Templated Self-Assembly of sub-10 nm Quantum Dots

by

Joshua C. Leu

B.S., Physics and Electrical Engineering
Stanford University, 2005

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE

IN PARTIAL FULFILLMENT OF THE REQIUREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[June 2008]
MAY 2008

Signature of Author.....
Department of Electrical Engineering and Computer Science
May 23, 2008

Certified by.....
Karl K. Berggren
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by.....
Vladimir Bulovic
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by.....
Terry Orlando
Chairman, Department Committee on Graduate Students

Templated Self-Assembly of sub-10 nm Quantum Dots

by

Joshua C. Leu

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2008, in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Electrical Engineering and Computer Science

ABSTRACT

Patterned templates can guide the self-assembly of nanoparticles into ordered arrays. Our motivation in pursuing templated self-assembly is to develop a robust method for the creation of ordered structures at length scales below ten nanometers. The basic process entails creating surface relief templates via electron-beam lithography, and spin-coating a suspension of colloidal nanoparticles onto the template. As the solvent evaporates, the quantum dots self-assemble primarily through the capillary forces created by the dewetting of the template.

We demonstrate this technique at sub-10nm length scales by spin-coating a solution of organically-capped CdZnS semiconducting quantum dots onto nanopatterned grating structures on silicon substrates. We observe the geometric confinement of the quantum dots via physical templating and capillary forces into well-ordered monolayer aggregates with defined lattice orientations. While recent research has demonstrated the ability to self-assemble sub-10nm metallic nanoparticles via capillary forces into physical templates of similar size, this work is unique in the demonstration of lattice orientation control via physical templating at sub-10nm length scales.

Thesis Supervisor: Karl. K. Berggren
Title: Associate Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Vladimir Bulovic
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

The research presented in this thesis would not have been possible without the dedication, advice, guidance, and patience of my thesis advisors, Professor Karl K. Berggren and Professor Vladimir Bulovic. I am thankful to them for creating a supportive and intellectually exciting environment. They truly embody the highest virtues of both teacher and mentor, and for that I am grateful.

Particular thanks also go to Bryan Cord and Polina Anikeeva, who have both been integral to this research. I am thankful to Bryan Cord for his advice in nanofabrication, his wit and humor, as well as the samples he provided for this research. I am thankful to Polina Anikeeva for her advice in quantum dot preparation and processing, for her assistance in quantum dot calibration and spin-coating, and for her enthusiasm and professionalism in research. They are the best lab-mates I could have possibly hoped for.

Additional thanks go to Hassen Abdu and Jonathan Halpert, for their contributions to this research. Hassen Abdu provided valuable assistance in the quantum dot memory simulations, and Jonathan Halpert synthesized the quantum dots used in this research.

There are many others who have contributed to this project, or to my development as a research, to whom I am indebted.

To Professor Kathryn A. Moler, for introducing me to academic research.

To Professor Henry Smith and Professor Terry Orlando, for the advice that they have given me.

To Dr. Delia Milliron, for her advice and mentorship in quantum dot synthesis and functionalization.

To Mark Mondol, for the use of the Raith 150, for the invaluable advice and technical knowledge, and for an infectious enthusiasm towards research.

To Jim Daley, for the administration and maintenance of the NanoStructures Laboratory.

To Vikas Anant, Eric Dauler, and Joel Yang, for advice, friendship, and interesting discussions that have inspired this work.

To my group mates, both past and present, in both the Laboratory of Organic Optics and Electronics and the Quantum Nanostructures and Nanofabrication Group, for an intellectually interesting workplace.

To staff, members, and users of the NanoStructures Laboratory, for their advice, and for making the NanoStructures Laboratory an enjoyable environment for research.

To IBM Corporation, for generously providing the funding for this research.

And finally, to my family in both Taiwan and the United States, for their constant support, understanding, and faith in me.

Table of Contents

Chapter 1

Introduction

This thesis work focuses on a method for the controlled self-assembly of nanoparticles. The motivation for this line of research was to create arbitrary structures that either have length scales too small or pattern densities too high to achieve with traditional lithographic methods, such as electron-beam lithography or optical projection lithography. This research focused on templated quantum dot (QD) self-assembly, for the creation of controlled nanostructures in the sub-10 nm regime. We used electron-beam lithography to create nanopatterned grating structures to template the self-assembly of sub-10 nm nanoparticles into lattice structures of controlled orientation. Additionally, using high-resolution scanning-electron-beam microscopy, we quantitatively characterized the resulting self-assembled nanoparticle aggregates, as a first step towards the systematic engineering of templated self-assembled structures.

Templated Self-Assembly

Templated self-assembly is an approach to extending the fabrication capabilities of existing techniques to smaller length scales and higher densities [1]. First, a template is created by a method which allows the placement of arbitrary patterns. Because the template pattern is often directly controlled by the lithographer, it is often referred to as a "top-down" patterning

technique. This template provides the initial conditions that control the subsequent self-assembly. The self-assembled component then settles into a low energy configuration induced deliberately through the constraints of the designed template. Since self-assembly cannot be directly controlled, but instead relies on spontaneous processes, it is often referred to as a "bottom-up" patterning technique. In summary, templated self-assembly involves the creation of a template which contains enough information to guide the self-assembled element into a set of desired spatial states.

Templated self-assembly presents two main technical challenges. First, the top-down lithographic technique must create surface features of sufficiently small dimensions with sufficient quality to interact in a controlled way with the self-assembled element. As we will discuss in this thesis, since the template guides the process of self-assembly, the quality of the self-assembled structures depends critically on the quality of the template. Secondly, the self-assembly mechanism must be robust enough to fill lithographically defined features of varying shapes and dimensions in a repeatable manner. The reliability of the patterning depends on the ease with which the self-assembled element can be configured into the desired low-energy state. Often, the self-assembled component settles into multiple localized energy minimums, creating grain boundaries and defects. The elimination of these defects, and the creation of large, controlled, ordered regions is a continuing challenge for templated self-assembly.

Templated self-assembly seeks to combine of the capabilities of conventional nanolithography in creating large-scale, arbitrary features, with the molecular-level control of self-assembly. These two technologies, if properly integrated, promise a level of resolution and control far beyond what either technology could achieve alone.

Previous Work in Colloidal Particle Self-Assembly and Templated Self-Assembly

Colloidal nanocrystals of various types, including quantum dots, have been demonstrated to self-assemble into large 2-D [2] and 3-D crystals [3] through careful control of nanocrystal monodispersity. In particular, extremely large opaline structures have been formed from the physical confinement of spherical colloidal particles under continuous sonication [4] or through the attractive capillary forces between particles in the colloidal suspension [5-6]. Using polystyrene beads ranging from 200 nm to 500 nm, these opaline structures can achieve perfectly-ordered domain sizes in excess of several square centimeters [7]. These methods of self-assembly have been used with varying degrees of success on colloidal spheres with diameters as small as 35 nm [8]. The particular class of sub-10 nm quantum dots used in this research has already been shown to self-assembly ordered monolayers by spin-coating [9], and micron-sized domain areas can be formed via phase-separated spin-casting [10]. Controlled dewetting of drop-cast colloidal nanoparticles solutions can also yield large ordered monolayers

over areas of several square microns [11].

Efforts in templated self-assembly of colloidal microspheres and nanospheres have included physical templating [12-13], and chemical templating [14-16], or other means [2, 17]. Several previous works have particular relevance to this research. Through the controlled dewetting of monodispersed colloidal suspensions in fluidic cells, silica and polystyrene spheres as small as 150 nm in diameter have been templated in well-ordered configurations into surface relief structures by capillary forces [18]. Recently, gold nanoparticles as small as 2nm have been self-assembled into pores by capillary forces [19].

Templated Self-Assembly of sub-10nm Colloidal Quantum Dots

In this work, the templates were fabricated by electron-beam lithography (EBL), using a poly(methyl methacrylate) (PMMA) positive electron-beam resist to create a resist profile, which was then transferred by the evaporation of metal and then the lift-off of the sacrificial resist layer. The final result was the transfer of the aerial dose pattern to a metal template. For this research, the templates consisted of gold with adhesion layers of titanium. On these templates, we spin-coated a solution of approximately 8nm diameter ZnCdS alloyed nanoparticles solvated in chloroform. The nanoparticles self-assemble primarily through the capillary forces created by the dewetting of the template as the solvent evaporates. Figure 1.1 shows the geometric confinement

of the quantum dots via physical templating and capillary forces into well-ordered monolayer aggregates with orientations defined by the template. This work demonstrates templated self-assembly using semiconducting quantum dots via spin-coating, and also demonstrates control of the position and alignment of two-dimensional lattices of close-packed nanoparticles in the sub-10 nm length scale.

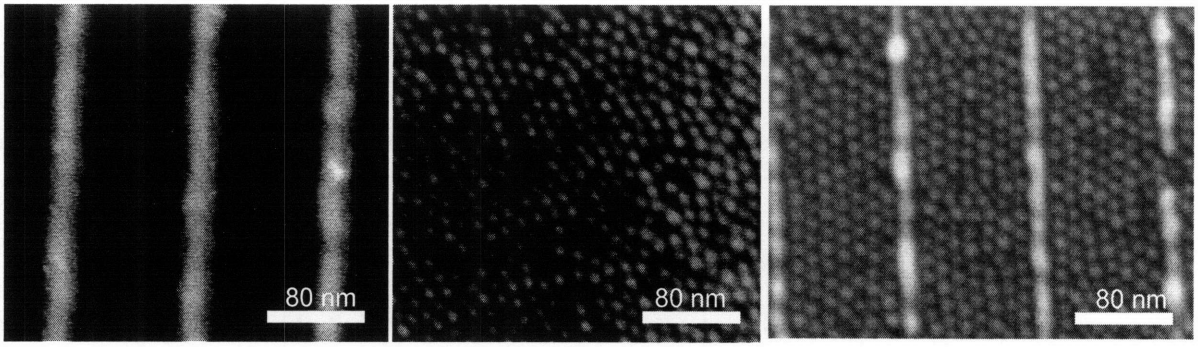

Figure 1.1 (Left) A scanning electron microscope (SEM) image of a grating pattern made of Au with a Ti adhesion layer on a silicon substrate. The textured surface visible between the grating lines results from the Au/Pd layer sputtered on the sample to enhance imaging. (Center) A SEM image of 8 nm-diameter CdZnS quantum dots, which were spin-coated onto a silicon substrate with native oxide. (Right) A SEM image of 8 nm-diameter CdZnS quantum dots spin-coated onto a grating template. The grating pattern in both the left and right images were fabricated by identical processes, while the quantum dots imaged in the center and right images were from the same colloidal solution. Scale bars for all images are 80 nm.

Applications for Templated QD Self-Assembly

Colloidal semiconducting quantum dots are a particularly useful class of materials, because of their size-dependent electrical and optical properties [19]. Because of the diversity of II-VI and III-V material systems used in quantum dots, a wide range of material properties can be selected.

Additionally, the quantum dots can be functionalized with an array of organic capping layers, to facilitate solubility on organic solvents. Due to these properties, colloidal QDs have already found applications in various solution-processed optoelectronic devices, including photovoltaic cells [20], light-emitting devices [9], transistors [21], memory devices [22], and photodetectors [23], where a thin-film layer of colloidal QDs are integrated into the device stack. In such devices, the controlled deposition and morphology of the QD layer is often critical to device performance. A templated, self-assembled monolayer of QDs could improve device performance and reliability.

Additionally, a single semiconducting QD could be incorporated into a single-electron transistor, and photoactive QDs could be incorporated into hybrid silicon-QD optoelectronic devices [24]. Templated QD self-assembly could also be leveraged for nanopatterning and nanolithography. The QD pattern could be transferred into a host material, creating small periodic structures beyond the lithographic capabilities of electron-beam or optical projection lithography [25]. These nanostructured materials could be used for a variety of applications, such as higher charge mobility for transistor channels, or higher charge storage densities for memory cells. Periodic structures in the sub-10 nm regime may permit direct manipulation of the electronic bandgap structure of the host material, creating new possibilities for silicon-based optoelectronic devices.

Summary of Thesis Work

I demonstrate the templated self-assembly of colloidal quantum dots at sub-10nm length scales by spin-coating a solution of organically capped semiconducting QDs onto nanopatterned grating structures on silicon substrates, and observe the geometric confinement of the quantum dots via physical templating and capillary forces into well-ordered monolayer aggregates with defined orientations. This work involved three distinct tasks.

1. The fabrication of the templates by electron-beam lithography and metal lift-off, and developing techniques for the spin-coating the colloidal QD solution.
2. Development of SEM imaging techniques for the challenge of imaging sub-10 nm QDs, which are highly susceptible to charging.
3. Development of numerical analytical tools to characterize the order and orientation of the QD aggregates, to quantitatively assess the effects of the template on QD self-assembly.

[1] J. Y. Cheng, A. M. Mayes, and C. A. Ross, *Nature Materials* 3, 823 (2004)

[2] E. Kumacheva, R. K. Golding, M. Allard, and E. H. Sargent, *Adv. Mater.* 14, 221 (2002)

[3] F. X. Redl, K. S. Cho, C. B. Murray, and S. O'Brien, *Nature* 423, 968 (2003)

[4] S. H. Park, D. Qin, and Y. X. Xia, *Adv. Mater.* 10, 1028 (1998)

[5] G. S. Lazarov, N. D. Denkov, O.D. Velev, P. A. Kralchevsky, and K. Nagayama, *J. Chem. Soc. Faraday Trans.* 90, 2007 (1994)

[6] P. Jiang, J. F. Bertone, K. S. Hwang, and V. L. Colvin, *Chem. Mater.* 11, 2132 (1999)

[7] B. Gate, D. Qin, and Y. Xia, *Adv. Mater.* 11, 446 (1999)

[8] S. A. Johnson, P. J. Ollivier, and T. E. Mallouk, *Science* 283, 963 (1999)

[9] S.A. Coe, W.-K. Woo, M. G. Bawendi, and V. Bulovic, *Nature* 420, 800 (2002)

[10] S. A. Coe-Sullivan, J. S. Steckel, W.-K. Woo, M. G. Bawendi, and V. Bulovic, *Adv. Funct. Mat.* 15, 1117 (2005)

[11] X. M. Lin, H. M. Jaeger, C. M. Sorensen, and K. J. Kalbunde, *J. Phys. Chem. B* 105, 3353 (2001)

[12] Y. Xia, Y. Yin, Y. Lu, and J. McLellan, *Adv. Func. Mater.* 13, 907 (2003)

[13] J. A. Liddle, Y. Cui, and A.P. Alivsatos, *J. Vac. Sci. Technol.* 22, 3409 (2004)

[14] H. Zhang, M. F. Rubner, and P. T. Hammond, *Langmuir* 18, 4505 (2002)

[15] M. Himmelhaus, and H. Takei, *Phys. Chem. Chem. Phys.* 4, 496 (2002)

[16] J. Aizenberg, A. J. Black, and G. M. Whitesides, Nature 398, 495 (1999)

[17] J. Lin, W. Zhou, A. Kumbhar, J. Wiemann, J. Fang, E. E. Carpenter, and C. J. O'Connor, *J. Solid State Chem.* 159, 26 (2001)

[18] Y. Yin, Y. Lu, B. Gates, and Y. Xia, *J. Am. Chem. Soc.* 123, 8718 (2001)

[19] C. B. Murray, C. Kagan, M. G. Bawendi, *Annu. Rev. Mat. Sci.* 30, 545 (2000)

[20] W. U. Huynh, J.J. Dittmer, and A. P. Alivisatos, *Science* 295, 2425 (2002)

[21] D. V. Talapin, C. B. Murray, *Science* 310, 86 (2005)

[22] M. Geller, E. Stock, C. Kapteyn, R. L. Sellin, and D. Bimberg, *Phys. Rev. B* 73, 205331 (2006)

[23] G. Konstantatos, I. Howard, A. Fisher, S. Hoogland, J. Clifford, E. Klem, L. Levina, and E. H. Sargent, *Nature* 442, 180 (2006)

[24] M. Hegg, and L. Y. Lin, *Optics Express* 15, 17163 (2007)

[25] R. Tabone, Master of Science Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2005.

Chapter 2

Scanning Electron Microscopy of Quantum Dots

Scanning electron microscopy (SEM) was the primary tool used for the characterization of templated quantum dot self-assemblies studied in this thesis. Quantum dots (QDs) have generally been imaged using transmission electron microscopy and atomic force microscopy, and scanning electron microscopy has not been widely used, particularly for characterizing colloidal semiconducting quantum dots. High resolution imaging of quantum dots by SEM is difficult due to the small size of QDs, and their susceptibility to charging during imaging. However, these difficulties can be mitigated through the optimization of instrument parameters and the use of imaging techniques detailed in this chapter.

The main tool used for this research was the Raith 150 scanning electron microscope, in the Scanning Electron Beam Lithography (SEBL) facility.

Principles of Scanning Electron Microscopy

The scanning electron microscope is a type of electron microscope that uses a focused beam of accelerated electrons to interact with the sample [1]. There are several different interactions between the electron beam and the sample, and these interactions can be selectively recorded by various types of detectors. As the electron beam is moved along the surface of the sample in a raster pattern, the time-varying detector signal is used to build an image of sample.

Scanning electron microscopy is a very versatile form of microscopy, due to its mode of operation. SEMs are capable of creating images with great depth of focus at a wide range of magnifications, with fields-of-view up to one millimeter [2-3]. This makes the SEM ideal for

imaging samples with high degrees of topographical variation, such as templated QD self-assemblies.

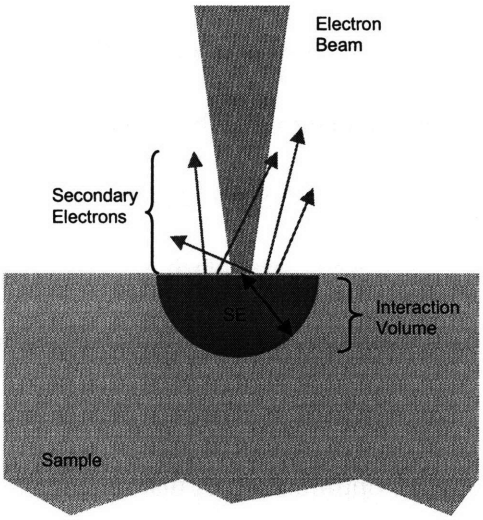

Figure 2.1 An illustration of the generation of secondary electrons in the sample by the electron beam. For a 20 keV electron beam, the typical interaction radii for secondary electrons is 5-10 nm. Also illustrated are a few sample trajectories of secondary electrons emitted from the sample surface.

The most commonly used method of scanning-electron microscopy uses secondary electrons, generated through the ionization of atoms in the sample material by the electron beam. As illustrated in figure 2.1, these secondary electrons, which are generated within a small radius around the primary beam, typically provide the best resolution for any given sample, because their low energy (< 50 eV) limits their distance of travel in the sample material. This short travel distance ensures that the collected secondary electron signal can be attributed to the smallest possible volume of the sample. The size of this "interaction volume" between the sample and the beam depends on beam energy and beam diameter, as well as the material properties, but resolutions of 2~5 nm are typically possible [2]. Secondary electron imaging was the primary method used in inspecting the self-assembled QD samples.

Figure 2.2 shows an overview schematic of the LEO GEMINI electron-beam column [4], which is used in both in the Raith 150 of the SEBL facility and the LEO SEM of the

NanoStructures Laboratory. This electron-beam column is a typical architecture for a scanning electron microscope, although the GEMINI column does contain a few enhancements, such as an in-lens secondary electron detector, as well as other features which enhance its performance but do not affect its operating principles.[1]

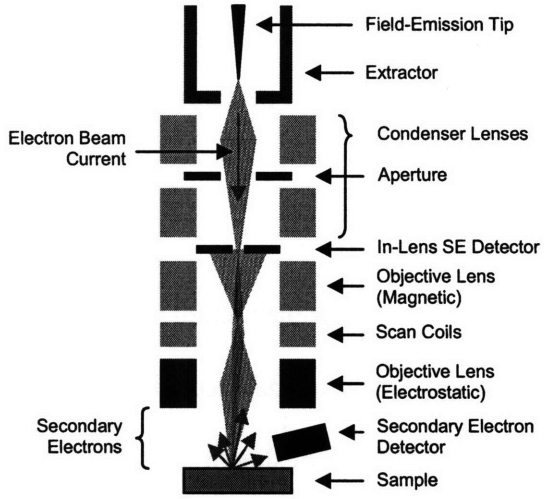

Figure 2.2 A schematic depiction of the LEO GEMINI column found in the Raith 150 SEM system. The electron source is a field-emission tip, which creates a beam current (shown in red), that is first focused and apertured by the condenser lens assembly. The beam is further focused by the objective lenses, and directed towards the sample by scan coils. The secondary electrons (shown in green) emitted by the sample can be either collected by the in-lens detector, or by an off-axis secondary electron detector. In this schematic, the yellow components are magnetic lenses; the blue components are electrostatic elements.

Beginning from the top of the electron-beam column, the extractor assembly pulls a highly monochromatic and bright beam of electrons from the Schottky field-emission source. This beam is focused through the condenser lenses, and an aperture controls the effective size of the source and the beam current for any particular acceleration voltage. The electron beam is directed by the scan coils and focused onto the sample by the objective lenses. The secondary electrons generated from the sample can be collected by either an off-axis secondary-electron detector, or they can be collected through the electrostatic objective lens and accelerated back to

[1] For example, the beam booster, which accelerates the electron beam to very high voltages regardless of the final acceleration voltage – the electrons are decelerated by the electrostatic objective lens after they have passed through the deflection coils. This high acceleration voltage gives reduced sensitivity to magnetic field interference and higher source brightness at low beam voltages.

an annular secondary electron detector. Since this detector is integrated into the lens assembly, situated on the beam axis, it is also known as an in-lens detector. Because the in-lens detector only collects secondary electrons in the direct vicinity of the electron beam, it has a higher signal-to-noise ratio than the off-axis secondary electron detector, which also collects background secondary electrons, which can be generated by backscattered electrons. However, for large features, the off-axis secondary electron detector is better suited to capture topography.

SEM Imaging of Quantum Dots

Quantum dots are difficult to image because of their small size and susceptibility to charging. Charging occurs when the sample area being imaged acquires an electric charge, distorting the image. Semiconducting quantum dots have been shown to create quantum confinement of charge carriers within the quantum dot, behaving as quantum potential wells [4]. Additionally, due to the poor surface conductivity of the capping layer, organically-capped quantum dots cannot easily dissipate the electric charge created during imaging, and are particularly prone to charging. Large uniform layers of quantum dots are particularly hard to image, because most of the image contrast is at the QD spatial frequency. Because the 8-nm-diameter CdZnS quantum dots used in these experiments approached the resolution limit of the imaging tool, even small changes in beam focusing resulted in total loss of contrast. However, despite these difficulties, these 8-nm-diameter quantum dots were imaged through the technique detailed below.

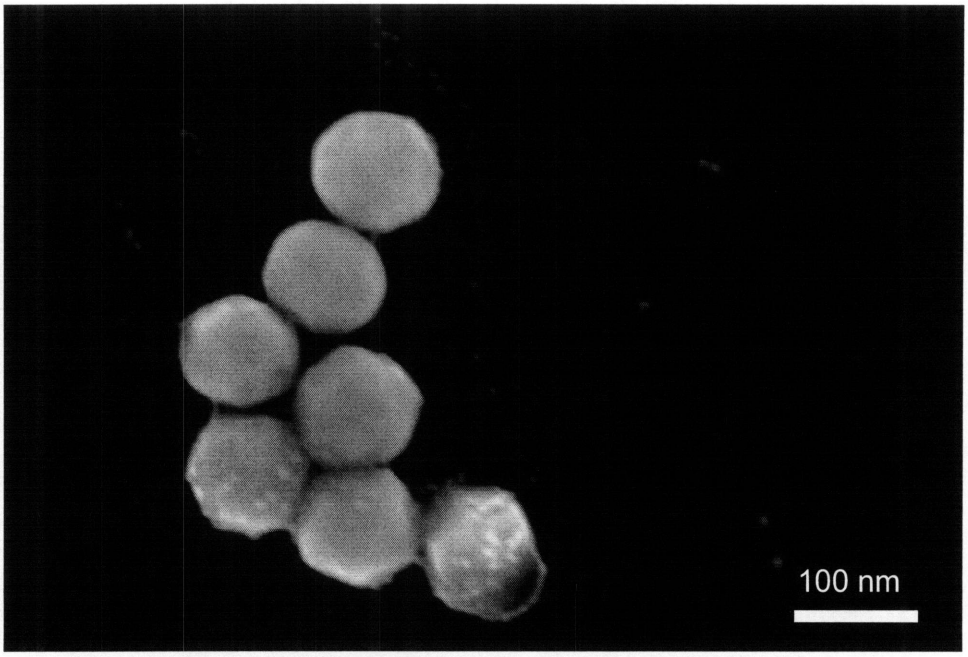

Figure 2.3 An SEM image of 100-nm-diameter Au spheres and 8-nm-diameter CdZnS quantum dots, taken using the in-lens detector on the Raith 150, with an aperture of 30 microns and an acceleration voltage of 10 keV, and a working distance of 6 mm.

Since the QD size was close to the resolution limit of the Raith 150, care was taken to maximize the signal-to-noise ratio of the image by adjusting the column parameters. Because the Raith 150 is limited by spherical aberration [5], reducing the aperture size would also reduce the beam diameter; however, reducing the aperture would also reduce the beam current and the total signal collected by the in-lens detector. After testing a range of aperture sizes from 20 microns to 150 microns, we found that a 30 micron aperture yielded the best results. Similarly, we tested a range of acceleration voltages from 5 keV to 15 keV, finding that a 10 keV acceleration voltage yielded the best results. Increasing the acceleration voltage would decrease the local secondary electron yield, decreasing total signal, but also alleviating sample charging and increasing imaging averaging time. Figure 2.3 shows a typical SEM image taken under these column parameters. Notice that the 8 nm CdZnS quantum dots are clearly resolved even when adjacent,

and that those areas with small radii of curvature, such as the 8 nm QDs and the facet edges of the 100 nm Au nanoparticles, appeared as brighter parts of the image due to larger numbers of secondary electrons that escaped from the interaction volume.

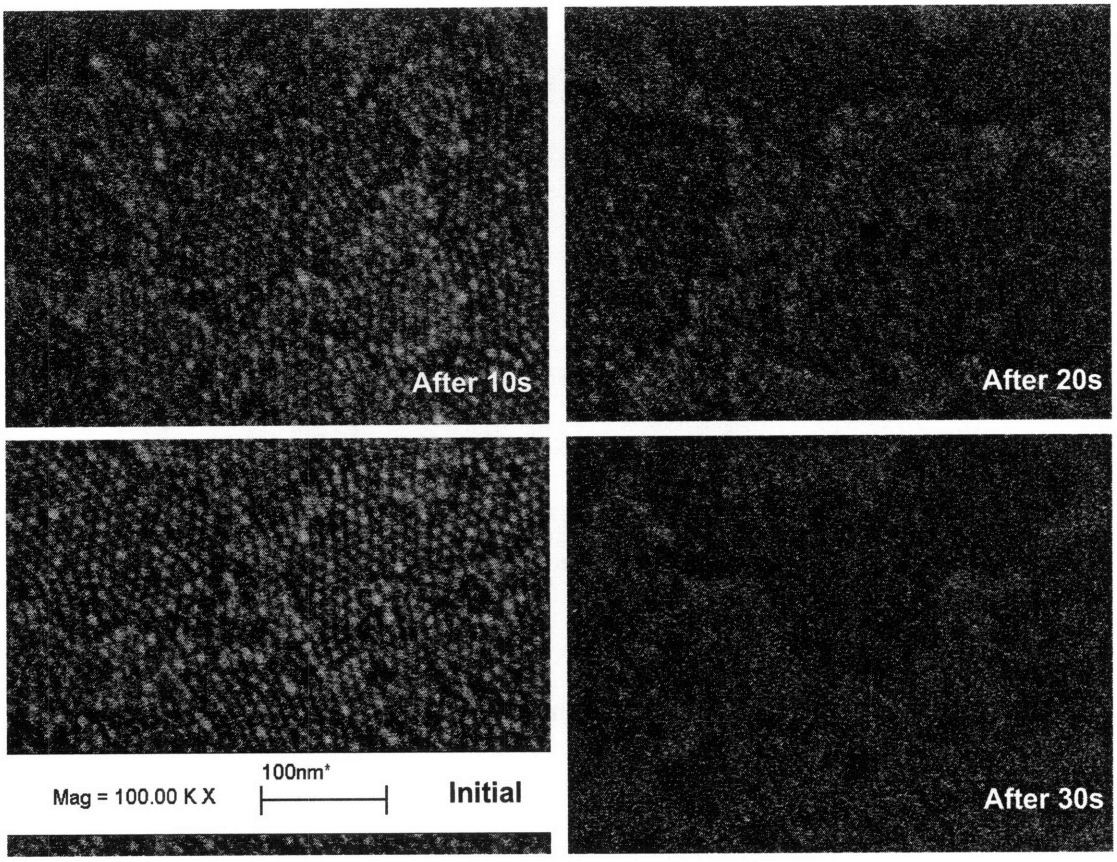

Figure 2.4 Successive images of a 8 nm CdZnS quantum dot multilayer showing the image contrast degradation due to charging. After 30 seconds of continuous imaging, the individual quantum dots can no longer be resolved.

The second difficulty in imaging quantum dots is their easy susceptibility to charging. As mentioned earlier, the poor surface conductivity of the organically-capped quantum dots, as well as their electrical behavior as quantum wells, prevented charge accumulated during imaging from efficiently dissipating into the conducting silicon substrate. At high magnifications, when the electron beam is scanning over a small sample area, charging can cause total loss of contrast within 30 seconds. As shown in figure 2.4, even at moderate magnifications, the loss of contrast

due to charging was severe. Additionally, sputtering an Au/Pd imaging layer on top of the QDs did not improve image contrast, but worsened it.

Since we could not reduce charging by introducing conducting layers adjacent to the quantum dots, long averaging times for SEM images were not possible. In general, longer image averaging times are desirable to suppress shot noise, thus increasing the signal-to-noise ratio. However, due to charging, the image contrast fell within seconds, quickly negating any gain in image quality due to suppression of shot noise. We found that the optimal imaging time at high resolutions similar to figure 2.5 was roughly four seconds, if using the column parameters previously laid forth. The images of QDs shown in figure 2.5 represent the highest resolution and highest image quality achieved using the Raith 150 for imaging quantum dots.

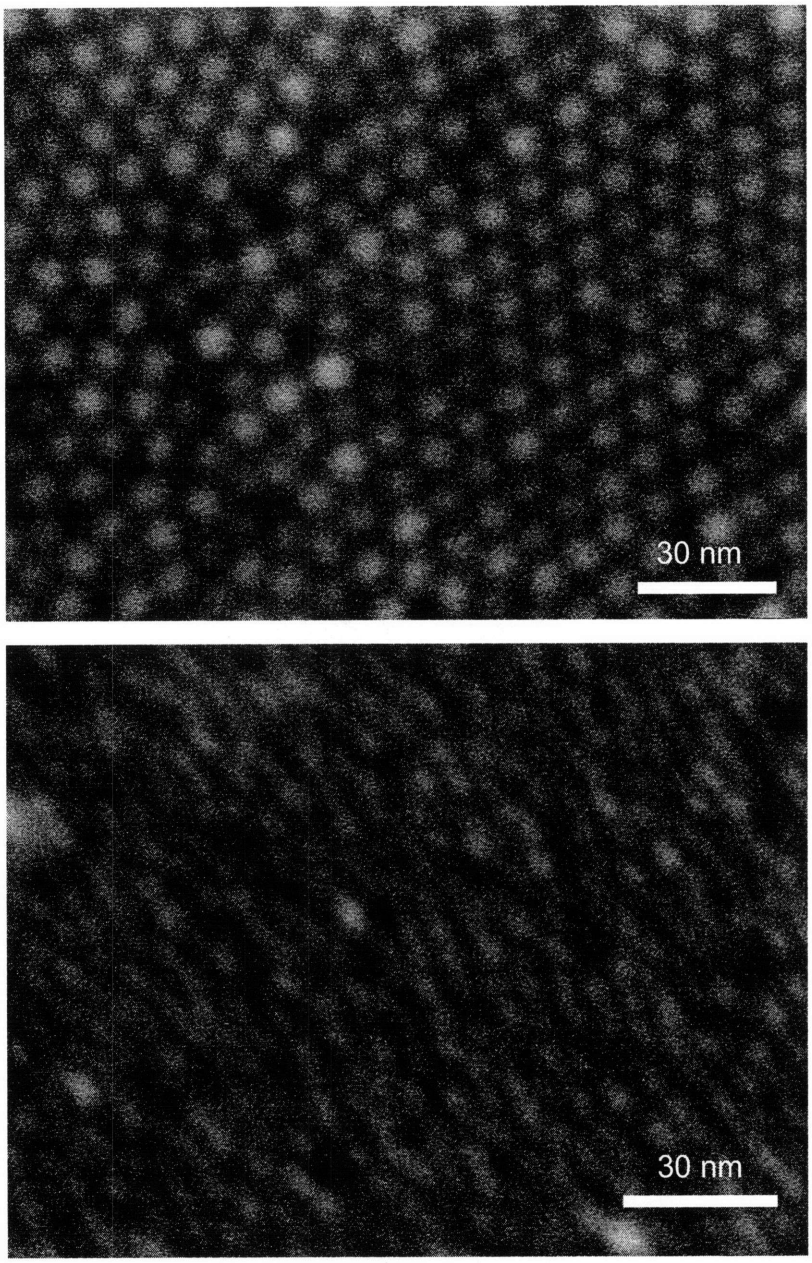

Figure 2.5 (Top) An 8 nm CdZnS quantum dot monolayer. (Bottom) 5 nm CdZnS quantum dots. Like all images in this chapter, these images were taken using the Raith 150 in-lens detector, with a 30 micron aperture, a 10 keV acceleration voltage, and a 6 mm working distance.

[1] Reimer, Ludwig. *Scanning Electron Microscopy: Physics of Image Formation and Microanalysis.* Berlin: Springer-Verlag, 1998.

[2] Joseph I. Goldstein ... [et al.]. *Scanning Electron Microscopy and X-ray Microanalysis.* New York : Kluwer Academic/Plenum Publishers, 2003.

[3] Martin Prutton, Mohamed M. El Gomati. Ed. *Scanning Auger Electron Microscopy* Chichester, England ; Hoboken, NJ : Wiley, 2006.

[4] A. Mews, A. V. Kadavanich, U. Banin, and A. P. Alivisatos, *Phys. Rev. B.* 53(20), 13242 (1996)

[5] Carl Zeiss AG. *GEMINI field-emission electron optics: core technology for the LEO SUPRA range*, 2003.

Chapter 3

Fabrication of Templated Quantum Dot Self-Assemblies

As stated in the introductory chapter, templated self-assembly presents two main technical challenges. The "top-down" lithographic technique must create high-fidelity surface features with sufficiently small dimensions to interact in a controlled way with the self-assembled element. This template provides the information from which a desired low-energy state can be defined. Then, "bottom-up" self-assembled element must be robust enough to settle into that low-energy state in a repeatable manner.

This chapter describes the material systems and processes used to create the templated QD self-assemblies described in this thesis. The templates consisted of lithographically defined gold wires on a silicon substrate with native oxide. The gold wires were approximately 10 nm wide, with heights ranging from 20 to 60 nm. Additionally, silicon trenches 50 to 90 nm wide were also used as templates. The self-assembled element consisted of colloidal 8-nm-diameter CdZnS quantum dots, organically capped with oleic acid.

Template Fabrication

Both the gold-nanowire and silicon-trench templates were patterned using Raith 150 electron-beam lithography tool in the SEBL facility, and sample processing was done using the facilities provided by the NanoStructures Laboratory (NSL). This fabrication work was done in collaboration with Bryan Cord, who was responsible for the development of the techniques described in this section, and performed the majority of the template fabrication work used for this research.

The intent of the gratings was to create a wide range of templating conditions for the subsequent quantum-dot self-assembly, so we designed gratings with varying orientation, grating linewidth and line spacing. We designed gratings oriented at right angles relative to each other, with each grating covering a 4 micron by 4 micron area. Each pair of orthogonally-oriented gratings was repeated twice, creating the basic patterned for each linewidth and line spacing. This pattern was produced for linewidths ranging from 8 nm to 30 nm, and for line spacings from 30 to 100 nm. This entire array of gratings was then repeated at different doses, ranging from 600 $\mu C/cm^2$ to 2000 $\mu C/cm^2$.

The gold nanowire templates were fabricated by a metal lift-off process, shown schematically in figure 3.1. First a thin layer of poly(methyl methacrylate) or PMMA was spin-coated onto a silicon substrate with native oxide. A 2.2 Mdalton PMMA manufactured by Microchem was used as the electron-beam resist, and spin-coated layer was either 80 or 160 nm thick, depending on the desired amount of gold evaporation. A 160-nm-thick PMMA layer was used for depositing 80-thick gold layers, and 80-nm-thick PMMA layer was used for gold deposition thicknesses below 80 nm.

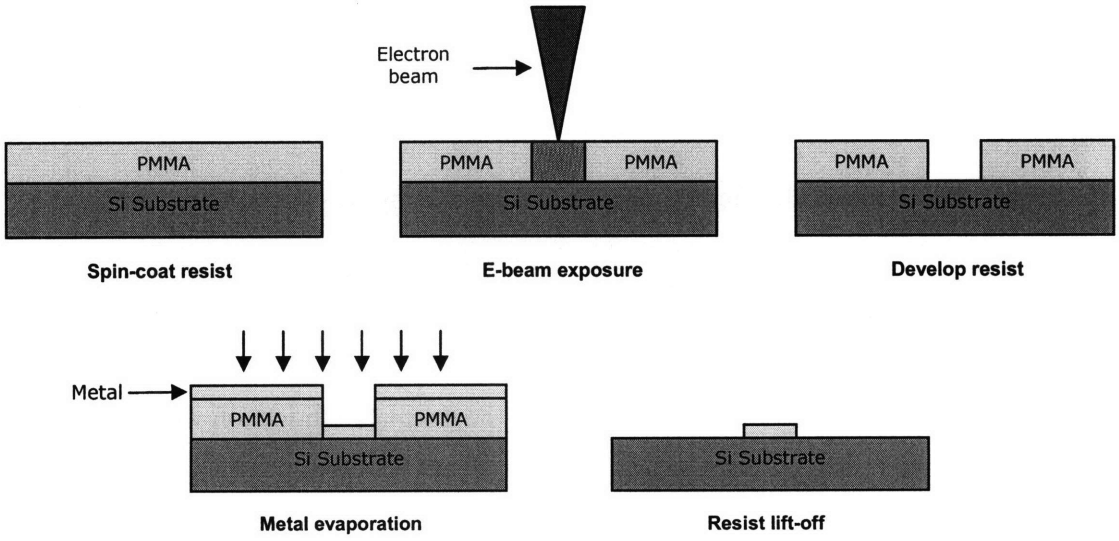

Figure 3.1 The process flow for electron-beam lithography, using metal lift-off for pattern transfer.

After spin-coating the PMMA resist, the samples were exposed via electron-beam lithography. Similar to scanning-electron microscopy, a tightly focused beam of accelerated electrons are directed at specific areas of the PMMA resist, exposing the resist. The exposed area is indicated in figure 3.1 as the purple region in the resist layer. The Raith 150, which was previously described in chapter 2, was also used for electron-beam lithography, using a 30 keV acceleration voltage and a 30 micron aperture, with a resulting beam current of between 0.21~0.23 nA. Because a wide range of linewidths and line spacings were used in each pattern, the differing levels of area dose caused by back-scattering meant that each different parts of the patterns had different optimal dosages. A series of dosages, from 600 $\mu C/cm^2$ to 2000 $\mu C/cm^2$, were used for each template pattern to ensure the accurate transfer of the intended pattern onto the resist. As the accelerated electrons pass through the resist, they generate secondary electrons which causes chain-scission of the polymers, reducing their molecular weight in the exposed regions [1].

This difference in molecular weight is exploited during resist development. The exposed PMMA resist layer was developed by a solution of 3:1 isopropanol:methylisobutylketone (IPA:MIBK) at 20 °C for 30 seconds. The exposed regions, with short polymer chains, are readily dissolved by the developer, while the unexposed regions, whose polymer chains remain intact, do not dissolve easily and remain fixed to the substrate [1]. This difference creates a negative image of the exposure pattern in the resist.

To transfer the pattern from the resist, metals were deposited onto the substrate by evaporation. First, a nominal thickness of 10 nm of Ti was evaporated onto the samples to serve as an adhesion layer for the gold. Then, a nominal thickness of 20, 40, or 80 nm of gold was deposited onto the samples.

After metal deposition, the unexposed PMMA was removed via lift-off. The lift-off solvent was n-methylpyrrolidone (NMP), which was heated to between 90 and 100 C. The NMP completely dissolves the PMMA, and any metal not attached to the substrate is removed from the substrate. The result is a gold template pattern with a Ti adhesion layer on a silicon substrate. A typical template result is shown in figure 3.2.

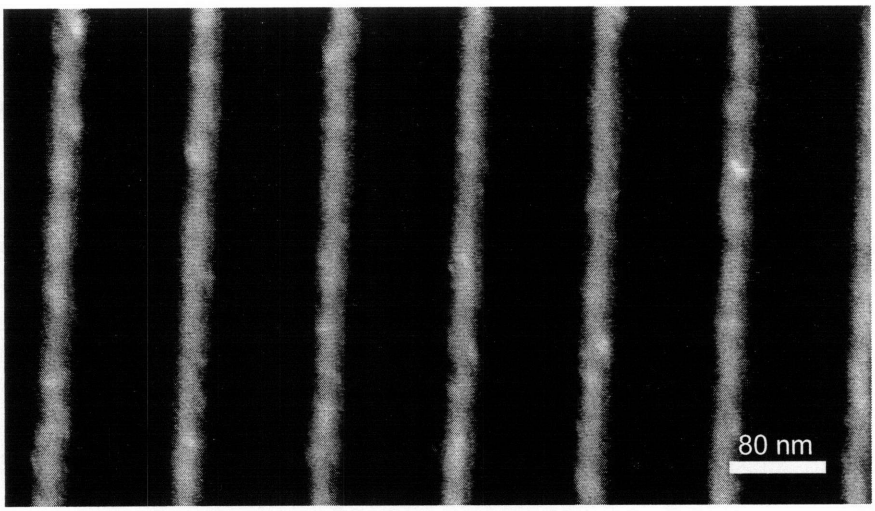

Figure 3.2 A scanning electron microscope (SEM) image of a grating pattern made of Au with a Ti adhesion layer on a silicon substrate. The textured surface visible between the grating lines results from the Au/Pd layer sputtered on the sample to enhance imaging.

However, while the aerial view of the gold nanowires closely matches the intended pattern, a side-profile reveals the widely-varying line height of the template grating, shown in figure 3.3. Such variation is often found when attempting to fabricate small-scale high-aspect ratio features by metal lift-off. This variation is caused by a variety of factors, such as variation in the aerial dose, resist feature deformation or collapse during development, self-shadowing during metal-evaporation, and metal-film tearing during lift-off. Future research will seek to address these processing issues.

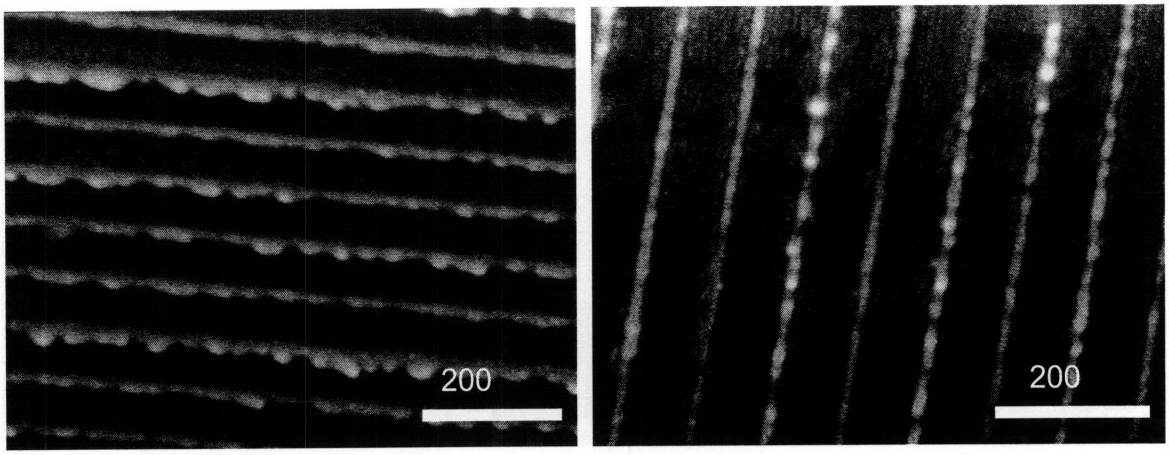

Figure 3.3 Side-profile SEM images of a 10-nm-wide grating lines, with 80 nm line spacing. There is significant variation in the height of the grating lines.

The fabrication of the silicon trench templates was done by a different process flow, schematically outlined in figure 3.4. First, 30 nm of SiO_X was evaporated onto the silicon substrate prior to the spin-coating of 85 nm of PMMA. The resist layer was exposed and developed in the same manner as the gold nanowire templates, with the only difference being a change in the development temperature. Instead of developing at room temperature, the solution of 3:1 IPA:MIBK was first cooled to -15 °C, and the samples were developed in the cooled solution for 30 seconds.

Using the resist profile as an etch mask, the SiO_X layer was reactive-ion etched by 10 mtorr of CF_4 at a fixed sample-voltage bias of 100 V, with a corresponding RF power of 130 W. The surviving resist was cleaned off the substrate by oxygen plasma. Then, using the etched SiO_X layer as a etch mask, the underlying silicon substrate was etched, again by RIE, using 10 mtorr of Cl_2, at a fixed RF power of 90W, and a corresponding sample-voltage bias of 425 V.

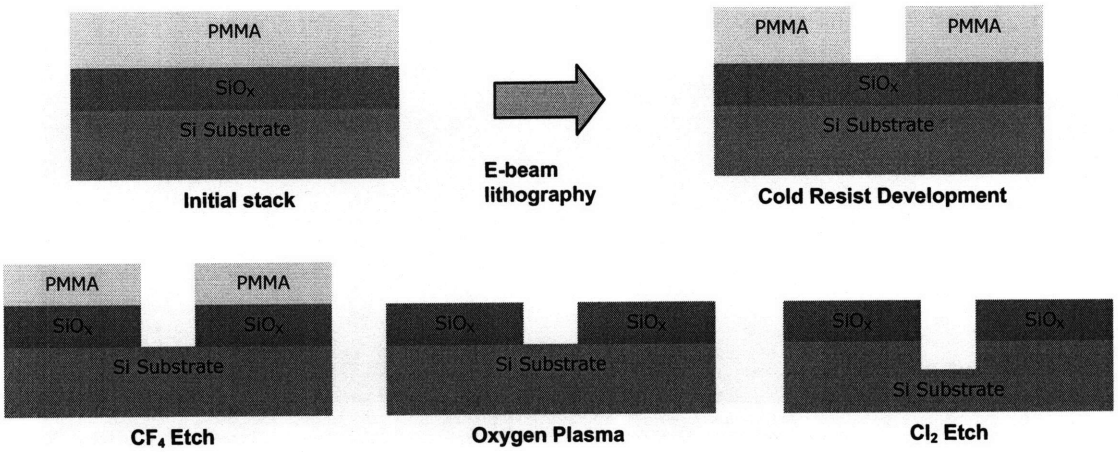

Figure 3.4 The process flow for silicon trench templates.

Quantum Dot Synthesis

While this thesis work did not directly involve the synthesis of quantum dots, I believe it is still instructive to briefly describe the synthesis process for colloidal nanoparticles. First, one organometallic precursor is heated in coordinating solvent under an inert atmosphere – in this case, dimethylcadmium is heated in trioctylphosphine oxide (TOPO) under nitrogen. The solution is heated until the dimethylcadmium has fully decomposed, and the solution becomes clear. The second organometallic precursor, the precursor for sulfur (typically hexamethyldisilathiane) is then rapidly injected into the solution. The introduction of the second precursor immediately starts QD growth. By a size-sensitive growth process known as Ostwald ripening, the synthesized QDs converge to a uniform size. The QDs are then rinsed by precipitation and redispersion into a non-polar solvent such as hexane or chlorobenzene. This process is then repeated with a second set of organometallic precursors, such as dimethylzinc and hexamethyldisilathiane (precursors for Zn and S, respectively), to grow a ZnS shell around the QD. This two-stage growth method often yields core-shell QDs, but, particularly with small QDs, such as the ones used in this work, this growth yields alloyed QDs instead. In either case,

the QD can undergo a ligand-exchange step to replace the capping layer with the desired functionality. Colloidal semiconducting nanoparticle synthesis is quite flexible, and applicable to a broad range of II-VI material systems, which can then be functionalized with organic capping layers to enhance their properties [2-4]. For this work, 8-nm-diameter CdZnS QD with organic-capping layers of oleic acid were used.

Templating the Self-Assembly of Quantum Dots

The first step was to calibrate the concentration of the QD solution by taking aliquots of the QD solution and diluting them in with their host solvent (chloroform), until the QDs formed a uniform layer when spin-coated onto a blank silicon test substrate, as observed by either SEM or atomic-force microscopy (AFM). Due to the sensitivity of the quantum-dot capping-layer to oxidation, and environmental hazards of handling CdZnS quantum dots, all QD solution processing, including spin-coating, was conducted in a nitrogen glove box. The test QD solution were spin-coated onto the blank substrates using a single spin speed of 3,000 rpm, with a 10,000 rpm/sec ramp, and spun for 60 seconds to ensure solvent evaporation. This work was done in collaboration with Polina Anikeeva, who developed the technique described here, and performed the majority of the work described in this section.

The QD solutions were calibrated by drawing off aliquots of the source QD solution, and diluting them with their host solvent in various ratios, typically ranging from 1:2 to 1:30. After dilution, the diluted solution was drawn into a syringe, and the syringe needle was replaced with a Millipore 1 micron filter unit. 0.2 mL of the solution was filtered through the Millipore and deposited onto the sample. Immediately after the solution was placed onto the sample, the sample was spun, using the same spin parameters mentioned above. These spin-coated samples

were then imaged by either AFM or SEM, to check for QD layer uniformity. This process was repeated until a satisfactory dilution ratio was found.

After calibrating the QD solution, aliquots with the desired concentration were drawn off and spin-coated onto the fabricated templates, using the same spin parameters as the calibration tests. The position and orientation of the gratings relative to the spin axis varied randomly from sample to sample. The results are shown in figure 3.5.

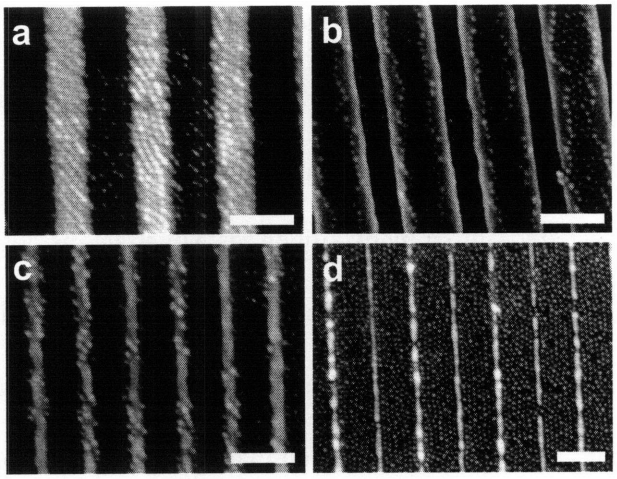

Figure 3.5 Comparison of different self-assembly regimes with differing template dimensions. (a) Si substrate with Au grating structures that are 25 nm high, 50 nm wide. (b) Etched Si substrate, with raised lines 80 nm wide, 100 nm high. (c) Si substrate with Au grating structures that are 25nm high, 10 nm wide. (d) Si substrate with Au grating structures that are 80 nm high, 10 nm wide. Scale bars: (a-d) 100 nm.

[1] B. Cord, J. Lutkenhaus, and K. K. Berggren, *J. Vac. Sci. Technol. B*, 25(6), 2013 (2007)

[2] B. O. Dabbousi, J. Rodriguez-Viejo, F. V. Mikulec, J. R. Heine, H. Mattoussi, R. Ober, K. F. Jensen, and M. G. Bawendi, *J. Phys. Chem. B*, 101 (46), 9463 -9475 (1997).

[3] C. B. Murray, D. J. Norris, and M. G. Bawendi, *J. Am. Chem. Soc.*, 115, 8706 (1993).

[4] O. I. Micic, J. R. Sprague, C. J. Curtis, K. M. Jones, J. L. Machol, A. J. Nozik, H. Giessen, B. Fluegel, G. Mohs, N. Peyghambarian, *J. Phys. Chem. 99*, 7754 (1995).

Chapter 4

Quantitative Analysis of Templated Quantum Dot Self-Assemblies

One of the primary difficulties of templated self-assembly is an objective quantification of the self-assembly results achieved. This objective analysis is critically important, if this field is to move from a phenomenological exploration of different self-assembly mechanisms towards the application of these self-assembly mechanisms to engineered systems. Quantification is the first step towards characterizing and modeling self-assembly, and a prerequisite for optimizing the templated self-assembly process. This chapter presents software developed for the automated extraction of quantitative metrics from image data.

Morphological Image Processing

Morphological image processing is the application of image processing to the analysis of spatial structures [1]. Unlike other, more widely used forms of image processing, such as image sharpening and contrast enhancement, morphological image processing does not seek to enhance the visual quality of an image, but seeks to isolate regions of interest and to identify individual features. In this thesis, morphological image processing was used to identify and locate quantum dots, and to quantitatively analyze the self-assembly of those quantum dots into the template. The overall process is illustrated in figure 4.1.

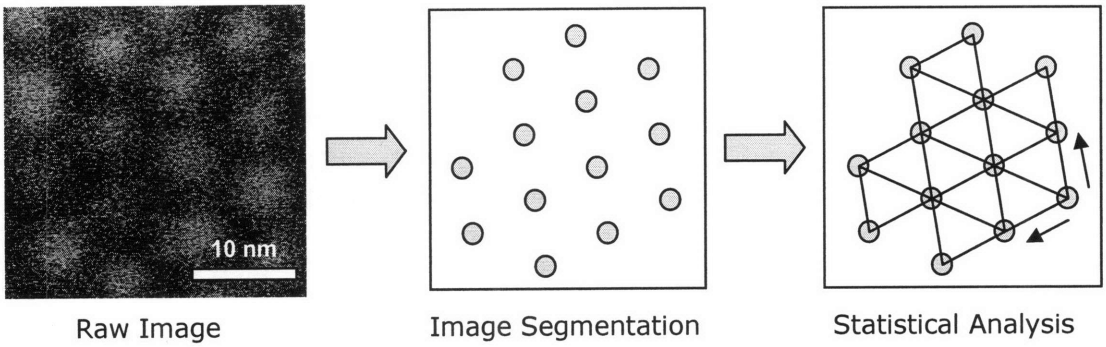

Figure 4.1 A schematic overview of the image processing process. First, the image is segmented into meaningful components – in this case, the location of each quantum dot center. From these points, measurements can be performed for each location, and the adjacency relations between the quantum dots can be investigated.

Image Segmentation

The first stage in morphological image processing is to segment the image into regions of interest – in this case, the location of each quantum dot center. This was done in several steps. The SEM image taken on the Raith 150 was stored as a monochromatic digital image. This image was first inspected manually to identify the representative quantum dot size, which was entered into the image analysis program as an *estimated quantum dot size*. This value was used to create an idealized quantum dot based on a hemispherical intensity profile with a diameter equal to the estimated QD size. A 2-D correlation of the idealized QD intensity profile with the SEM image was performed, resulting with an intensity map where the peaks of the intensity profile are likely to be the locations of quantum dots. Both the SEM image and the 2-D correlation intensity map, as well as the idealized QD intensity profile, are shown in figure 4.2.

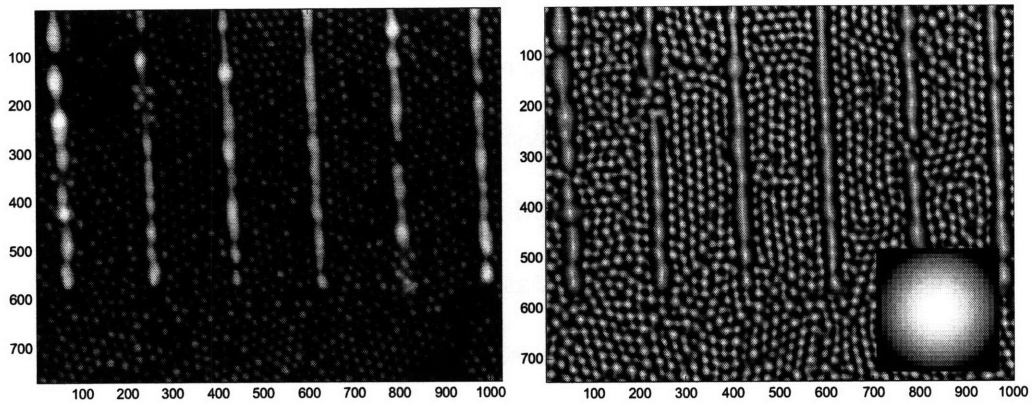

Figure 4.2 (Left) A SEM image of templated quantum dots. (Right) A 2-D correlation intensity map of the SEM image with an idealized quantum dot intensity image. (Inset) The intensity profile of an idealized quantum dot is scaled to the estimated quantum dot size to best match each SEM image. In all images, whiter shades indicated higher intensity.

As one can see from the figure 4.2, the template grating lines, which are of approximately equal length-scale to the quantum dots themselves, also have a high correlation with the idealized QD. To provide a gross discrimination mechanism, we took advantage of the large difference in intensity in the SEM image. The templated gold lines, due to their 30-60 nm height and comparatively high atomic density, yielded many more secondary electrons and were significantly brighter that the majority of the quantum dots. An appropriate threshold, determined by the value of the brightest quantum dot, was used to screen out the majority of false positives that would have otherwise arose from the template lines.

Using these appropriate thresholds, a local maxima search of the correlation intensity map was used to find the individual peaks of the image, which corresponded to the locations of the individual quantum dots. To perform this task, a sliding-neighborhood maxima search was used, with the size of the neighborhood corresponding to the estimated quantum dot size. This method prevented the erroneous classification of multiple local maximums on the same quantum dot as separate quantum dots – only the highest intensity maximum would be chosen. Combined

with the smoothing effect of the 2-D correlation, this method was very effective in mitigating the effects of image noise on QD identification and registration.

As shown in figure 4.3, the image analysis algorithm detailed above reliably detected virtually all of quantum dots present in the SEM image. Two image analysis defects are worth noting. First, at the edges of the image, there was insufficient information for the correlation function and the sliding-neighborhood maxima search to work as intended, so a border with a depth equal to the QD radius was disregarded in the final analysis. Secondly, a few false positives on the grating lines are still present. These were handled by the next stage of image processing. Overall, this algorithm provided a reliable method for identifying the location of each quantum dot to within a single image pixel.

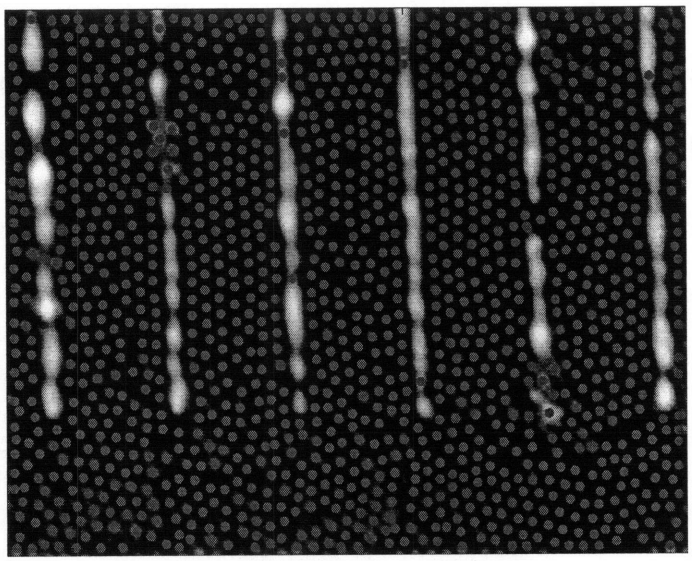

Figure 4.3 A SEM image of templated quantum dots, with the estimated positions of the quantum dots indicated by red dots.

Adjacency Relations Analysis

Having developed a technique for reliably identifying the location of quantum dots, we then analyzed the spatial relations between neighboring quantum dots, in order to begin the development of quantitative metrics for templated self-assembly.

First, again by using a sliding-neighborhood method, the nearest-neighbors for each quantum dot were identified. The connected graph of nearest-neighbors is shown in figure 4.4. As shown in the connected graph, the well-ordered regions, where the quantum dots were hexagonally close-packed (HCP), each quantum dot had six nearest-neighbors. In the regions with less ordering, the quantum dots formed irregular adjacency relations, with fewer nearest-neighbors.

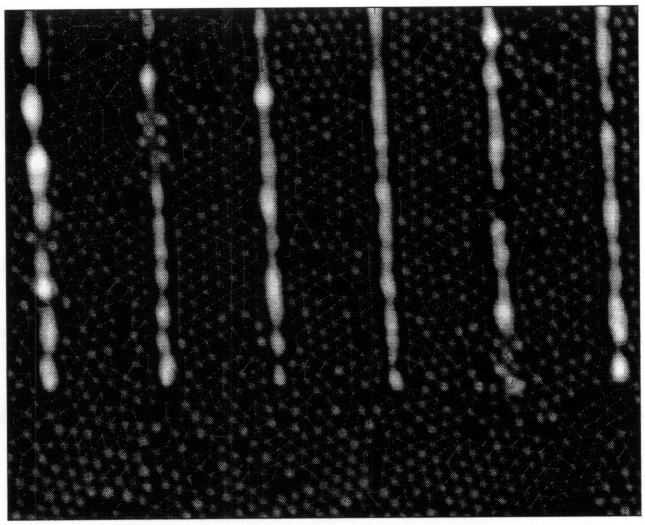

Figure 4.4 A connected graph, indicating the adjacency relations between neighboring quantum dots identified by the image processing algorithm

Next, we counted the "neighbors'-neighbors", by summing the number of nearest-neighbors each nearest-neighbor had for each quantum dot. We found that this was a useful metric for characterizing well-ordered regions. As shown in figure 4.5, the regions with highest number of neighbor's-neighbors, shown in red, were well-ordered.

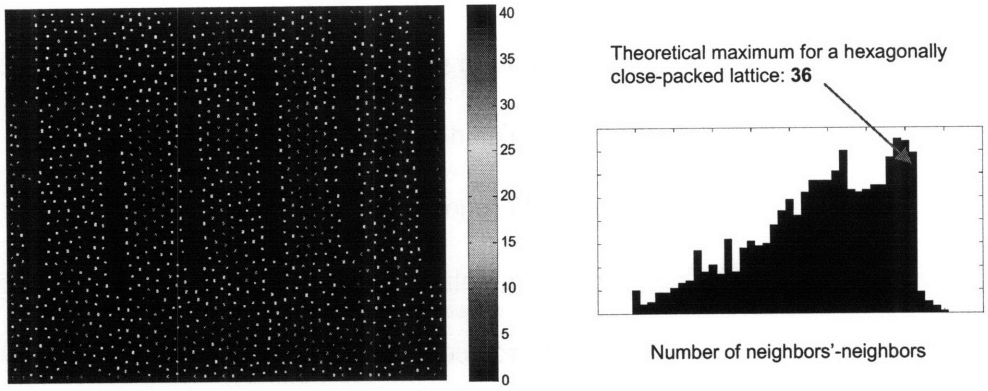

Figure 4.5 (Left) A graphical display indicating the number of neighbors'-neighbors for each quantum dot. The well-ordered, hexagonally close-packed (HCP) regions have the highest numbers of neighbors'-neighbors. (Right) A histogram of the number of neighbors'-neighbors. Notice the steep drop in quantum dot number after the theoretical maximum for an HCP lattice.

This neighbors'-neighbor metric was used to identify well-ordered regions. By identifying large concentrations of quantum dots with a value between 30 and 36, well-ordered aggregates of quantum dots were identified. Quantum dots with an intermediate value, between 20 and 29 or greater than 36, were classified as partially-ordered, and these quantum dots generally formed the boundary between well-ordered and randomly-packed regions. Partially-ordered QDs that were directly adjacent to well-ordered QDs were included into the well-ordered aggregate. The results are shown in figure 4.6. By this process, the image processing algorithm could automatically identify crystalline-like structures for analysis.

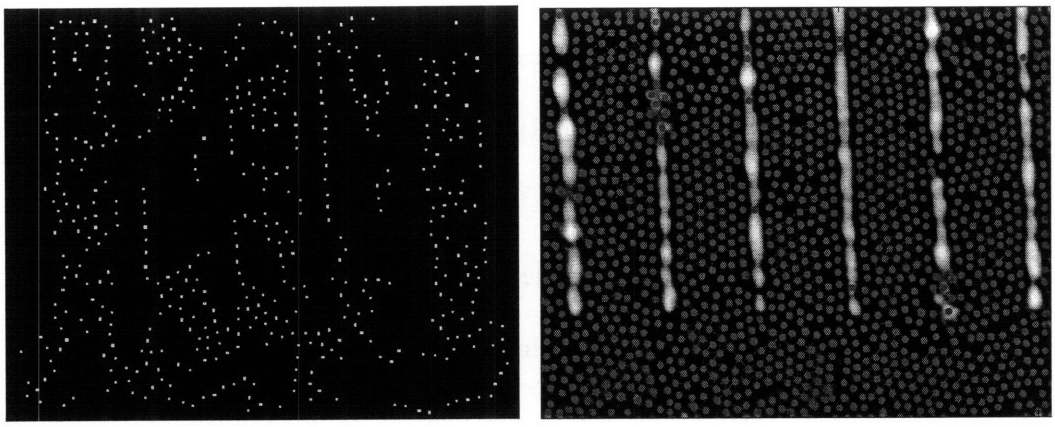

Figure 4.6 (Left) An image showing well-ordered regions in red, and partially-ordered regions in pale green. Notice that the partially-ordered regions form boundaries around the well-ordered regions. (Right) Well-ordered aggregates as identified by the image processing software. In each region, partially-ordered QDs directly adjacent to well-ordered QDs are included in the aggregate.

Statistical Analysis

Having developed the tools for the automated identification of quantum dot position and adjacency relations, we evaluated the effects of the templates on the self-assembly of the quantum dots. By comparing the adjacency relation statistics in templated and un-templated regions, the effects of the templating were studied.

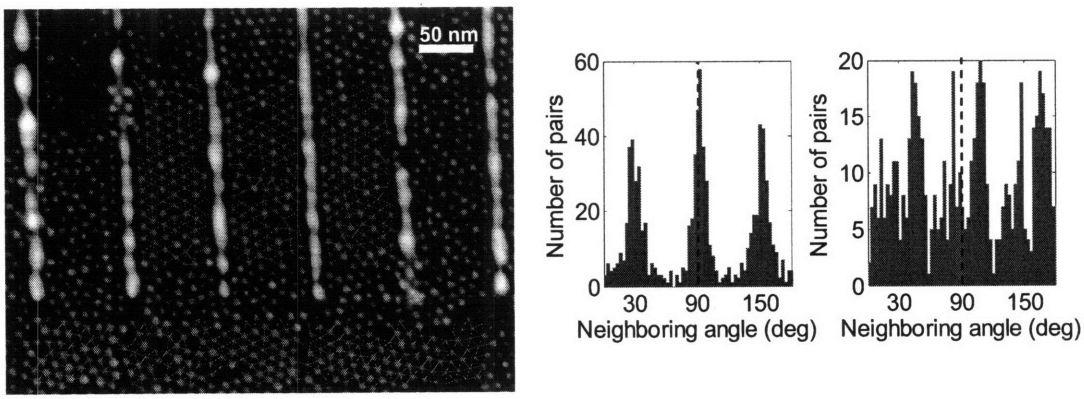

Figure 4.7 (Left) Two regions of the image were chosen for analysis. The templated region, indicated by blue, and the untemplated region, indicated by red. (Right) Histograms of the adjacency angle of the two regions indicate a strong templating effect by the gold nanowires on the self-assembly of the quantum dots.

Figure 4.7 shows such an analysis. Here, the neighboring angles refer to the angle of neighboring quantum dots relative to the horizontal axis of the image. The effect of the gold nanowires on the self-assembly of the quantum dots is clearly apparent – the templated regions showed strong ordering in the direction of the template lines, while no such alignment was present in the un-templated regions. In general, we found that, in templated regions, between 55% to 70% of the QD adjacency relations were within 10 degrees of the orientations defined by the template. This compares to between 30% to 40% for the untemplated regions, which is consistent with the 33% expected for a completely randomly-oriented quantum dot layer.

Similarly, the same statistical analysis was done on well-ordered aggregates that had been previously identified by morphological image processing. For the particular instance shown in figure 4.8, over 95% of the QD adjacency relations were within 10 degrees of the orientations defined by the template – the standard deviation was 4.8 degrees. This shows the potential level of ordering that can be achieved by this type of templated self-assembly.

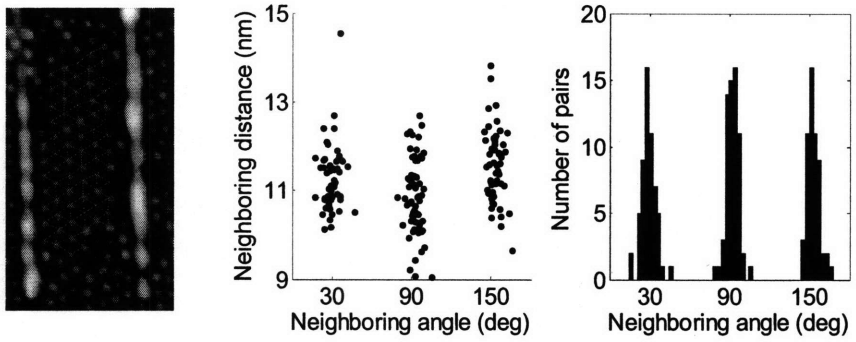

Figure 4.8 (Left) A single well-ordered, hexagonally close-packed (HCP) aggregate chosen for statistical analysis. (Center) A scatterplot showing the adjacency relation between neighboring quantum dots in the aggregate. (Right) A histogram of the adjacency relations. Notice the strong clustering around 30, 90, and 150 degrees, indicating a HCP ordering along the templated direction.

Summary

In this chapter, we present a method for the objective analysis of quantum-dot templated self-assembly through morphological image processing. We have developed methods for assessing the ordering and influence of the template on the quantum-dot self-assembly, and shown that there is a significant templating effect.

[1] P. Soille *Morphological Image Analysis*, Springer (2003)

Chapter 5

Statistical Simulations of QD Self-Assembly for QD Memory Applications

In recent years, quantum-dot based floating gate memory has been identified as a promising application for self-assembled QD techniques. Quantum-dot floating-gate memories (QD-FGMs) can potentially offer fast memory access times, long memory retention, and low power consumption. [1] However, with the constantly improving device density of conventional flash memory, self-assembled QD memories will need to meet certain performance benchmarks to become competitive. This chapter presents a theoretical examination the quantum-dot size requirements needed to satisfy ever-increasing industry standards, and the potential value of templated QD self-assembly in realizing QD-based memory architectures.

Quantum Dot Floating Gate Memories (QD-FGMs)

The structure of a floating gate memory device follows that of a field-effect transistor. In conventional floating gate memory devices, a conductive poly-crystalline silicon layer is inserted in the gate oxide, as shown in figure 5.1. This layer can be charged via tunneling current from a high gate voltage. The charge is trapped between the insulating layers, creating an effective gate voltage, which changes the threshold voltage of field-effect transistor. Because this polysilicon layer can provide a persistent gate voltage even when no voltage is applied to the gate electrode, it is called a "floating gate." In this way, the floating gate acts as a non-volatile memory, with the tunneling current from the gate electrode as a writing mechanism, and modulation in the source-drain conductivity as a readout mechanism.

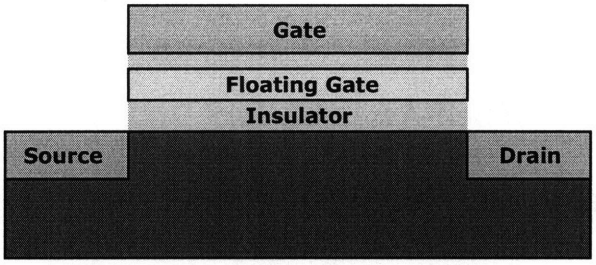

Figure 5.1 A schematic cutaway view of a floating-gate memory cell. In floating gate memories, the floating gate can be charged with tunneling current from the gate. The persistent charge in the gate can be read by detecting the change in the source-drain conductivity.

However, as the semiconductor industry seeks to further increase the density of floating-gate memories, engineering difficulties have emerged. The tunneling oxide between the channel and the floating gate is the main bottleneck preventing further scaling and higher densities. The defects in the insulator, caused by thermal cycling in fabrication, create charge-leakage pathways that reduce charge retention and device lifetime, as repeated write cycles continue to degrade the quality of the insulator. These problems limit the scaling of insulator thickness to 9~11 nm, and thinner tunneling oxides reduce yields to unacceptably low levels [2].

The quantum-dot floating-gate memory (QD-FGM) seeks to circumvent this bottleneck by exploiting the relatively poor tunneling conductivity between QDs to create a defect-tolerant floating gate. As illustrated in figure 5.2, each quantum dot constitutes an individual charge storage site [3-4] that is electrically isolated from other quantum dots by an insulating layer. Due to the resulting poor lateral conductivity in the QD layer [5-6], a single defect in the insulating oxide will only discharge the QDs in the immediate vicinity of the defect, as dictated by tunneling probabilities. This defect tolerance could potentially allow further reductions on the gate oxide thickness.

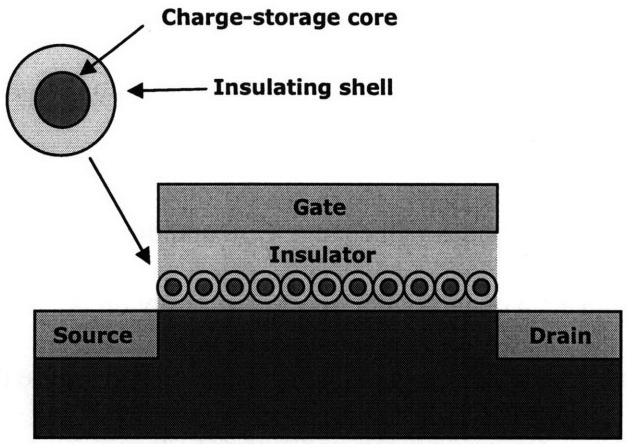

Figure 5.2 A schematic cutaway view of a quantum-dot floating-gate memory (QD-FGM) cell. In the QD-FGM, the polysilicon floating gate is replaced by a layer of quantum dots

Simulation Theory and Methodology

The simulations sought to answer the following questions:

- What is the maximum QD size that can be competitive with commercial flash memory?
- How does a well-ordered QD monolayer affect the QD size constraint?
- How does templated self-assembly affect the QD size constraint?

To examine these questions, we first created a measure for device yield. A memory device is constructed by organizing memory cells into arrays connected by bit and word lines, and it is not economically feasible to individually address each memory cell. Consistent electrical properties and minimum variation between devices is required for reliable device operation and high yields.

As mentioned earlier, the charge stored the floating gate changes the threshold voltage of the field-effect transistor. This was modeled simply as $\Delta V_T = Q / C_{OX}$, where Q is the charge present in the floating gate, and C_{OX} is the gate oxide capacitance. In this model, the shift in the threshold voltage is directly proportional to the stored charge. If we assume a consistent gate

oxide capacitance across devices and equal numbers of charge stored in each quantum dot, then the shift in threshold voltage is proportional to the number of quantum dots enclosed in each floating gate. These two assumptions do not account for non-idealities in device processing. Variations in the quality of the gate oxides will change C_{OX}, and the tunneling conductivity between the gate electrode and each quantum dot depends strongly on the intervening distance and bandgap energy structure. Using these assumptions allowed us to ignore these variances in processing, and to focus on the relative importance of quantum dot size and ordering on QD-FGM devices.

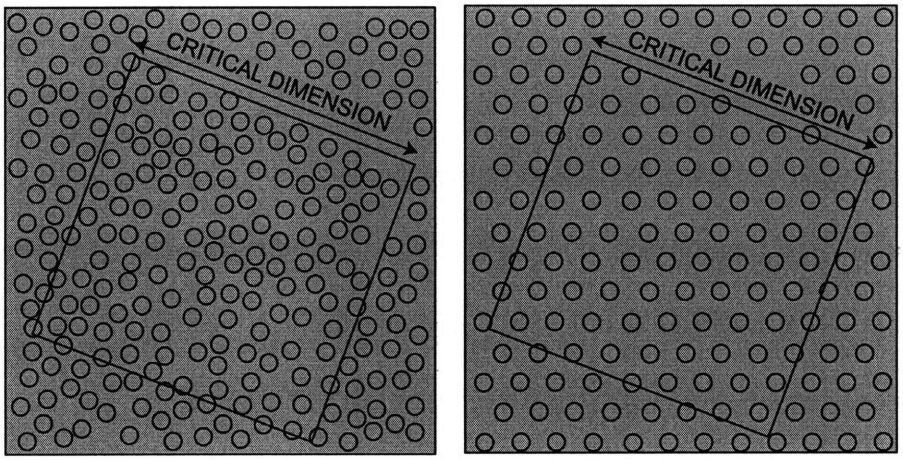

Figure 5.3 (left) A simulated gate electrode, shown as a square of the critical dimension, is superimposed onto a random monolayer of quantum dots. (right) The simulated gate electrode is superimposed onto a perfectly hexagonal array of quantum dots.

The overall simulation process is illustrated in figure 5.3. In this simulation, we consider a random quantum-dot monolayer and a perfectly-ordered hexagonally close-packed monolayer. The average distances between quantum dots in each monolayer were normalized to allow for a valid comparison. A simulated gate electrode was superimposed onto these monolayers. These gate electrodes were squares of the critical dimension at each technology-design node, where the critical dimension is the smallest feature size available at that node. The number of quantum dots covered by the gate electrode were counted for statistical analysis. Because the tunneling

conductivity is dominated by the point of the QD closest to the gate electrode, a quantum dot was considered to be 'covered' if the center of the QD was encompassed within the gate electrode boundary. For the random monolayer simulations, the positions of the quantum dots were obtained from an actual quantum dot monolayer, whose positions were detected using the imaging software described in Chapter 4 (figure 5.4).

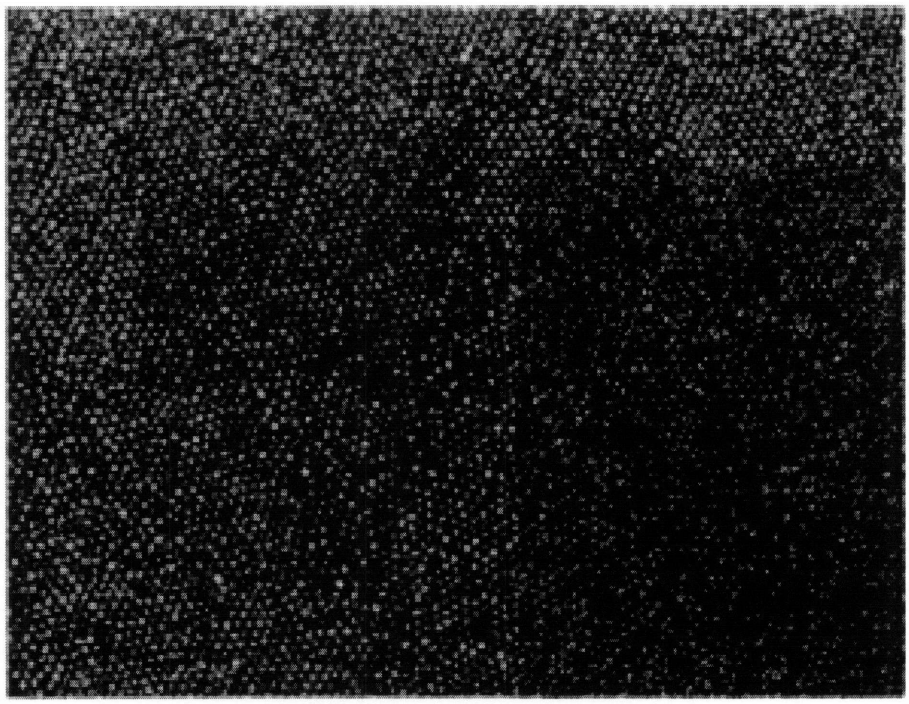

Figure 5.4 Automated identification of quantum dot positions by image processing techniques described in Chapter 4. Here, a monolayer of 8 nm CdZnS QDs have been spun-coated onto a silicon substrate. The red dots indicate the identified positions of each quantum dot.

Simulation Results

For the random QD monolayer, simulated gate electrodes were placed at random positions and orientations relative to the monolayer, and the number of quantum dots covered by the gate electrode were counted, with hundreds of thousands of iterations tried for each gate electrode size. This Monte Carlo simulation assessed the feasibility of using a random monolayer for quantum-dot floating gate memories. The results are shown in figure 5.5.

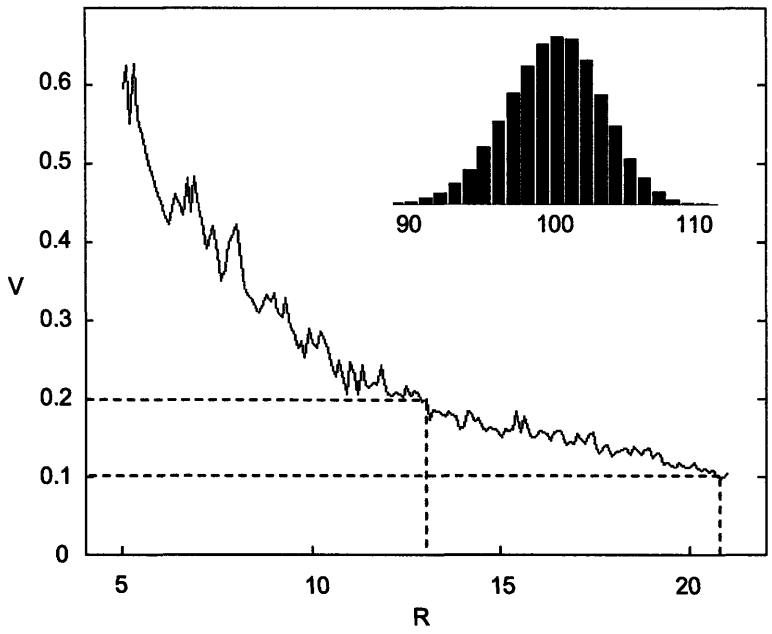

Figure 5.5 In this graph, the normalized maximum variation *V* in quantum-dot number is plotted against the size of the gate electrode *R*, as expressed in units of quantum-dot diameter. (Inset) A histogram showing the distribution of quantum-dot number when the average when the average is approximately 100 (R = 11.6).

Here, the normalized maximum variation V is defined as

$$V = \frac{Maximum\#of\ QDs - Minumum\#of\ QDs}{Average\#of\ QDs},$$

and gate electrode length R is defined as $R = \frac{Square\ Length}{Quantum\ Dot\ Diameter}$.

The variation in quantum-dot number decreased with increasing gate size, consistent with central limit theorem. To limit the variation to ±5% or ±10%, an R of 21 or 13 is required, respectively. We can readily translate these findings to each technology design node. For example, at the 50-nm design node, a random monolayer must consist of quantum dots with diameters of 2.5 nm or smaller to achieve ≤5% variation.

We repeated the Monte Carlo simulation on the ordered hexagonal array of quantum dots. In one simulation, the simulated gate electrodes were placed in random orientations and positions relative to the QD hexagonal lattice. In the other simulation, the simulated gate electrodes were placed in a fixed orientation along a lattice vector, to simulate process control over the orientation of the ordered hexagonally-packed monolayer.

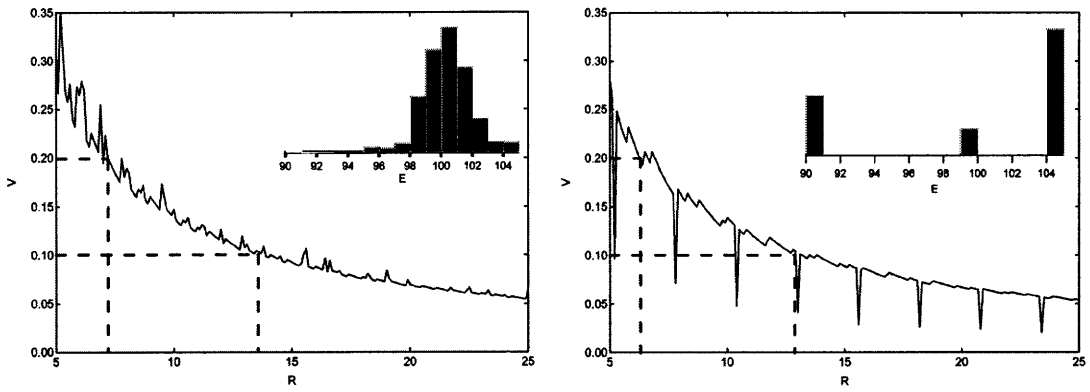

Figure 5.6 (left) The normalized maximum variation *V* plotted against gate electrode size *R* for a ordered monolayer with random orientation and position. (right) The normalized maximum variation *V* plotted against gate electrode size *R* for a ordered monolayer with defined orientation but not position. In both figures, the inset indicates the QD distribution at R = 11.6, where the average quantum-dot number is approximately 100.

Our simulations show that an ordered QD monolayer substantially relaxes the conditions required for a ±5% or ±10% variation. For ordered monolayers with random orientation and position, an R of 14 is sufficient for ±10% variation, and an R of 8 is sufficient for ±5% variation. Controlling the orientation of the QD hexagonal array gives a marginal improvement over the randomly oriented monolayer, as indicated by the dashed lines in figure 5.6. As the inset histogram shows, the number of QDs covered by the gate electrode converges on several distinct values, as rows of QDs are added or removed from coverage, creating the periodic discontinuities show in the graph. The results are summarized in the table 5.1.

	±5% variation	±10% variation
Unordered monolayer	21	13
Ordered monolayer with random orientation	14	8
Ordered monolayer with defined orientation	13	7~8

Table 5.1 The minimum size of the gate electrode, expressed in multiples of quantum-dot diameter, necessary to achieve either ±5% or ±10% variation.

Implications and Conclusions

Recent literature by commercial flash manufacturers shows that current memory cells are fabricated to about 5% tolerance – the number of charges stored in the floating gate must be within about 5% of designed specifications [2]. For our simulations, we have translated this requirement to a 5% variation in the number of QDs. These simulations show that significant improvements in device density could be achieved from controlled self-assembly. The size of the gate electrode relative to the QD diameter could be decreased with increased self-assembly control. However, even with ordered monolayers, the QD diameters required are still very small.

Technology Design Node	±10% variation	±5% variation
50 nm	7 nm	4 nm
39 nm	6 nm	3 nm
28 nm	4 nm	2 nm
20 nm	3 nm	1 nm

Table 5.2 The minimum QD diameter size required at each fabrication technology node for ordered monolayers of defined orientation.

These simulations illustrate one potential use of quantum dots where templated self-assembly of can increase the feasibility of that application. If controlled orientation and

positioning of the QD hexagonal array could be achieved, then the position of every QD would be known, and gate electrodes could be placed so that there is no variation number of covered QDs. However, even with the controlled orientation of ordered QDs shown in this thesis, significant improvements can be made.

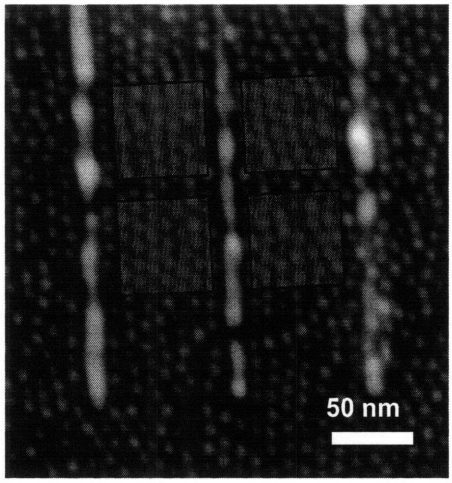

Figure 5.7 A hypothetical architecture for a quantum-dot floating-gate memory. The ordered arrays of quantum dots within the template lines are used as floating gates. The red squares indicate gate electrode placement.

[1] J. De Blauwe, *IEEE Trans. Nanotechnology* 1, 72 (2002)

[2] K. Kinam, IEDM, 1609340, 323-326 (2005)

[3] S. Tiwari, F. Rana, H. Hanafi, A. Hartstein, E. F. Crabbe, and K. Chan, *Appl. Phys. Lett.* 68, 1377, (1996).

[4] O. Nayfeh, D. Antoniadis, K. Mantey, and M. Nayfeh, *Appl. Phys. Lett.* 90, 153105 (2007)

[5] X.B. Lu, P.F. Lee, and J.Y. Dai, *Appl. Phys. Lett.* 86, 203111 (2005)

[6] A.R. Kortan, R. Hull, R.L. Opila, M.G. Bawendi, M. L. Steigerwald, P.J. Carroll, and L. E. Brus. *J. Am. Chem. Soc.* 112, 1327-1332 (1990)

Appendix 1

Matlab Code for Chapter 4

This appendix contains the MATLAB code developed over the course of this thesis research for the quantitative analysis of templated quantum-dot self-assemblies.

Integrated.m

The main MATLAB module that contains the majority of the image processing logic.

```
SP_clearallvariables = 1;

if SP_clearallvariables==1;
    clear all;
end

% modules to run

SP_runcorrelation = 1;
SP_searchmax = 1;
SP_dispimage1 = 1;                    % just image output
SP_generatedatastructure = 1;
SP_determineposition = 1;
SP_neighborimage = 1;                 % just image output
SP_graphimage = 1;                    % just image output
SP_latticeextract = 1;
SP_connectivityfinder = 1;
SP_dispallcrystals = 1;               % just image output
SP_calculatecrystalstat = 1;

% adjustable image search params
% values vary for different images
rawvalue_threshold = 200;
corr_threshold = 0.15;
QDdiameter = 24;

% graphical output parameters
dispsize = 2;

if SP_runcorrelation == 1


    % import file, fft for the heck of it
    filelist = dir('*process4.tif');
    rawimage = imread(filelist(1).name,'tif');
    [imgheight imglength] = size(rawimage);
    cheight = imgheight/2+1;
```

```
    clength = imglength/2+1;

    % run correlation
    rawimage = double(rawimage);
    searchkey = QDtemplate(QDdiameter);
    [sheight slength] = size(searchkey);
    corrimage = zeros(imgheight-sheight+1,imglength-slength+1);
    for ii = 1:imgheight-sheight+1
        for jj = 1:imglength-slength+1
            corrimage(ii,jj) = corr2(searchkey,rawimage(ii:ii+sheight-
1,jj:jj+slength-1));
        end
    end
end

if SP_searchmax == 1
    % find local maximums with threshold search of corr-maximums
    [theight tlength] = size(corrimage);
    thresimage4 = zeros(theight,tlength);
    maxsearchsize = round(QDdiameter/5);
    QDoffset = ceil(QDdiameter/2);

    for ii = 1:theight
        for jj = 1:tlength
            localmax = max(max(corrimage(max(1,ii-
maxsearchsize):min(theight,ii+maxsearchsize),max(1,jj-
maxsearchsize):min(tlength,jj+maxsearchsize))));
            if localmax == corrimage(ii,jj) & rawimage(ii+QDoffset-
1,jj+QDoffset-1) < rawvalue_threshold % & localmax > corr_threshold
                thresimage4(ii,jj) = 1;
            end
        end
    end
end



if SP_dispimage1 == 1

    compositeimage3 = rawimage;
    for ii = 1:theight
        for jj = 1:tlength
            if thresimage4(ii,jj) == 1
                compositeimage3(max(1,ii+QDoffset-1-
dispsize):min(imgheight,ii+QDoffset-1+dispsize),max(1,jj+QDoffset-1-
dispsize):min(imglength,jj+QDoffset-1+dispsize)) = -10;
            end
        end
    end
    figure; imagesc(compositeimage3);
end;

if SP_generatedatastructure == 1
    % number of QDs detected
    dotcount = sum(sum(thresimage4));
```

```
    % decrease runtime by explicit memory allocation
    for nn = 1:dotcount
        dotgraph(nn).position = [0 0];
        dotgraph(nn).estposition = [0 0];
        dotgraph(nn).numneighbors = 0;
        dotgraph(nn).numlatneighbors = 0;
        dotgraph(nn).numneighbors2 = 0;
        dotgraph(nn).neighbors = zeros(8,1);
        dotgraph(nn).latneighbors = zeros(8,1);
        dotgraph(nn).latticeadj = 0;                % used in proto5.m
        dotgraph(nn).lattice = 0;                   % used in proto5.m
        dotgraph(nn).groupnum = 0;                  % used in proto6.m
    end
end

if SP_determineposition == 1
    % set parameters
    searchsize = ceil(QDdiameter*sqrt(2)+1);

    % set graph discrimination thresholds
    %low_thresh = 0.83;
    low_thresh = 0.75;
    high_thresh = 1.29;

    % read in dot positions
    % create image with dot index numbers
    dottally = 0;
    dotindeximage = zeros(theight,tlength);
    for ii = 1:theight
        for jj = 1:tlength
            if thresimage4(ii,jj) == 1
                dottally = dottally + 1;
                dotindeximage(ii,jj) = dottally;
                dotgraph(dottally).position = [ii jj];
                % calculated estimated sub-pixel placement of QDs
                if ii > 1 & ii < theight & jj > 1 & jj < tlength
                    [X Y] = meshgrid(jj-1:jj+1,ii-1:ii+1);
                    Z = corrimage(ii-1:ii+1,jj-1:jj+1);
                    a = twodcurvefittest(X,Y,Z);
                    dotgraph(dottally).estposition = [a(4) a(3)];
                else
                    dotgraph(dottally).estposition =
dotgraph(dottally).position;
                end
            end
        end
    end

    % first-neighbor search using sliding neighborhood
    % link graph nodes
    neighborcount = zeros(1,dotcount);
    dottally = 0;
    neighborimage = zeros(theight,tlength);

    for nn = 1:length(dotgraph)
```

```
        kludge = dotgraph(nn).position;
        ii = kludge(1);
        jj = kludge(2);
        kludge2 = dotgraph(nn).estposition;
        rii = kludge2(1);
        rjj = kludge2(2);
        ystart = max(1,ii-searchsize);
        yend = min(theight,ii+searchsize);
        xstart = max(1,jj-searchsize);
        xend = min(tlength,jj+searchsize);
        for ii2 = ystart:yend
            for jj2 = xstart:xend
                if thresimage4(ii2,jj2) == 1
                    dotnumber = dotindeximage(ii2,jj2);
                    kludge3 = dotgraph(dotnumber).estposition;
                    rii2 = kludge3(1);
                    rjj2 = kludge3(2);
                    if sqrt((rii-rii2)^2 + (rjj-rjj2)^2) <
(QDdiameter*high_thresh) & sqrt((rii-rii2)^2 + (rjj-rjj2)^2) >
(QDdiameter*low_thresh)
                        neighborcount(nn) = neighborcount(nn) + 1;
                        dotgraph(nn).neighbors(neighborcount(nn)) = dotnumber;
                    end
                end
            end
        end
        neighborimage(ii,jj) = neighborcount(nn);
        dotgraph(nn).numneighbors = neighborcount(nn);
    end

    % Use graph to find second-neighbor
    neighborcount2b = zeros(1,dotcount);

    for nn = 1:length(dotgraph)
        count = 0;
        for ii = 1:dotgraph(nn).numneighbors
            count = count + dotgraph(dotgraph(nn).neighbors(ii)).numneighbors;
        end
        dotgraph(nn).numneighbors2 = count;
        neighborcount2b(nn) = count;
    end
end

if SP_neighborimage == 1
    % draw first-neighbor image
    compositeimage5 = zeros(imgheight,imglength);
    for nn = 1:length(dotgraph)
        kludge = dotgraph(nn).position;
        ii = kludge(1);
        jj = kludge(2);
        compositeimage5(max(1,ii+QDoffset-1-
dispsize):min(imgheight,ii+QDoffset-1+dispsize),max(1,jj+QDoffset-1-
dispsize):min(imglength,jj+QDoffset-1+dispsize)) = dotgraph(nn).numneighbors;
    end
    figure; imagesc(compositeimage5);
```

```
    % draw second-neighbor image
    compositeimage6 = zeros(imgheight,imglength);
    for nn = 1:length(dotgraph)
        kludge = dotgraph(nn).position;
        ii = kludge(1);
        jj = kludge(2);
        compositeimage6(max(1,ii+QDoffset-1-
dispsize):min(imgheight,ii+QDoffset-1+dispsize),max(1,jj+QDoffset-1-
dispsize):min(imglength,jj+QDoffset-1+dispsize)) = dotgraph(nn).numneighbors2;
    end
    figure; imagesc(compositeimage6);
end

if SP_graphimage == 1
    % draws connected graph (run cleanproto, proto3 first)

    compositeimage7 = zeros(imgheight,imglength);
    for nn = 1:length(dotgraph)
        kludge = dotgraph(nn).position;
        ii = kludge(1);
        jj = kludge(2);
        compositeimage7(max(1,ii+QDoffset-1-
dispsize):min(imgheight,ii+QDoffset-1+dispsize),max(1,jj+QDoffset-1-
dispsize):min(imglength,jj+QDoffset-1+dispsize)) = dotgraph(nn).numneighbors2;
    end
    figure; imagesc(compositeimage6);

    for nn = 1:length(dotgraph)
        kludge = dotgraph(nn).position;
        yy = kludge(1);
        xx = kludge(2);
        for ii = 1:dotgraph(nn).numneighbors
            if dotgraph(nn).neighbors(ii) > nn
                kludge = dotgraph(dotgraph(nn).neighbors(ii)).position;
                yy2 = kludge(1);
                xx2 = kludge(2);
                line([xx+QDoffset-1, xx2+QDoffset-1], [yy+QDoffset-1,
yy2+QDoffset-1], 'Color', 'r' );
            end
        end
    end
end

if SP_latticeextract == 1
    % catagorize "lattice-ness" by second-neighbor parameter
    thres1 = 19;
    thres2 = 29;
    thres3 = 36;

    for nn = 1:length(dotgraph)
        if dotgraph(nn).numneighbors2 <= thres1
            dotgraph(nn).lattice = 0;
            dotgraph(nn).latticeadj = 0;
        else
            if dotgraph(nn).numneighbors2 <= thres2
                dotgraph(nn).lattice = 1;
```

```
                dotgraph(nn).latticeadj = 0;
            else
                if dotgraph(nn).numneighbors2 <= thres3
                    dotgraph(nn).lattice = 2;
                    dotgraph(nn).latticeadj = 1;
                else
                    dotgraph(nn).lattice = 2;
                    dotgraph(nn).latticeadj = 1;
                end
            end
        end
    end

    % display image
    compositeimage8 = zeros(imgheight,imglength);
    for nn = 1:length(dotgraph)
        kludge = dotgraph(nn).position;
        ii = kludge(1);
        jj = kludge(2);
        compositeimage8(max(1,ii+QDoffset-1-
dispsize):min(imgheight,ii+QDoffset-1+dispsize),max(1,jj+QDoffset-1-
dispsize):min(imglength,jj+QDoffset-1+dispsize)) = dotgraph(nn).lattice;
    end
    figure; imagesc(compositeimage8);

    for nn = 1:length(dotgraph)
        if dotgraph(nn).lattice == 1
            latticeneighbors = 0;
            for ii = 1:dotgraph(nn).numneighbors
                if dotgraph(dotgraph(nn).neighbors(ii)).lattice == 2
                    latticeneighbors = latticeneighbors + 1;
                end
            end
            if latticeneighbors > 1
                dotgraph(nn).latticeadj = 1;
            end
        end
    end

    compositeimage7 = zeros(imgheight,imglength);
    for nn = 1:length(dotgraph)
        kludge = dotgraph(nn).position;
        ii = kludge(1);
        jj = kludge(2);
        compositeimage7(max(1,ii+QDoffset-1-
dispsize):min(imgheight,ii+QDoffset-1+dispsize),max(1,jj+QDoffset-1-
dispsize):min(imglength,jj+QDoffset-1+dispsize)) = dotgraph(nn).numneighbors2;
    end
    figure; imagesc(compositeimage7);


    startpos = [];
    endpos = [];
    theta = [];
    rho = [];
    linecount = 0;
```

```
    for nn = 1:length(dotgraph)
        if dotgraph(nn).latticeadj == 1
            kludge = dotgraph(nn).estposition;
            yy = kludge(1);
            xx = kludge(2);
            for ii = 1:dotgraph(nn).numneighbors
                if dotgraph(nn).neighbors(ii) > nn &&
dotgraph(dotgraph(nn).neighbors(ii)).latticeadj == 1
                    kludge  =  dotgraph(dotgraph(nn).neighbors(ii)).estposition;
                    yy2 = kludge(1);
                    xx2 = kludge(2);
                    linecount = linecount + 1;
                    dotgraph(nn).latneighbors(ii) =
dotgraph(nn).neighbors(ii);
                    line([xx+QDoffset-1, xx2+QDoffset-1], [yy+QDoffset-1,
yy2+QDoffset-1],'Color','r');
                    startpos(linecount) = nn;
                    endpos(linecount) = dotgraph(nn).neighbors(ii);
                    [theta(linecount) rho(linecount)] = cart2pol(xx2-xx,yy2-
yy);
                    [ktheta(linecount) krho(linecount)] = cart2pol(xx2-
xx+rand(1)-0.5,yy2-yy+rand(1)-0.5);
                end
            end
        end
    end


end

if SP_connectivityfinder == 1
    % connectivity finder
    numnodes=length(dotgraph);
    graphmatrix = eye(numnodes);

    for nn = 1:numnodes
        if dotgraph(nn).latticeadj == 1
            [whatever groupa] = max(graphmatrix(nn,:));
            for ii = 1:dotgraph(nn).numneighbors
                if dotgraph(dotgraph(nn).neighbors(ii)).latticeadj == 1
                    [whatever groupb] =
max(graphmatrix(dotgraph(nn).neighbors(ii),:));
                    if groupa ~= groupb
                        graphmatrix(:,groupa) = graphmatrix(:,groupa) |
graphmatrix(:,groupb);
                        graphmatrix(:,groupb) = zeros(numnodes,1);
                    end
                end
            end
        else
            graphmatrix(nn,nn) = 0;
        end
    end

    groups = sum(graphmatrix) > 0;
```

```
    groupnum = 0;

    for nn = 1:numnodes
        if groups(nn) == 1
            groupnum = groupnum + 1;
            numgroupnodes = 0;
            for ii = 1:numnodes
                if graphmatrix(ii,nn) == 1;
                    numgroupnodes = numgroupnodes + 1;
                    dotgraph(ii).groupnum = groupnum;
                    groupgraph(groupnum).nodes(numgroupnodes) = ii;
                end
            end
        end
    end
end

if SP_dispallcrystals == 1
    % display differentiated crystals
    compositeimage9 = zeros(imgheight,imglength);
    for nn = 1:length(dotgraph)
        dotcoords = dotgraph(nn).position;
        ii = dotcoords(1);
        jj = dotcoords(2);
        compositeimage9(max(1,ii+QDoffset-1-
dispsize):min(imgheight,ii+QDoffset-1+dispsize),max(1,jj+QDoffset-1-
dispsize):min(imglength,jj+QDoffset-1+dispsize)) = dotgraph(nn).groupnum;
    end
    figure; imagesc(compositeimage9);
end

if SP_calculatecrystalstat == 1
    selectedgroupnum = 5;
    startpos = [];
    endpos = [];
    kthetac = [];
    krhoc = [];
    linecount = 0;

    figure; imagesc(rawimage); colormap('gray');
    for jj = 1:length(groupgraph(selectedgroupnum).nodes)
        nn = groupgraph(selectedgroupnum).nodes(jj);
        dotcoords = dotgraph(nn).position;
        yy = dotcoords(1);
        xx = dotcoords(2);
        for ii = 1:dotgraph(nn).numneighbors
            if dotgraph(nn).neighbors(ii) > nn &
dotgraph(dotgraph(nn).neighbors(ii)).groupnum == selectedgroupnum
                dotcoords = dotgraph(dotgraph(nn).neighbors(ii)).position;
                yy2 = dotcoords(1);
                xx2 = dotcoords(2);
                linecount = linecount + 1;
                line([xx+QDoffset-1, xx2+QDoffset-1], [yy+QDoffset-1,
yy2+QDoffset-1],'Color','r');
                startpos(linecount) = nn;
                endpos(linecount) = dotgraph(nn).neighbors(ii);
```

```
                [thetac(linecount) rhoc(linecount)] = cart2pol(xx2-xx,yy2-yy);
                [kthetac(linecount) krhoc(linecount)] = cart2pol(xx2-
xx+rand(1)-0.5,yy2-yy+rand(1)-0.5);
            end
        end
    end
end
```

QDtemplate.m

Created the idealized QD template intensity profile

```
function corrtemplate = QDtemplate(diameter)
corrtemplate = zeros(diameter,diameter);
radius = diameter/2+0.5;

for ii = 1:diameter
    for jj = 1:diameter
        corrtemplate(ii,jj) = sqrt(max(0,radius^2-(ii-radius)^2-(jj-
radius)^2))/radius;
    end
end
```

twodcurvefittest.m

Tested multiple 2-D parabolic fits for sub-pixel positioning of each QD.

```
function [a] = twodcurvefittest(x,y,z)

[ymaxvect, ymaxcoordvect] = max(z);
[maxZval, maxXcoord] = max(ymaxvect);
maxYcoord = ymaxcoordvect(maxXcoord);

XcoordYrow = x(maxYcoord,:);
YcoordYrow = y(maxYcoord,:);
ZvalYrow = z(maxYcoord,:);
XcoordXcol = x(:,maxXcoord)';
YcoordXcol = y(:,maxXcoord)';
ZvalXcol = z(:,maxXcoord)';
a = curvefittest(XcoordYrow,ZvalYrow);
b = curvefittest(YcoordXcol,ZvalXcol);


xdata = [XcoordYrow,XcoordXcol];
ydata = [YcoordYrow,YcoordXcol];
zdata = [ZvalYrow,ZvalXcol];

data=[xdata;ydata];

a0 = [a.a b.a a.b b.b (a.c+b.c)/2];  % Starting guess
[a,resnorm] = lsqcurvefit(@fitfunction,a0,data,zdata);
```

curvefittest.m

Tested multiple 1-D parabolic fits along the horizontal and vertical axis of the image to generate starting parameters for the 2-D fit.

```
function [cfun] = curvefittest(xcoords, ycoords)
%CURVEFITTEST Summary of this function goes here
%   Detailed explanation goes here

[ymax, ymaxcoord] = max(ycoords);
estA = ((ycoords(ymaxcoord+1)-ycoords(ymaxcoord)) + (ycoords(ymaxcoord-1)-
ycoords(ymaxcoord)))/2;
estB = ymaxcoord;
estC = xcoords(ymaxcoord);

s = fitoptions('Method','NonlinearLeastSquares','Startpoint',[estA estB
estC]);
ffun = fittype('a*(x-b)^2+c','independent','x','options',s);

[cfun gof] = fit(xcoords',ycoords',ffun);
```

fitfunction.m

The target fit-function for the 2-D parabolic fitting.

```
function F = fitfunction(a,data)
x=data(1,:);
y=data(2,:);

ax = a(1);
ay = a(2);
x0 = a(3);
y0 = a(4);
c  = a(5);

F = ax.*(x-x0).^2+ay.*(y-y0).^2+c;
```