);

370     if ((value + newsamp) > 32767L)
                    value = 32767L;
                else if ((value + newsamp) < -32767L)
                    value = -32767L;
                else
375     value += newsamp;
            }

            SavePixel(c, linenum, j, (short)value);
        }
    }
380 }
}
}

```

LDaq and its subclasses depend on two other object classes in order to obtain all the relevant information and capabilities to acquire data. The first of these is StagePos, which encapsulates

scanning and positioning operations. The StagePos class is not listed here. The second object class is UserData, a structure in which information about the input scan is stored. The header for UserData follows.

Listing E.3: Class Header userdata.h.

```

#ifndef __USERDATA__
#define __USERDATA__

#include "coordinate.h"
5
class UserData
{
public:
    double amp;           // scan range
10    long   pixCount;    // number of pixels in one line
    double pixRate;      // pixel capture rate

    short* ch;           // analog channel numbers
    short* gain;         // analog channel gains
15    short* polarity;   // analog channel polarities

    short   numChan;      // number of analog channels
    short   numDigital;   // number of digital channels

20    short   aspectRatio; // square or rectangular pixel area
    long     statPix;
    double   statRate;
    double   threshold;

25    double skewness;    // to accommodate non-rectangular coordinates
    double   stretch;    // for anisotropic coordinates
    Coordinate offset; // where to start the scan

    char     bname[64];

30    UserData(const UserData&);
    UserData();

    void WriteOutput();
35 };

#endif

```

E.1.1 Optical Scans

The acquisition of optical-only images by scanning the tip/tilt mirror is done by code from class DDaq6602. This code relies on the presence of both a PCI-6602 card and a PCI-6052E card for data acquisition.

Listing E.4: Class Header ddaq6602.h.

```

#ifndef __DDAQ6602__
#define __DDAQ6602__

#include "userdata.h"
5 #include "stagepos.h"
#include "ldaq.h"

/** Digital DAQ
*
10 * DDaq uses the on-board counters in order to acquire the optical
* data. For some reason, the E-series cards have a bug whereby they
* drop the occasional pixel (putting zero instead of putting some
* other value in it). In the old days, this got replaced by CDaq. In
* more modern days, it's back in action, using the 6602 card. It
15 * doesn't have the bug that the 6502E has, but it has other problems.
*
* (C) 1998-2003, Guillaume Lessard
* All rights reserved.
*/
20
class DDaq6602: public LDaq
{
protected:
    unsigned long *bogusBuffer;
25
    unsigned long *digitalBuffer6052;
    unsigned long *digitalBuffer6602;

    unsigned long *digitalImage6052;
30 unsigned long *digitalImage6602;

    unsigned long dcDeviceNumber;

    long numPeriods;
35
    virtual void InitializeInstrument();

    virtual void PrepareBuffers();
    virtual void DeallocateBuffers();
40
    virtual long SetupScan(long linenum);
    virtual long DoScan(long linenum);
    virtual long DAQComplete(long linenum);
    virtual void PostProcessLineData(long linenum);
45
    void SaveOutput();
    virtual void WriteOutput();

public:
50 DDaq6602(StagePos* stagePos, UserData* theData);

    virtual void initialize();
};

55 #endif

```

Listing E.5: Class Definition ddaq6602.cpp.

```

#include <math.h>
#include <stdlib.h>
#include <stdio.h>

5  #include "nidaq.h"
    #include "nidaqcns.h"
    #include "nidaqerr.h"

    #include "userdata.h"
10  #include "stagepos.h"

    #include "ldaq.h"
    #include "ddaq6602.h"

15  #include "errorbox.h"

    /** Code for the DDAQ6602 class
    *
    * Timing pulses are generated by the PCI6052E, and routed to a pin on
20  * the PCI6602. These edges are used as gates for counting photons per
    * "pixel". The first edge of each line is also used to trigger the
    * output scan, so that input and scanning are synchronized.
    *
    * Unfortunately, a problem with the PCI6602 has forced the use of the
25  * counters of the E-series card (PCI6052E): for some reason, the
    * event-counting mode cannot generate the value 0 (zero) as a result
    * on the PCI6602. It can be any other 32-bit value. *not*
    * zero. According to the documentation (or my understanding of it),
    * deactivating the "synchronous counting" mode should solve this. In
30  * reality, deactivating that mode does nothing. Another National
    * Instruments (TM) bug. Or maybe it was intended that way, in which
    * case it's merely an egregious design decision.
    *
    * Note that in the code below, calls that refer to aiDeviceNumber use
35  * the PCI6052E card, while calls that refer to dcDeviceNumber use the
    * PCI6602 card.
    *
    * (C) 1998-2003 Guillaume Lessard
    * All rights reserved.
40  */

void DDAQ6602::initialize()
{
    LDAQ::initialize();
45    volatile long err = 0;

    err = Set_DAQ_Device_Info(aiDeviceNumber,
                               ND_DATA_XFER_MODE_AI,
50    ND_INTERRUPTS);
}

void DDAQ6602::InitializeInstrument()
{
55    LDAQ::InitializeInstrument();

```

```

    long sampintmult = 100;

60    if(samptimebase == -1)
        sampintmult = 4L;
    else if(samptimebase == 1)
        sampintmult = 20L;
    else if(samptimebase == 2)
65        sampintmult = 200L;
    else if(samptimebase == 3)
        sampintmult = 2000L;
    else if(samptimebase == 4)
        sampintmult = 20000L;
70    else if(samptimebase == 5)
        sampintmult = 200000L;
    else
        sampintmult = 100L;

75    if(userData->numChan > 0)
        numPeriods = sampint*sampintmult*overSamplingFactor*(userData->numChan);
    else
        numPeriods = sampint*sampintmult;
}

80    long DDaq6602::SetupScan(long linenum)
    {
        volatile long e = 0;
        e = LDaq::SetupScan(linenum); //stagePos->PrepareXAxisScan(linenum);
85        if(e)
            return e;

        // Make sure it's really reset.

90        e = GPCTR_Control(aiDeviceNumber, ND_COUNTER_0, ND_RESET);
        e = GPCTR_Control(aiDeviceNumber, ND_COUNTER_1, ND_RESET);

        if (userData->numDigital > 1)
        {
95            e = GPCTR_Control(dcDeviceNumber, ND_COUNTER_4, ND_RESET);
            e = GPCTR_Control(dcDeviceNumber, ND_COUNTER_5, ND_RESET);
        }

        // Let's program the PCI6052E's counter 0 (what a mouthful)

100        e = GPCTR_Set_Application(aiDeviceNumber, ND_COUNTER_0,
                                   ND_PULSE_TRAIN_GNR);

        e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_0, ND_COUNT_1,
105                                   (long)floor(numPeriods*0.5));
        e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_0, ND_COUNT_2,
                                   (long)floor(numPeriods*0.5));

        e = Select_Signal(aiDeviceNumber, ND_GPCTR0_OUTPUT,
110                           ND_GPCTR0_OUTPUT, ND_LOW_TO_HIGH);

        if(e)
            return e;

```

```

115  // Let's program counter 1

    e = GPCTR_Set_Application(aiDeviceNumber, ND_COUNTER_1,
                             ND_BUFFERED_EVENT_CNT);

120  e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_1,
                             ND_SOURCE, ND_PFI_2);
    e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_1,
                             ND_SOURCE_POLARITY, ND_LOW_TO_HIGH);
    e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_1,
125  ND_GATE, ND_OTHER_GPCTR_OUTPUT);
    e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_1,
                             ND_GATE_POLARITY, ND_POSITIVE);
    e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_1,
                             ND_INITIAL_COUNT, 0);
130  e = GPCTR_Change_Parameter(aiDeviceNumber, ND_COUNTER_1,
                             ND_UP_DOWN, ND_COUNT_UP);

    e = GPCTR_Config_Buffer(aiDeviceNumber, ND_COUNTER_1, 0,
                             (imageFormat*(userData->pixCount)+1),
135  digitalBuffer6052);

    e = GPCTR_Control(aiDeviceNumber, ND_COUNTER_1, ND_PROGRAM);

    if (e)
140  return e;

    // Program Counters on card 6602...

145  if (userData->numDigital > 1)
    {
        // Program counter 4...

        e = GPCTR_Set_Application(dcDeviceNumber, ND_COUNTER_4,
150  ND_BUFFERED_EVENT_CNT);

        e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_4,
                                    ND_SOURCE, ND_INTERNAL_20_MHZ);

155  e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_4,
                                    ND_GATE, ND_PFI_22);
        e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_4,
                                    ND_GATE_POLARITY, ND_POSITIVE);
        e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_4,
160  ND_INITIAL_COUNT, 0);
        e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_4,
                                    ND_UP_DOWN, ND_COUNT_UP);

        e = GPCTR_Config_Buffer(dcDeviceNumber, ND_COUNTER_4, 0,
165  (imageFormat*(userData->pixCount)+1),
        bogusBuffer);

        e = GPCTR_Control(dcDeviceNumber, ND_COUNTER_4, ND_PROGRAM);

170  if (e)
        return e;

```



```

// Program counter 5...

175     e = GPCTR_Set_Application(dcDeviceNumber, ND_COUNTER_5,
                                ND_BUFFERED_EVENT_CNT);

    /*
    // If this line did what they say it's supposed to,
    // life would be immeasurably easier.
180     e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_5,
                                ND_COUNTING_SYNCHRONOUS, ND_NO);

    */
    e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_5,
                                ND_SOURCE, ND_OTHER_GPCTR_GATE);
185
    e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_5,
                                ND_GATE, ND_PFI_10);
    e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_5,
                                ND_GATE_POLARITY, ND_POSITIVE);
190     e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_5,
                                ND_INITIAL_COUNT, 0);
    e = GPCTR_Change_Parameter(dcDeviceNumber, ND_COUNTER_5,
                                ND_UP_DOWN, ND_COUNT_UP);

195     e = GPCTR_Config_Buffer(dcDeviceNumber, ND_COUNTER_5, 0,
                                (imageFormat*(userData->pixCount)+1),
                                digitalBuffer6602);

    e = GPCTR_Control(dcDeviceNumber, ND_COUNTER_5, ND_PROGRAM);
200
    if (e)
        return e;
    }

205     return 0;
    }

    long DDaq6602::DoScan(long linenum)
    {
210        // Fire the waveform generation on counter 0

        return GPCTR_Control(aiDeviceNumber, ND_COUNTER_0, ND_PROGRAM);

        // There is no need to call LDaq::DoScan(), as all it does is a subset of
215        // DDaq6602::SetupScan() and DDaq6602::DoScan().
    }

    long DDaq6602::DAQComplete(long linenum)
    {
220        if(LDaq::DAQComplete(linenum) == 0)
            return 0;

        unsigned long counting;
        volatile long e, i;
225

        // Stop the counters when counter 1 is not armed anymore
        e = GPCTR_Watch(aiDeviceNumber, ND_COUNTER_1, ND_ARMED, &counting);
        if ((counting == ND_NO) && (userData->numDigital > 1))
        {
230            e = GPCTR_Watch(dcDeviceNumber, ND_COUNTER_5, ND_ARMED, &counting);

```

```

        if (counting != ND_NO)
        {
            for (i=0; i<100; i++)
235         {
                Sleep(20);
                e = GPCTR_Watch(dcDeviceNumber, ND_COUNTER_5, ND_ARMED, &counting);
                if ((counting == ND_NO) || (e != 0))
                    break;
240         }

            return 1;
        }
    }
245 }

    return ((counting == ND_NO) || (e != 0));
}

void DDaq6602::PostProcessLineData(long linenum)
250 {
    LDaq::PostProcessLineData(linenum);

    long j, index;
    long value = 0;
255

    // Copy data into image array
    // The data is spaghetti -- unjumbling might be important!!
    for(j=0; j<(imageFormat*(userData->pixCount)); j++)
    {
260         index = linenum*imageFormat*(userData->pixCount) + j;

        value = digitalBuffer6052[j+1] - digitalBuffer6052[j];
        digitalImage6052[index] = (long)(value);

265         if (userData->numDigital > 1)
        {
            value = digitalBuffer6602[j+1] - digitalBuffer6602[j];
            digitalImage6602[index] = (long)(value);
        }
270     }

    volatile long e;
    e = GPCTR_Control(aiDeviceNumber, ND_COUNTER_0, ND_RESET);
    e = GPCTR_Control(aiDeviceNumber, ND_COUNTER_1, ND_RESET);
275

    if (userData->numDigital > 1)
    {
        e = GPCTR_Control(dcDeviceNumber, ND_COUNTER_4, ND_RESET);
        e = GPCTR_Control(dcDeviceNumber, ND_COUNTER_5, ND_RESET);
280     }
}

```

E.2 FANSOM Data Acquisition

SyncDaq is the class that implements the NanoScope™-synchronized FANSOM. Unlike the other classes presented in this appendix, this one saves its data asynchronously, during data acquisition. This approach was taken since the size of the data files is essentially random, and precise timing is unknown to the driver software prior to the start of data acquisition.

Listing E.6: Class Header syncdaq.h.

```

#ifndef __SYNCDAQ__
#define __SYNCDAQ__

#include "userdata.h"
5 #include "stagepos.h"

#include "ldaq.h"
#include "mdaq.h"

10 /** SyncDaq
    *
    * SyncDaq implements an externally-synchronized acquisition of
    * timestamps for one, two or three channels. The synchronization is
    * done by starting and stopping the clock against which the
15 * timestamps are measured. A "long enough" pause causes the data
    * acquisition code to initiate cleanup so that a next line of data
    * acquisition can be initiated before the clock restarts.
    *
    * (C) 2002,2003 Guillaume Lessard
20 * All rights reserved.
    */

#define NI_PCI_6602 232
#define NI_PCI_6052E 273
25
class SyncDaq: public LDaq
{
    long bufferbins;

30 char localPrefix[32], localSuffix[32];

protected:
    unsigned long** localBuf;

35 unsigned long* copyBuf;

    unsigned long counter[3], source[3], gate[3];
    unsigned long counternum[3], saved[3];
    HANDLE file[3];
40 long stop[3];

    short usingPCI6602;
    short countersUsed;

45 short dcDeviceNumber;

```

```

    virtual void InitializeInstrument();
    virtual void PrepareBuffers();
    virtual void DeallocateBuffers();
50     virtual long LoopLines();

    virtual long SetupScan(long linenum);
    virtual long DoScan(long linenum);
55     virtual long DAQComplete(long linenum);
    virtual long CopyNewBufferData(unsigned long counter, HANDLE file,
                                   unsigned long min, unsigned long& saved);
    virtual void FinalizeLineScan(long linenum);
60     virtual void GetFilenamePrefix(char* prefix, long linenum);
    virtual void GetFilenameSuffix(char* suffix);

public:
65     SyncDaq(StagePos* stagePos, UserData* theData);
};

#endif

```

Listing E.7: Class Definition syncdaq.cpp.

```

#include <math.h>
#include <stdlib.h>
#include <stdio.h>

5  #include <windows.h>

#include "nidaq.h"
#include "nidaqcns.h"
#include "nidaqerr.h"
10 #include "userdata.h"
#include "stagepos.h"

#include "ldaq.h"
15 #include "syncdaq.h"

#include "errorbox.h"

/** SyncDaq, featuring external synchronization.
20  *
   * (C) 2002,2003 Guillaume Lessard
   * All rights reserved.
   */

25 SyncDaq::SyncDaq(StagePos* stagePos, UserData* userData)
    : LDaq(stagePos, userData)
    {
        localBuf = NULL;
        copyBuf = NULL;
30     bufferbins = (1024*1024);
    }

```

```

    dcDeviceNumber = 2;

35    usingPCI6602 = 1;
    countersUsed = userData->numDigital;

    counternum[0] = 2;
    counternum[1] = 0;
40    counternum[2] = 4;

    counter[0] = ND_COUNTER_2;
    counter[1] = ND_COUNTER_0;
    counter[2] = ND_COUNTER_4;
45

    source[0] = ND_PFI_39;
    source[1] = ND_PFI_39;
    source[2] = ND_PFI_39;

50    gate[0] = ND_PFI_30;
    gate[1] = ND_PFI_38;
    gate[2] = ND_PFI_22;

    for (int i=0; i<countersUsed; i++)
55    {
        stop[i] = 1;
    }
}

60
void SyncDaq::InitializeInstrument()
{
    LDaq::InitializeInstrument();

65    Set_DAQ_Device_Info(dcDeviceNumber,
                        ND_DATA_XFER_MODE_GPCTR0,
                        ND_UP_TO_1_DMA_CHANNEL);

    if (usingPCI6602)
70    {
        Set_DAQ_Device_Info(dcDeviceNumber,
                            ND_DATA_XFER_MODE_GPCTR2,
                            ND_UP_TO_1_DMA_CHANNEL);

75        Set_DAQ_Device_Info(dcDeviceNumber,
                            ND_DATA_XFER_MODE_GPCTR4,
                            ND_UP_TO_1_DMA_CHANNEL);
    }
    else
80    {
        Set_DAQ_Device_Info(dcDeviceNumber,
                            ND_DATA_XFER_MODE_GPCTR1,
                            ND_UP_TO_1_DMA_CHANNEL);
    }

85    // This is the "Reset" signal, to make sure the laser is going to be on
    // during the next frame...

    DIG_Line_Config(dioDeviceNumber, 0, 1, 1);
90    DIG_Out_Line(dioDeviceNumber, 0, 1, 1);

```

```

    }

    long SyncDaq::LoopLines()
    {
95      // Set the trigger for the out scan to the frame-start signal
        stagePos->SetXAxisScanSyncToSignal(ND_PFI_4, ND_LOW_TO_HIGH);

        // Reset frame
        DIG_Out_Line(dioDeviceNumber, 0, 1, 0);
100      Sleep(100);
        DIG_Out_Line(dioDeviceNumber, 0, 1, 1);

        return LDaq::LoopLines();
    }
105
    long SyncDaq::SetupScan(long linenum)
    {
        long err = LDaq::SetupScan(linenum);
        if(err)
110          return err;

        GetFilenamePrefix(localPrefix, linenum);
        GetFilenameSuffix(localSuffix);

115      volatile long e = 0;

        // Make sure it's really reset.

        for (int i=0; i<countersUsed; i++)
120      {
            e = GPCTR_Control(dcDeviceNumber, counter[i], ND_RESET);
            stop[i] = 0;
            saved[i] = (unsigned long)-1;
        }
125      // Program counters

        for (int i=0; i<countersUsed; i++)
        {
130          volatile unsigned long theGate, theSource, theCounter;
            theGate = gate[i];
            theSource = source[i];
            theCounter = counter[i];

135          if (stop[i] == 0)
            {
                e = GPCTR_Set_Application(dcDeviceNumber, counter[i],
                                           ND_BUFFERED_EVENT_CNT);

140                e = GPCTR_Change_Parameter(dcDeviceNumber, counter[i],
                                           ND_SOURCE, source[i]);

                e = GPCTR_Change_Parameter(dcDeviceNumber, counter[i],
                                           ND_GATE, gate[i]);
145                e = GPCTR_Change_Parameter(dcDeviceNumber, counter[i],
                                           ND_GATE_POLARITY, ND_POSITIVE);

                e = GPCTR_Config_Buffer(dcDeviceNumber, counter[i],

```

```

                                0, bufferbins+1, localBuf[i]);
150
        if (usingPCI6602)
        {
            e = GPCTR_Change_Parameter(dcDeviceNumber, counter[i],
                                ND_COUNTING_SYNCHRONOUS, ND_NO);
155
            e = Select_Signal(dcDeviceNumber, ND_START_TRIGGER,
                                ND_PFI_39, ND_LOW_TO_HIGH);

            e = GPCTR_Change_Parameter(dcDeviceNumber, counter[i],
160                                ND_START_TRIGGER, ND_ENABLED);
        }

        e = GPCTR_Change_Parameter(dcDeviceNumber, counter[i],
                                ND_BUFFER_MODE, ND_CONTINUOUS);
165

        if(e)
            return e;
    }
}
170
return e;
}

long SyncDaq::DoScan(long linenum)
175 {
    volatile int e = 0;
    for (int i=0; i<countersUsed; i++)
    {
        e = GPCTR_Control(dcDeviceNumber, counter[i], ND_PROGRAM);
180
        if (e) return e;
    }

    char fname[64];
185
    for (int i=0; i<countersUsed; i++)
    {
        if (stop[i] == 0)
        {
190            sprintf(fname, "%s%d%s", localPrefix, counternum[i], localSuffix);
            file[i] = CreateFile(fname, GENERIC_WRITE, FILE_SHARE_READ, NULL,
                                CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
        }
    }
195
    return e;
}

long SyncDaq::DAQComplete(long linenum)
200 {
    volatile long finished = LDaq::DAQComplete(linenum);
    volatile long countersDone = 0;

    long min[3] = {100000, 10000, 100000};
205    countersDone = 1;

```

```

    for (int i=0; i<countersUsed; i++)
    {
        if (stop[i] == 0)
210    {
            stop[i] = CopyNewBufferData(counter[i], file[i], min[i], saved[i]);
            countersDone *= stop[i];
        }
    }
215
    return ((finished != 0) && (countersDone != 0));
}

long SyncDaq::CopyNewBufferData(unsigned long counter, HANDLE file,
220                                unsigned long min, unsigned long& saved)
{
    unsigned long readcounts = 0, savedcounts = 0;

    volatile long e = 0;
225
    e = GPCTR_Watch(dcDeviceNumber, counter, ND_READ_MARK, &readcounts);
    e = GPCTR_Watch(dcDeviceNumber, counter, ND_WRITE_MARK, &savedcounts);

    if ((savedcounts > (readcounts + min)) || (savedcounts == saved))
230    {
        unsigned long counts = 0;

        e = GPCTR_Read_Buffer(dcDeviceNumber, counter,
                                ND_READ_MARK, 0,
235                                (savedcounts - readcounts), 0,
                                &counts, copyBuf);

        if (e == overWriteError)
            return e;
240

        if (e && (e != gpctrDataLossError))
            return e;

        if ((counts > 0)&&(copyBuf[counts-1] > 0))
245    {
            DWORD written;
            BOOL okay;
            okay = WriteFile(file, copyBuf, counts*sizeof(long), &written, NULL);

250            // When the clock (ND_SOURCE) has stopped ticking, we stop
            // the operation... We detect stoppage if at each gate for a while,
            // the counter has been at the same value).

            // PCI-6052E case
255            if (copyBuf[outcounts-1] == copyBuf[0])
            {
                GPCTR_Control(dcDeviceNumber, counter, ND_RESET);
                return 1;
            }
260        }
        else if ((e == gpctrDataLossError)&&(readcounts != 0))
        {
            GPCTR_Control(dcDeviceNumber, counter, ND_RESET);
            return 1;
        }
    }
}

```



```

265     }
    }
    else if ((savedcounts == 0) && (readcounts == 0) && (!usingPCI6602))
    {
        unsigned long counts = 0;
270     e = GPCTR_Watch(dcDeviceNumber, counter, ND_COUNT, &counts);

        if (counts != 0)
        {
            Sleep(5);
275
            e = GPCTR_Watch(dcDeviceNumber, counter, ND_WRITE_MARK, &savedcounts);
            return (savedcounts == 0);
        }
    }
280 else
    {
        saved = savedcounts;
    }

285 return 0;
}

void SyncDaq::FinalizeLineScan(long linenum)
{
290     LDaq::FinalizeLineScan(linenum);

    if (linenum == 0)
    {
        // Set the trigger for the out scan to the line-start signal
295     stagePos->SetXAxisScanSyncToSignal(ND_PFI_6, ND_LOW_TO_HIGH);
    }

    for (int i=0; i<countersUsed; i++)
    {
300     GPCTR_Control(dcDeviceNumber, counter[i], ND_RESET);
        CloseHandle(file[i]);
    }
}

```

Bibliography

- [1] Anthoni van Leeuwenhoek. Letter to the Royal Society of London. Published in the Philosophical Transactions of the Royal Society, September 1683.
- [2] E.H. Synge. A suggested method for extending microscopic resolution into the ultra-microscopic region. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6, 7th series:356–362, 1928.
- [3] H. H. Pattee. The scanning x-ray microscope. *Journal of the Optical Society Of America*, 43(1):61–62, 1953.
- [4] John Aloysius O’Keefe. Resolving power of visible light. *Journal of the Optical Society of America*, 46(5):359, 1956.
- [5] A. V. Baez. Is resolving power independent of wavelength possible—an experiment with a sonic macroscope. *Journal of the Optical Society of America*, 46(10):901, 1956.
- [6] E. A. Ash and G. Nicholls. Super-resolution aperture scanning microscope. *Nature*, 237:510–512, 1972.
- [7] Dieter W. Pohl. Optical near-field scanning microscope. European Patent 0112401 (filed December 27, 1982; published July 4, 1984); U.S. Patent 4,604,520, December 20, 1983.
- [8] A. Lewis, M. Isaacson, A. Muray, and A. Harootunian. Scanning optical spectral microscopy with 500 Å spatial-resolution. *Biophysical Journal*, 41(2):A405, 1983.
- [9] D.W. Pohl, W. Denk, and M. Lanz. Optical stethoscopy—image recording with resolution $\lambda/20$. *Applied Physics Letters*, 44(7):651–653, 1984.
- [10] A. Lewis, M. Isaacson, A. Harootunian, and A. Muray. Development of a 500 Å spatial-resolution light-microscope .1. light is efficiently transmitted through gamma-16 diameter apertures. *Ultramicroscopy*, 13(3):227–231, 1984.
- [11] B. Hecht, H. Bielefeldt, Y. Inouye, D. W. Pohl, and L. Novotny. Facts and artifacts in near-field optical microscopy. *Journal of Applied Physics*, 81(6):2492–2498, 1997.

- [12] H. K. Wickramasinghe and C. C. Williams. Apertureless near field optical microscope. U.S. Patent 4,947,034, 28 April 1989.
- [13] F. Zenhausern, M. P. O'Boyle, and H. K. Wickramasinghe. Apertureless near-field optical microscope. *Applied Physics Letters*, 65(13):1623–1625, 1994.
- [14] M. A. Paesler and P. J. Moyer. *Near Field Optics: Theory, Instrumentation and Applications*. Wiley Interscience, New York, 1996.
- [15] F. Zenhausern, Y. Martin, and H. K. Wickramasinghe. Scanning interferometric apertureless microscopy—optical imaging at 10 angstrom resolution. *Science*, 269(5227):1083–1085, 1995.
- [16] Y. Martin, S. Rishton, and H. K. Wickramasinghe. Optical data storage read out at 256 Gbits/in². *Applied Physics Letters*, 71(1):1–3, 1997.
- [17] R. Bachelot, P. Gleyzes, and A. C. Boccara. Near-field optical microscopy by local perturbation of a diffraction spot. *Microscopy Microanalysis Microstructures*, 5(4–6):389–397, 1994.
- [18] P. Gleyzes, A. C. Boccara, and R. Bachelot. Near-field optical microscopy using a metallic vibrating tip. *Ultramicroscopy*, 57(2–3):318–322, 1995.
- [19] A. Lahrech, R. Bachelot, P. Gleyzes, and A. C. Boccara. Infrared-reflection-mode near-field microscopy using an apertureless probe with a resolution of $\lambda/600$. *Optics Letters*, 21(17):1315–1317, 1996.
- [20] Y. Inouye and S. Kawata. Near-field scanning optical microscope with a metallic probe tip. *Optics Letters*, 19(3):159–161, 1994.
- [21] S. Kawata and Y. Inouye. Scanning probe optical microscopy using a metallic probe tip. *Ultramicroscopy*, 57(2–3):313–317, 1995.
- [22] Bernard Valeur. *Molecular Fluorescence: Principles and Applications*. Wiley–VCH, 2002.
- [23] G. G. Stokes. On the change of the refrangibility of light. *Philosophical Transactions*, 142:463–562, 1852.
- [24] S. M. Nie, D. T. Chiu, and R. N. Zare. Probing individual molecules with confocal fluorescence microscopy. *Science*, 266(5187):1018–1021, 1994.
- [25] X. S. Xie and R. C. Dunn. Probing single-molecule dynamics. *Science*, 265(5170):361–364, 1994.
- [26] M. Unger, E. Kartalov, C. S. Chiu, H. A. Lester, and S. R. Quake. Single-molecule fluorescence observed with mercury lamp illumination. *Biotechniques*, 27(5):1008–1014, 1999.

- [27] E. J. Sanchez, L. Novotny, and X. S. Xie. Near-field fluorescence microscopy based on two-photon excitation with metal tips. *Physical Review Letters*, 82(20):4014–4017, 1999.
- [28] N. Hayazawa, Y. Inouye, and S. Kawata. Evanescent field excitation and measurement of dye fluorescence in a metallic probe near-field scanning optical microscope. *Journal of Microscopy–Oxford*, 194(2-3):472–476, 1999.
- [29] H. F. Hamann, A. Gallagher, and D. J. Nesbitt. Near-field fluorescence imaging by localized field enhancement near a sharp probe tip. *Applied Physics Letters*, 76(14):1953–1955, 2000.
- [30] H. F. Hamann, M. Kuno, A. Gallagher, and D. J. Nesbitt. Molecular fluorescence in the vicinity of a nanoscopic probe. *Journal of Chemical Physics*, 114(19):8596–8609, 2001.
- [31] Virgil B. Elings and John A. Gurley. Tapping atomic force microscope. U. S. Patent 5,412,980, August 7, 1992.
- [32] Q. Zhong, D. Inniss, K. Kjoller, and V. B. Elings. Fractured polymer silica fiber surface studied by tapping mode atomic-force microscopy. *Surface Science*, 290(1–2):L688–L692, 1993.
- [33] N. A. Burnham, O. P. Behrend, F. Oulevey, G. Gremaud, P. J. Gallo, D. Gourdon, E. Dupas, A. J. Kulik, H. M. Pollock, and G. A. D. Briggs. How does a tip tap? *Nanotechnology*, 8(2):67–75, 1997.
- [34] Guillaume A. Lessard, T. J. Yang, Pierre Barritault, and Stephen R. Quake. A scanning apertureless fluorescence microscope. In Eiichi Tamiya and Shuming Nie, editors, *Scanning and force microscopies for biomedical applications*, volume 3607 of *Proceedings of the SPIE*, pages 158–165, 1999.
- [35] T. J. Yang, Guillaume A. Lessard, and Stephen R. Quake. An apertureless near-field microscope for fluorescence imaging. *Applied Physics Letters*, 76(3):378–380, January 2000.
- [36] Private Communication from Veeco/DI Customer Support, July 2002.
- [37] Alberto Leon-Garcia. *Probability and Random processes for Electrical Engineering*. Addison Wesley, second edition, 1994.
- [38] M. Kuno, D. P. Fromm, H. F. Hamann, A. Gallagher, and D. J. Nesbitt. "On"/"off" fluorescence intermittency of single semiconductor quantum dots. *Journal of Chemical Physics*, 115(2):1028–1040, 2001.
- [39] Anthony E. Siegman. *Lasers*. University Science Books, Mill Valley, California, 1986.
- [40] S. O. Rice. Mathematical analysis of random noise. *Bell Systems Technical Journal* 23:282–332, 1944; 24:46–156, 1945.

- [41] Athanasios Papoulis. *Signal Analysis*. McGraw Hill, 1977.
- [42] B. Sick, B. Hecht, and L. Novotny. Orientational imaging of single molecules by annular illumination. *Physical Review Letters*, 85(21):4482–4485, 2000.
- [43] B. Sick, B. Hecht, U. P. Wild, and L. Novotny. Probing confined fields with single molecules and vice versa. *Journal of Microscopy–Oxford*, 202(2):365–373, 2001.
- [44] K. H. Drexhage. Interaction of light with monomolecular dye layers. In Emil Wolf, editor, *Progress in Optics*, volume 12, pages 165–232. North-Holland Publishing Company, Amsterdam, 1974.
- [45] R. R. Chance, A. Prock, and R. Silbey. Molecular fluorescence and energy transfer near interfaces. In I. Prigogine and Stuart A. Rice, editors, *Advances in Chemical Physics*, volume 37, pages 1–66. Wiley Interscience, 1978.
- [46] H. Gersen, M. F. Garcia-Parajo, L. Novotny, J. A. Veerman, L. Kuipers, and N. F. van Hulst. Near-field effects in single molecule emission. *Journal of Microscopy–Oxford*, 202(2):374–378, 2001.
- [47] H. Gersen, M. F. Garcia-Parajo, L. Novotny, J. A. Veerman, L. Kuipers, and N. F. van Hulst. Influencing the angular emission of a single molecule. *Physical Review Letters*, 85(25):5312–5315, 2000.
- [48] John L. Bohn, D. J. Nesbitt, and A. Gallagher. Field enhancement in apertureless near-field scanning optical microscopy. *Journal of the Optical Society of America A*, 18(12):2998–3006, December 2001.
- [49] Y. C. Martin, H. F. Hamann, and H. K. Wickramasinghe. Strength of the electric field in apertureless near-field optical microscopy. *Journal of Applied Physics*, 89(10):5774–5778, 2001.
- [50] Werner Kern and David A. Puotinen. Cleaning solutions based on hydrogen peroxide for use in silicon semiconductor technology. *RCA Review*, 31:187–206, June 1970.

This document was prepared in $\text{\LaTeX}2_{\epsilon}$, using the class file `new_cit_thesis`, written by Daniel M. Zimmerman and made available on the Caltech Library System website. It was typeset directly to Adobe Portable Document Format using `pdfTeX`, included in version 2.0 of Thomas Esser's `teTeX` distribution. The figures were prepared using `MATLAB`[®]. Most schematic diagrams were drawn using Adobe[®] `Illustrator`[®] 10; the others were drawn using `OmniGraffle Professional 3.0`, by the Omni Group.

The typeface used for the main text is `Palatino`[™], designed Herman Zapf. Typewriter-style text is set using the `Courier` typeface by Howard Kettler. Mathematical symbols and formulas are typeset using the Pazo mathematical font by Diego Puga, including some borrowed glyphs from Donald E. Knuth's Computer Modern family of typefaces. Some greek characters from Adobe Symbol are interspersed throughout.