

Tip Based Nanomanipulation Through Successive Directional Push

Wei Zhao

e-mail: wzhao10@iit.edu

Kangmin Xu

e-mail: kxu8@iit.edu

Xiaoping Qian¹

e-mail: qian@iit.edu

Department of Mechanical, Materials and
Aerospace Engineering,
Illinois Institute of Technology,
Chicago, IL 60616

Rong Wang

Department of Biological, Chemical, and Physical
Sciences,
Illinois Institute of Technology,
Chicago, IL 60616
e-mail: wangr@iit.edu

Nanomanipulation refers to the process of transporting nanoscale components. It has found applications in nanodevice prototyping and biomolecular and cellular investigation. In this paper, we present an atomic force microscope (AFM) based approach for automated manipulation of nanoparticles to form designed patterns. The automated manipulation is based on a novel method, successive directional push. This method keeps pushing along a fixed forward direction until the particle reaches the baseline of the target position, and it then repeats the pushing process along the baseline direction. This process is iterated until the particle reaches its target position. By examining the topography of several local parallel scan lines, this method can determine the lateral coordinate of the particle. The novelty of this method lies in the fact that further pushing along the same pushing direction can be conducted without precise information about the forward position. The successive directional push method has been successfully implemented into an AFM system. We demonstrate that complex designed patterns including over 100 latex particles of 50 nm diameter can be fabricated with this method.

[DOI: 10.1115/1.4001676]

1 Introduction

Nanofabrication and nano-assembly are processes for fabricating devices of dimensions at the nanometer scale. Among various nanofabrication techniques, atomic force microscopy (AFM) has proved to be a promising and effective technique to prototype nanoscale devices and nanostructures. AFM can be used to manipulate various types of nanoscale samples, such as particles, tubes, and wires, and many types of materials, including conductive/nonconductive and magnetic, in ambient conditions and at room temperature, in air and in liquid. AFM-based nanomanipulation has been used to prototype nanoscale devices such as single-electron transistors [1], plasmonic waveguides [2], quantum-dot cellular automata gates [3], antioxidation mask [4], single wall carbon nanotube (SWCNT) based devices [5], and high density storage [6]. AFM manipulation has also found various biological applications [7]. DNA has been used as a template for chemically programmed assembly of molecular-scale devices or even biological structures [8]. Also, the desire to use DNA as building blocks in electronic circuits has motivated the studies on DNA manipulation [9].

AFM-based manipulation has been studied for some time [10]. During manipulation, the AFM tip acts as a robotic end effector to move the nano-objects in different manners, e.g., pushing, pulling, or picking-up, under various physical and chemical conditions. Among these manipulation manners, pushing is the dominant one due to its simplicity in implementation. Two modes of AFM operations, contact and noncontact (tapping mode), have been used in manipulation. In this paper, we choose the tapping mode since it provides reliable particle pushing without scratching the sample or damaging the tip. One drawback of AFM-based nanomanipulation is its low throughput. This can be resolved in two ways: increasing the automation level of nanomanipulation and using multiple probes. This paper focuses on automating particle pushing.

A critical challenge in automating AFM-based nanomanipulation is that there is lack of real-time visual feedback during the manipulation process since the tip is used in both imaging and manipulation. That is, real-time monitoring of the manipulation process is not possible since the same tip is used either as an imaging or a pushing tool at a given moment. Thus, the pushing process is *blind*. However, significant spatial uncertainty exists in nanomanipulation. Spatial uncertainty could be due to the instrumentation and environment such as thermal drift, creep, and hysteresis. It could also be due to the unstable process itself in tip-particle pushing. Note that the AFM tip is sharp and tiny, and usually less than 10 nm in diameter. In tip-particle pushing, the process for such a tip to push a larger spherelike particle is inherently not stable. The particles tend to drift away from the pushing path. Rotating, rolling, and sliding occur easily. Most commercial AFM systems do not provide readily available real-time access to the process information such as deflection, topography, or amplitude while the tip is moving. Without such real-time information on the pushing process, the manipulation via commercial AFM systems has to resort to acquiring a new image after every step of manipulation to locate where the particle is. Since scanning an image may take several minutes, this push-scan-push-scan iteration via existing commercial AFM systems is an extremely labor-intensive process.

Therefore despite research progress in AFM-based nanomanipulation, the combination of spatial uncertainty and lack of real-time spatial information during the pushing process poses severe challenge in automated nanomanipulation in AFM. The goal of our research is thus to develop a manipulation system that can automatically manipulate nanoscale components to form designed patterns.

In this paper, we present a new approach for manipulating nanoparticles to form designed patterns. This approach allows for automated manipulation of nanoparticles from source locations to their target locations. In this approach, an AFM tip is used to push a nanoparticle via successive directional push. In each direction, the particle is pushed repeatedly until it reaches the baseline of its target position. It uses several local scan lines parallel to the pushing path to locate the lateral coordinate of the drifted particle. Once the lateral coordinate is determined, the particle is again pushed along the same direction until it reaches the baseline. This

¹Corresponding author.

Contributed by the Manufacturing Engineering Division of ASME for publication in the JOURNAL OF MANUFACTURING SCIENCE AND ENGINEERING. Manuscript received August 14, 2009; final manuscript received March 31, 2010; published online May 26, 2010. Assoc. Editor: Ajay Malshe.

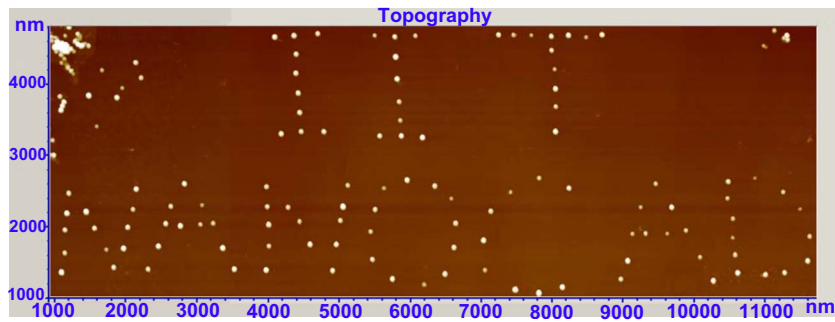


Fig. 1 An example pattern (IIT NANO CAD) created from our nanomanipulation system, consisting of 114 latex particles of 50 nm diameter each. The image size is $12 \times 5 \mu\text{m}^2$.

method thus bypasses the need for fully determining the particle position since it does not require precise forward (longitudinal) position. When the particle reaches the baseline, if there is any lateral discrepancy between the particle position and the target position, the pushing process is then repeated along the baseline direction. This process is iterated until the particle reaches the target position.

An AFM-based experimental nanomanipulation platform, based on Agilent AFM 5500, has been developed. This successive directional push method has been successfully applied onto the platform. We demonstrate that this new method is efficient and convergent. Complex designed patterns can be fabricated by this method. An example pattern “IIT NANO CAD” fabricated using our system with 114 latex particles of 50 nm diameter is shown in Fig. 1.

The remainder of this paper is organized as follows. Section 2 introduces the related literature. Section 3 details the manipulation platform and the successive directional pushing based method for automated particle pushing. Our experimental implementation, manipulation results, and pushing process analysis are presented in Sec. 4. Section 5 concludes this paper.

2 Related Work

Several groups have already performed substantial research in various manipulation schemes. A recent summary on AFM-based nanomanipulation is available in Ref. [10]. Based on the AFM scanning mode, these manipulation schemes can be divided into two categories: contact mode based and tapping mode based.

During the early period, contact mode was adopted by various research groups. Schaefer et al. [11] pushed 10–20 nm gold clusters using contact mode. They scanned the sample with tapping mode AFM. They stopped the cantilever vibration and approached the substrate until contact disabled feedback and then pushed the samples. Sheeham et al. [12] moved nanocrystals of molybdenum oxide on molybdenum disulfide surface in nitrogen environment by contact AFM with large force setpoint. Guthold et al. [13] moved particles by increasing contact force under control through a haptic device. Hashimoto et al. [14] pushed gold-coated latex particle on Si substrate. Li et al. [15] pushed latex particle on a polycarbonate surface. Hansen et al. [16] also manipulated the particles with contact mode.

Requicha et al. [17–19] found that the tapping mode-based manipulation can be of higher success rate. They moved tip in a straight line against the center of a particle, before reaching the particle, and turned off the z feedback, when the tip reached the desired end of the path, they turned on the z feedback. Yang et al. [20] manipulated the nanotube with the tapping mode.

Aside from the above two manipulation modes, Decossas et al. [21] performed manipulation with a pick-and-place manner. They succeeded in picking up Si nanocrystals deposited by silane chemical vapor deposition (CVD) on a Si surface. They placed the tip in contact with the particle and applied successive voltage

pulse of opposite polarity to the AFM tip. After the particle being picked up, the tip moved to the target location and lowered until contacting with the surface. A series of pulses was applied to release the particle.

In order to facilitate the user interaction in AFM-based manipulation, many groups have attempted to integrate path planning and augmented reality with the AFM systems. Ladjal and Ferreira [22] introduced a potential field based method in their path planning method under a virtual reality environment. Xi and co-workers [15,23] used augmented reality system to display the interaction force and real-time changes of nano-environment. It allows the operator to complete several operations without the need of a new image scan. Sitti et al. [24] realized tele-operated system with the force feedback rendered on a haptic device.

However, thus far, only limited work has been done in automated nanomanipulation. Chen et al. [25] presented a computer-aided design (CAD)-guided automatic nanomanipulation system where both nanoparticles and nanorods can be moved along the generated path. Requicha and co-workers [3,26,27] developed an automatic nanomanipulation system, where they resolved a set of automation issues, including path planning and spatial uncertainty due to thermal drift and creep.

3 Method for Automated Particle Manipulation

3.1 AFM-Based Manipulation System. Since our manipulation method is designed to work based on Application Program Interface provided by commercial AFM systems (Agilent 5500 in this case), we briefly present the hardware platform and the enhancement we added onto the commercial system.

Our AFM hardware platform is shown in Fig. 2, which consists of AFM (5500 Atomic Force Microscope, Agilent Technology Inc., Santa Clara, CA), data-acquisition (DAQ) card (NI USB 6229 BNC, National Instruments, Austin, TX), and one computer running Windows XP Professional. The microscopy, head electronics box, ac controller, and PicoScan controller constitute the AFM system; the signal access module, voltage divider, DAQ card, and computer constitute real-time data-acquisition system. The microscopy is equipped with a scanner with the x - y scan range of $90 \times 90 \mu\text{m}^2$ and the z range of $8 \mu\text{m}$.

The programmable nanomanipulation system is developed based on the AFM application program interface (API) provided by Agilent. These APIs allow the user to control the motion of tip, e.g., move and withdraw the tip, and set the operation parameters, e.g., the amplitude setpoint. However, no real-time process information (e.g., amplitude, deflection, and friction information) is available through the APIs. Thus additional hardware components have been added to the system. A DAQ card has been added to acquire real-time process information and a voltage divider is added to convert the high-voltage signal to the low-voltage signal. The computer runs the scanning and manipulation program with

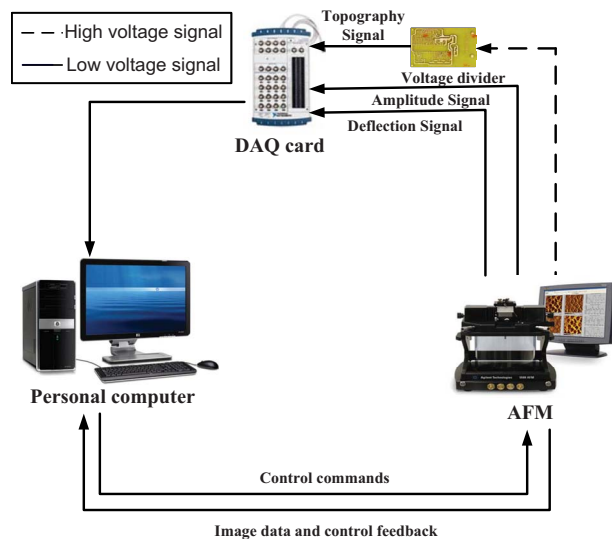


Fig. 2 Hardware architecture of a commercial AFM-based nanomanipulation system

AFM APIs. It also runs the APIs from DAQ card provider for data acquisition. The system can thus be programmed to image a sample, manipulate the sample, and acquire process data.

However, a fundamental constraint for systems with such a single-thread setup is the time delay between the data acquisition and tip movement. Typically the API command for DAQ card sampling is issued first and it is followed by the API command to move the tip. After the completion of the tip movement, the acquired data are then analyzed to monitor tip movement and the pushing process. However, due to the sequential order in the program, the precise time lapse before the tip starts moving is not uncertain. This often leads to extra 15–55 sampled data at the sample rate of 5000 samples/s. Further, the actual tip speed varies and is not directly controllable by the API. Assuming the tip travels at $20 \mu\text{m/s}$. This could mean 60–240 nm discrepancy in positioning accuracy.

A further limitation of vendor-provided APIs is that, in the API commands, the tip positions for scanning and manipulation can only be described by integer indices of the image. This would lead to low resolution in tip positioning. Thus the tip may be positioned off the particle center. Upon pushing, particles may easily move to the side.

Therefore, in order for any manipulation approach to be effective

in a commercial AFM system with minimal hardware extension and no controller redevelopment, we have to bypass the following machine constraints:

1. Precise particle location during the pushing process may not be available during the manipulation process.
2. The pushing path may not go through the particle center due to low position resolution.

3.2 Basic Operations: Pushing and Scanning. In this paper, we propose a novel nanoparticle manipulation method under the constraints described in Sec. 3.1. This section introduces two basic operations, pushing and scanning, that will be used in the algorithm for automated particle pushing.

3.2.1 Push. The pushing process is straightforward. We set the tip at the tapping mode. Before the tip reaches the particle, the feedback is turned off to avoid tip lifting. The tip will then touch the particle and move it. Upon completion of tip pushing movement, the feedback is then turned on.

In our manipulation approach, during the pushing process, the tip always moves from the initial position until the baseline of the target position. If the tip loses contact with the particle before the particle reaches the target position, the subsequent parallel line scan will then be used to locate its lateral position. A verification process will be used to verify if the particle has reached the target position.

3.2.2 Parallel Line Scan. The in-process scan is needed in order to locate the particle during the manipulation process. Note that during the particle pushing process, the particle may move to the side of the pushing path instead of going along with the tip. Once the particle loses the contact with the tip, we need to locate where the particle is in order to keep moving it until it reaches the target position. In order to avoid scanning the whole sample area, which may take several minutes, we scan several local lines parallel to the pushing path. It is thus called *parallel line scan*. It serves two purposes: (1) to determine the lateral position of the particle with respect to the pushing path and (2) verify if the particle has reached the baseline of the target position.

The DAQ card allows for the acquisition of process information during the tip movement process. However, since there is lack of correspondence between the DAQ data and the tip position, the precise location of particle position cannot be known from the DAQ data. Instead, we use several line scans that are parallel to the pushing path to locate the lateral coordinate of the particle position. After we identify which scan line the particle lies, we examine the scanning signal from that line again to verify if the particle has reached the baseline of the target position.

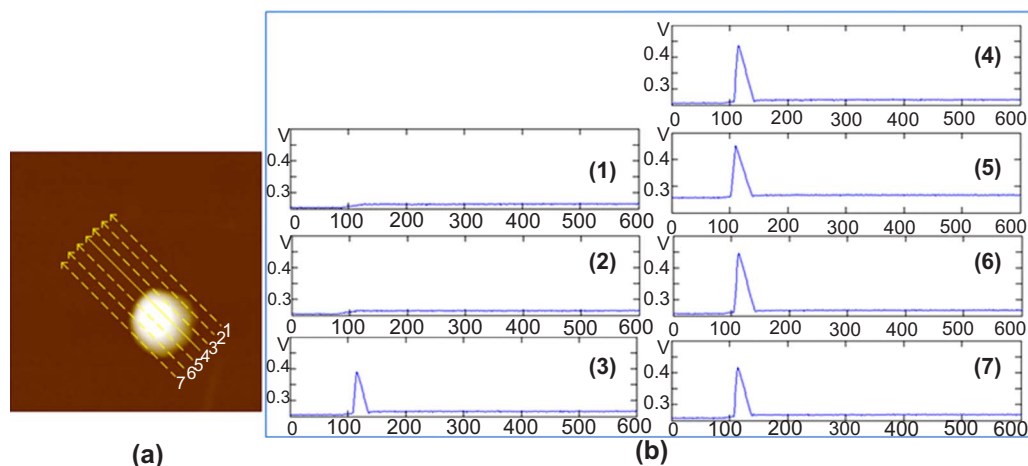


Fig. 3 Parallel line scans to locate the lateral position of the particle: (a) line scans along the push path and (b) the topography signal of the line scans

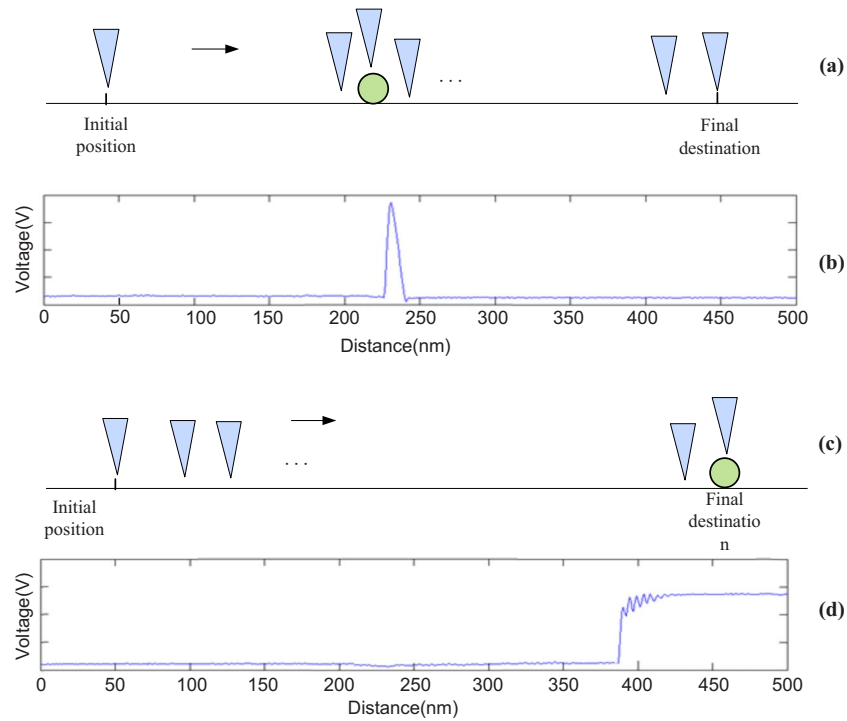


Fig. 4 Pushing verification: in (a) and (b) where a particle has not reached the destination and the corresponding topography signal; ((c) and (d)) a particle has been moved to its destination and the corresponding topography signal.

The basic idea for parallel line scans to locate the particle's lateral position is shown in Fig. 3. Even though the precise location of the particle along the pushing path is not directly known from the sampled data, however, we can compare which scan line has the highest topography signal. The particle center must lie on (or nearest) to this line.

An example of the parallel line scans is shown in Fig. 3. The center solid line (line 4) in Fig. 3(a) shows the pushing path. Figure 3(b) shows the real-time topography signal of the parallel line scans, which shows that line 5 has the highest topography signal. That is, due to the lateral drift during the pushing, the particle has moved away from the pushing path (line 4).

These parallel lines are offsets from the pushing path with $(0, \pm d, \pm 2d, \dots, \pm nd)$ separation, where d is the pixel size. The number of scan lines on each side, n , can be determined by the pixel resolution and the particle radius. The lateral drift due to the pushing is typically related to the tip radius R . Experimental information on this is available in Sec. 4.3.

To be safe, we consider $1.5R$ as the largest lateral drift, then we can determine n by

$$n = \left\lceil \frac{3R}{2} \frac{\text{scan size}}{\text{image size}} \right\rceil \quad (1)$$

For example, if the scan size is $2.5 \mu\text{m}$, the image size is 256×256 , and the radius of the dilated particle is 40 nm , then

$$n = \left\lceil \frac{3 \times 40}{2} \frac{2500}{256} \right\rceil = 7 \quad (2)$$

After a pushing operation, a verification operation will be done to determine if the particle has been pushed to its destination or not. After the lateral position of the particle has been identified, the verification involves the examination of the real-time topography signal from the line with highest topography signal (see Fig. 4). If the particle still lies in the middle of the pushing path, there would be a bump in the middle of the topography signal, see Fig. 4(b). If the particle has been successfully moved to the end posi-

tion of the pushing path, the tip would be end up tapping on the particle and the topography will be kept at a higher value at the end of the scan line (Fig. 4(d)). Using this characteristic of topography signal, we can determine whether the particle has been moved to the baseline of the target position successfully or not. The difference between the highest and lowest voltages can be calculated from the nanoparticle diameter. Suppose D is the diameter of nanoparticle (nm), S is the sensitivity of z -piezo (nm/V), A is the amplification factor of the voltage divider, and H is the difference between the highest and lowest voltages, then we have

$$H \times S \times A = D \quad (3)$$

therefore

$$H = \frac{D}{A \times S} \quad (4)$$

3.3 Algorithms for Automated Particle Pushing. Based on the previous basic pushing and scanning operations, we can now develop a method for automatically moving a particle from a source position to a target position.

This method for automated manipulation of nanoparticles from a source position to a target position involves two algorithms. The first algorithm is a basic one, called successive directional pushing (SDP). It can automatically push a nanoparticle from the source position to the baseline of the target position. The second algorithm for automated particle pushing then iterates the first one to move the particle to the target position.

The basic steps of sequential direction pushing are shown in Fig. 5. The input of the algorithm is two points: the source position $P_s = (x_s, y_s)$ and the target position $P_t = (x_t, y_t)$. The output is the particle end position P_n (i.e., after n times of pushing) that lies on the baseline of P_t . Note the baselines are lines going through P_s or P_t and are perpendicular to the line $P_s P_t$. Based on the two basic operations described in Sec. 3.2, pushing and parallel line

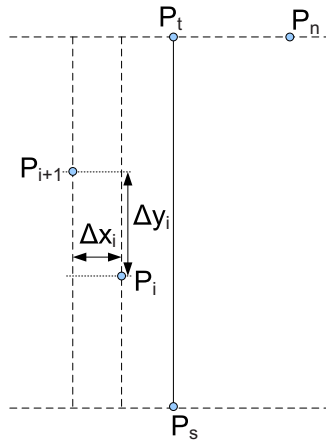


Fig. 5 Successive directional push with lateral drift Δx_i and forward movement Δy_i for the i th push

scans, we can now present the SDP algorithm.

Algorithm 1. Sequential directional pushing

- Step 1.** Set the pushing step counter $i=0$. The particle center location $x_i=0$.
- Step 2.** Push the particle along the path passing through x_i and parallel to the line P_sP_t .
- Step 3.** Conduct parallel line scans along the scan path in Step 2 and its n offsets on both sides. Identify the line with highest topography signal as the line t . The particle's current lateral position is then $x_{i+1}=x_i+t*d$. Set $i=i+1$.
- Step 3.** Verify if the particle has reached the baseline of the target by examining the signal on the line t . If the particle has reached the baseline, we have $P_n=(x_i, y_i)$ and stop the process. Otherwise, go to Step 2.

Note that, in the above algorithm, it is assumed that a local coordinate system has been established with the origin at P_s and the y axis along P_sP_t . This SDP algorithm ensures that the particle will be moved to a position P_n that lies on the baseline of the target position.

A second algorithm that iterates the SDP will then move the particle to the target position. To distinguish the pushing iterations within each SDP process from the SDP iterations, we call every pushing within a SDP process as an *iteration* and every SDP process as a *cycle*. In this algorithm, all SDP cycles will have the same target position P_t but a different starting position P_s . The starting position of the $(j+1)$ th cycle is the final position P_n of the particle from the previous cycle. The algorithm will terminate if the distance between the particle position and the target position is smaller than a user-specified threshold ϵ . The threshold ϵ can be set as the pixel resolution size.

The input for the *automated particle pushing* algorithm is the particle source position P_s and the target position P_t . We thus have the following algorithm for automated particle pushing.

Algorithm 2. Automated particle pushing

- Step 1.** Set the SDP cycle counter $j=0$. Let the current particle position be $P^j=P_s$ and the target position be P_t .
- Step 2.** Call SDP with P^j and P_t as the source and target positions. The output of this SDP is P_n . Set $j=j+1$;
- Step 3.** Calculate the distance $|P_n-P_t|$. If $|P_n-P_t|<\epsilon$, stop the process. Otherwise, Set $P^j=P_n$ and go to Step 2.

We have the following brief remarks on the two algorithms.

Remark 1. Within every SDP cycle, pushing and parallel line scan paths of all iterations are always parallel and of equal length as P_sP_t .

Remark 2. Every subsequent SDP pushing and parallel line

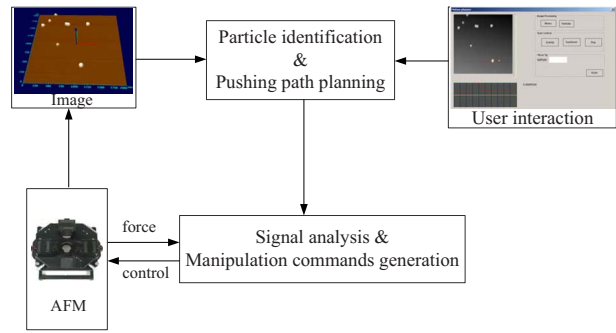


Fig. 6 General process flow for the nanomanipulation system

scan direction is orthogonal to previous SDP cycle's pushing direction.

As shown in Fig. 5, in the j th SDP cycle, the pushing direction is along P_sP_t . In the $(j+1)$ th SDP cycle, the pushing direction is along P_nP_t . Thus, every subsequent SDP pushing direction is orthogonal to previous SDP cycle's pushing direction.

The convergence of the automated particle pushing algorithm depends on, in each SDP cycle, whether the cumulative lateral drift $\sum_{i=1}^n \Delta x_i$ is smaller than the forward movement distance P_sP_t . Our experiments demonstrate this is indeed the case and SDP converges very rapidly. Details on the SDP convergence are in Sec. 4.3.

4 Experimental Implementation

4.1 Manipulation Platform and Process. The hardware platform, as shown in Fig. 2, has been developed based on Agilent 5500 AFM. The general process flow for automated nanoparticle manipulation with tapping mode is illustrated in Fig. 6.

In this system, the AFM system scans the sample and obtains the image. From the scanned image, the particle identification module locates the particle source positions. The desired particle target positions are specified by a user through a graphic user interface. Based on the initial source and final target positions, the paths are generated. Real-time voltage signals are obtained through DAQ card from the AFM scanner. They are then used to drive the automated particle pushing algorithm.

4.1.1 Manipulation Process. The overall manipulation process includes two types of operations: preprocessing, which includes image acquisition, particle identification, and pushing path generation, and automated particle pushing, which includes automated pushing and parallel line scanning. Among these operations, the parallel line scanning and automated pushing operations have been described in Algorithm 1 and Algorithm 2. We briefly describe below the preprocessing steps (Fig. 7).

- (1) *Image acquisition:* 2D images, e.g., topography and amplitude, are acquired from AFM scanning. The purpose of image acquisition is to obtain a 2D topography image from which the source particles can be identified.

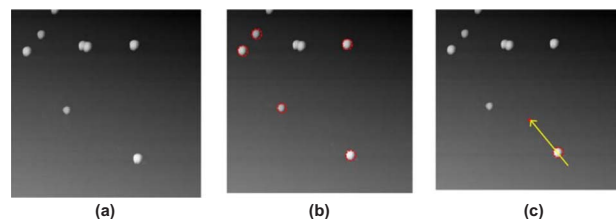


Fig. 7 Preprocessing steps for automated particle processing: (a) sample scanning, (b) particle identification, and (c) pushing path specification

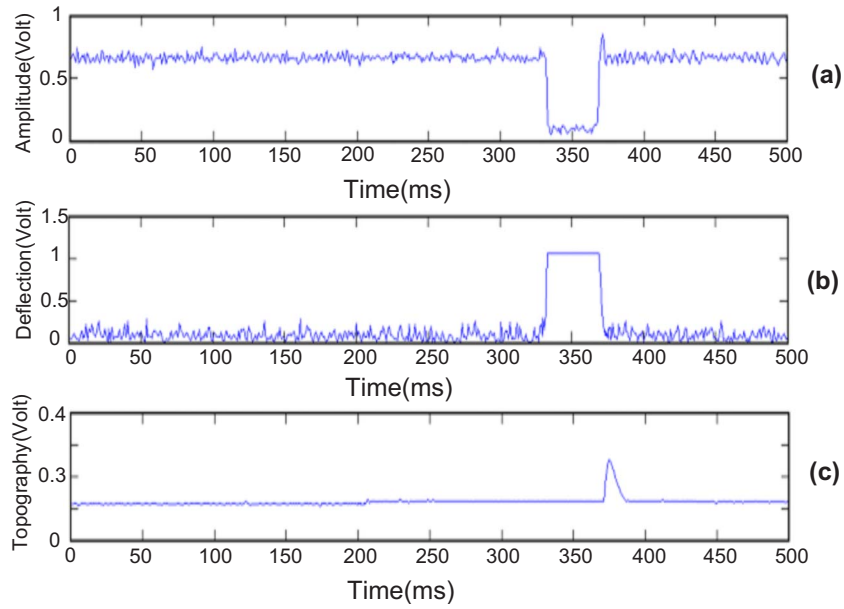


Fig. 8 Real-time signals during the pushing: (a) amplitude, (b) deflection, and (c) topography

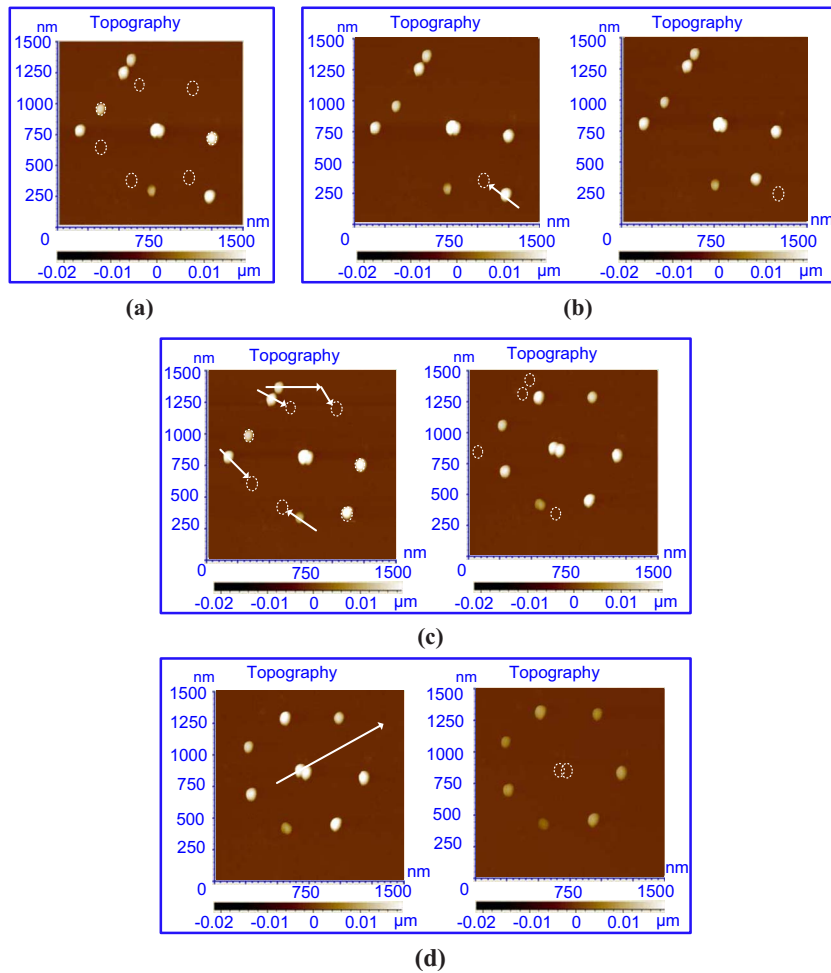


Fig. 9 Manipulation process for a seven-particle star: (a) initial scanned image and the user-designed final pattern, (b) a specified pushing path and the pushing result, (c) several particle pushing paths and the final result, and (d) removing unwanted particles

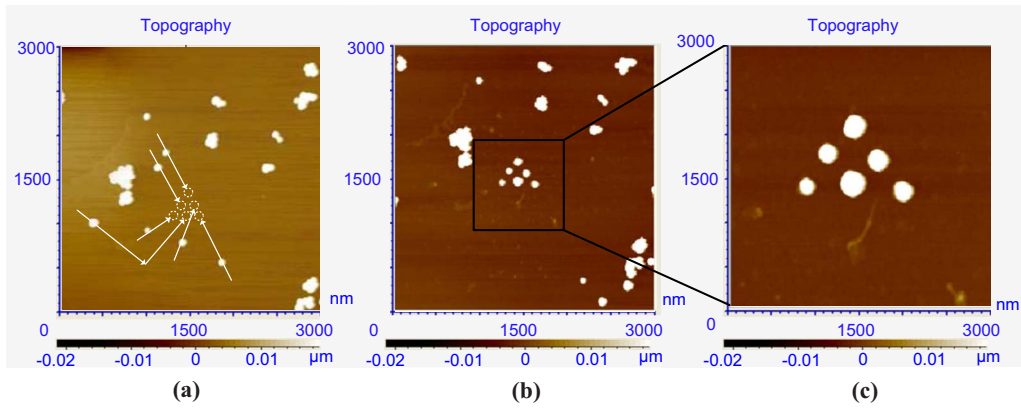


Fig. 10 Manipulation process for a six-particle triangle: (a) Initial image and the designed pattern, (b) final result, and (c) zoom-in of the result

- (2) *Particle identification*: In order to generate pushing path, one needs to identify and locate where the source particles are in the sample substrate. In our experiments, we consider the particles as a sphere. A height threshold algorithm can be used to identify spherical particles.
- (3) *Pushing path generation*: The pushing paths start from the centers of particles identified in the previous step and ends at their corresponding target positions. In our current system, the target positions are determined by a user through the graphics interface.

After a set of pushing paths has been generated, the automated particle pushing algorithm is then invoked for each path.

4.1.2 Sample Preparation. Colloidal solutions of latex beads of diameter 50 nm diluted in purified water have been used in sample preparation. A droplet of this solution is placed onto a silicon substrate and dried in air.

4.1.3 Real-Time Data Analysis. The AFM tapping mode is used in both scanning and pushing process with the feedback turned on in imaging and off in pushing. Real-time signals make it possible to verify whether a manipulation operation is successful or not. In our system, amplitude, topography, and deflection signals are acquired. Figure 8 shows the above signals during a pushing process. When the tip touches the particle during the pushing process, the signal voltage (amplitude, deflection and topography)

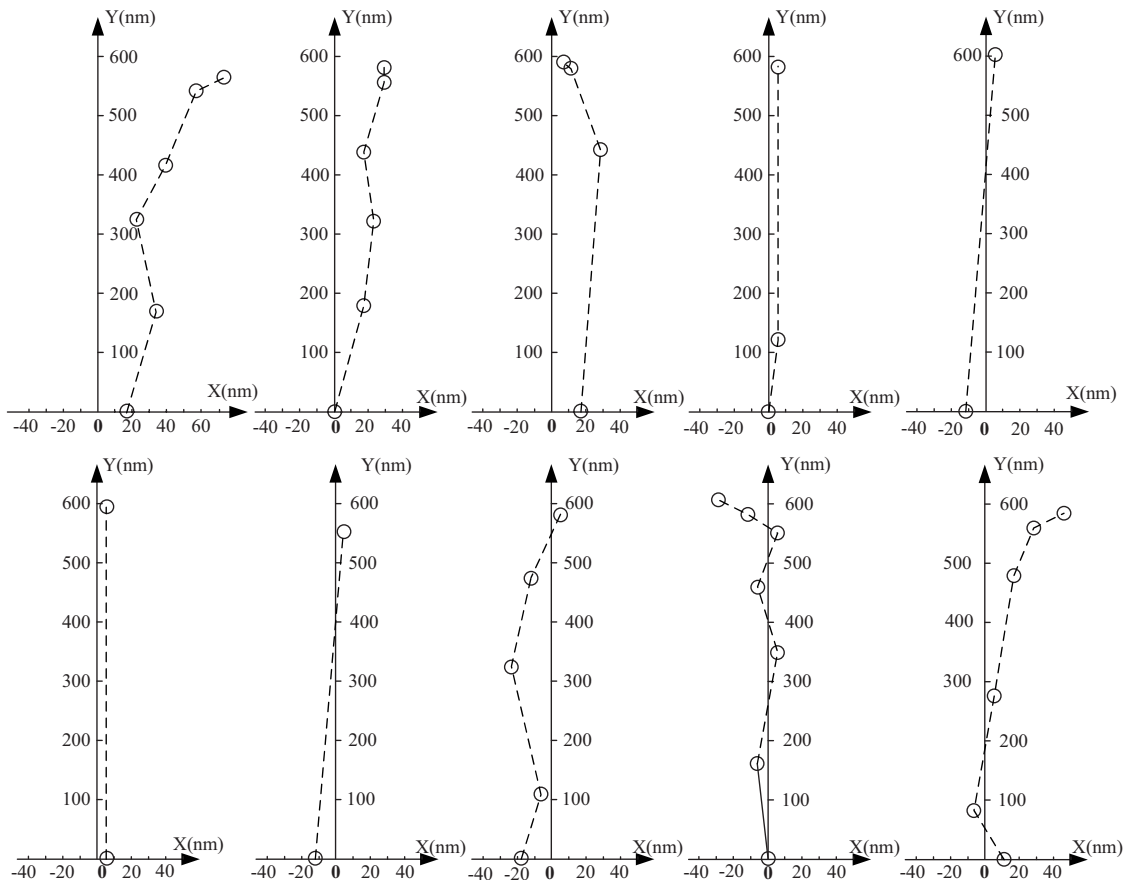


Fig. 11 Movement history of ten particles each moving forward 580 nm

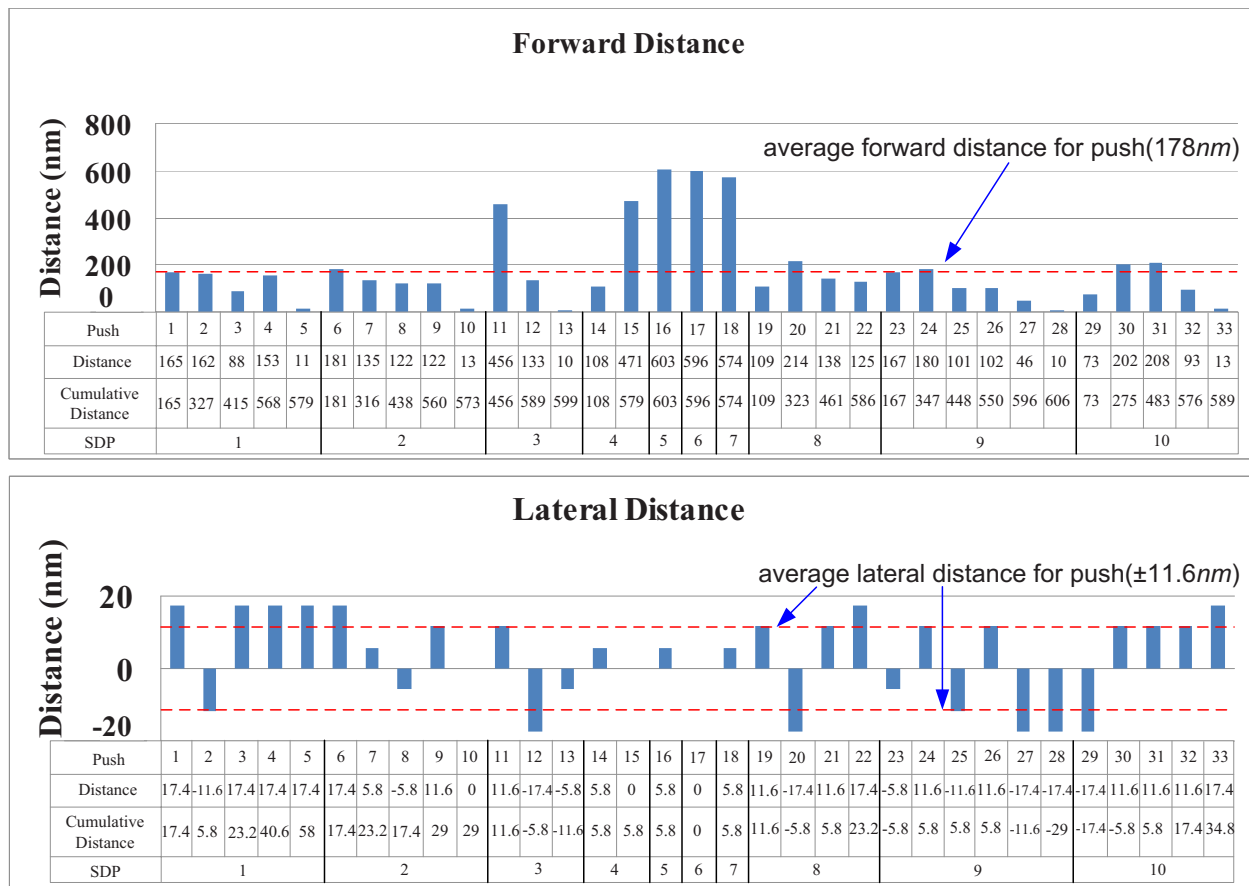


Fig. 12 Pushing statistics: forward and lateral movement

changes significantly from that where there is no contact. Deflection and amplitude signal change during the pushing process gives us a good criterion to determine whether the particle has been moved. As shown in Figs. 8(a) and 8(b), when the tip touches the particle, deflection signal jumps to a bigger value, meanwhile amplitude decreases noticeably because of the interaction force between the tip and the sample. Since the z feedback is turned off, the topography signal keeps steady until the pushing operation finishes and the z feedback is turned on, see Fig. 8(c). Both vibration amplitude and deflection signals can be used in parallel line scan to determine the particle lateral position and verify its end location. For simplicity, only topography is used in our current implementation.

4.2 Manipulation Results. We show below three examples that can demonstrate the effectiveness of our nanomanipulation system.

Figure 9 demonstrates the major steps involved in forming a seven-particle star pattern using our nanomanipulation system. The scan size is $1.5 \times 1.5 \mu\text{m}^2$. Figure 9(a) shows the initial scanned image and manually planned pushing paths to form the star pattern. Figure 9(b) shows one automated pushing operation and the result. Repeating the pushing for other particles, we obtained the results shown in Fig. 9(c). Finally, the two particles in the middle of the center are moved away from the star pattern (Fig. 9(d)).

Figure 10 shows the process of fabricating a six-particle triangular pattern. Source positions and target positions are identified in Fig. 10(a) and the result is shown in Fig. 10(b).

Figure 1 shows the complex pattern IIT NANO CAD, which is constructed from 114 particles. It shows that our nanomanipulation system, due to the automated particle pushing capability, is capable of forming complex designed patterns.

4.3 Manipulation Process Analysis. In each of the above three examples, only two SDP cycles were needed to complete the particle pushing. This demonstrates the effectiveness of our SDP method.

As the tip pushes the particle, the particle may drift to the side. The SDP algorithm works based on the assumption that, after each SDP cycle, the distance between the particle and the target position becomes shorter. That is, the cumulative lateral drift distance $P_n P_t$ should be smaller than $P_s P_t$.

In order to better understand the efficiency and the robustness of our pushing process, we conducted a set of experiments to examine the particle's lateral drift and forward movement during the pushing process. These experiments include pushing ten different particles with each moving 580 nm (100 pixels on a 256×256 image with a $1.5 \times 1.5 \mu\text{m}^2$ scan size). Each particle was limited to one SDP cycle. It took total 33 pushing to push the ten particles each moving forward approximately 580 nm. Figure 11 illustrates the movement of ten particles and Fig. 12 shows the statistics of particle forward and lateral movement.

These experiments demonstrate the high success rate of particle pushing, i.e., all ten SDPs have successfully moved the particles approximately 580 nm. Some particles only took one push (See particles 5–7 in Fig. 12). If we average the ten particle total travel distance divided by total 33 pushes, we obtain the average particle forward movement distance of 178.3 nm per push. If we discount the round-off push in the push numbers 5–33, we obtain the forward movement of 208.1 nm per push. The lateral drift distances for the 33 times of pushing operations fall within in a small range, i.e., (-17.4 nm, +17.4 nm), which corresponds to three pixels. The average lateral drift is 11.6 nm. The average ratio of lateral drift over forward movement, $\Delta x / \Delta y$, is 6.5%. The worse cumulative drift is 58 nm. The lateral drift can be either to the left or the

right of pushing paths. For SDPs involving multiple pushes (e.g., up to five pushes in these examples), the cumulative lateral drift is smaller than the sum of the absolute drifts. This is because the particles when pushed along the centers do not always move to one side. Consequently, the particles movement exhibit a zigzag pattern, as clearly shown in Fig. 11. This is the reason SDP can be very efficient since the lateral drift is not diverging too fast in either side.

5 Conclusion

This paper presents a new method for tip-based nanomanipulation. This method is based on the iterative use of successive directional pushing. This method enables the automated particle pushing from a source position to a target position.

We have successfully demonstrated automated particle pushing with this method on a commercial AFM system. Various fabricated particle patterns, including one consisting of over 100 nanoparticles, demonstrate that this method is effective for tip-based nanomanipulation. Statistical analysis on the manipulation process, including particles' lateral drift and forward movements, suggests that the SDP based automated particle pushing algorithm is convergent and efficient.

Future work will extend the nanomanipulation from the automated particle pushing to automated fabrication of designed patterns. This will be achieved by adding automated pushing path planning.

Acknowledgment

We gratefully acknowledge the financial support from NSF Award Nos. 0529165, 0800912, and 090057 and the support from SME Research Initiation Award.

We are also thankful for the discussion with Dr. Guangyong Li from the University of Pittsburgh.

References

- [1] Carlsson, S. B., Junno, T., Montelius, L., and Samuelson, L., 1999, "Mechanical Tuning of Tunnel Gaps for the Assembly of Single-Electron Transistors," *Appl. Phys. Lett.*, **75**, pp. 1461–1463.
- [2] Maier, S. A., Kik, P. G., Atwater, H. A., Meltzer, S., Harel, E., Koel, B. E., and Requicha, A. A. G., 2003, "Local Detection of Electromagnetic Energy Transport Below the Diffraction Limit in Metal Nanoparticle Plasmon Waveguides," *Nature Mater.*, **2**, pp. 229–232.
- [3] Mokaberi, B., and Requicha, A. A. G., 2006, "Drift Compensation for Automatic Nanomanipulation With Scanning Probe Microscopes," *IEEE Trans. Autom. Sci. Eng.*, **3**(3), pp. 199–207.
- [4] Zheng, J., Chen, Z., and Liu, Z., 2000, "Atomic Force Microscopy-Based Nanolithography on Silicon Using Colloidal Au Nanoparticles as a Nanooxidation Mask," *Langmuir*, **16**(24), pp. 9673–9676.
- [5] Watkins, A. N. I. J. L., and Jordan, J. D., 2004, "Single Wall Carbon Nanotube-Based Structural Health Sensing Materials," *Technical Proceedings of the 2004 NSTI Nanotechnology Conference and Trade Show*, Vol. 3, pp. 11–15.
- [6] Terris, B. D., and Rishton, S. A., 1998, "Atomic Force Microscope-Based Data Storage: Track Servo and Wear Study," *Appl. Phys. (Berlin)*, **66**, pp. 809–813.
- [7] Fotiadis, D., Scheuring, S., Müller, S. A., Engel, A., and Muller, D. J., 2002, "Imaging and Manipulation of Biological Structures With the AFM," *Micron*, **33**, pp. 385–397.
- [8] Zheng, L., Brody, J. P., and Burke, P. J., 2004, "Electronic Manipulation of DNA, Proteins and Nanoparticles for Potential Circuit Assembly," *Biosens. Bioelectron.*, **20**, pp. 606–619.
- [9] Lu, J. H., 2004, "Nanomanipulation of Extended Single-DNA Molecules on Modified Mica Surfaces Using the Atomic Force Microscopy," *Colloids Surf., B*, **39**, pp. 177–180.
- [10] Requicha, A. A. G., 2008, "Nanomanipulation With the Atomic Force Microscope," *Nanotechnology, Volume 3: Information Technology*, Wiley, Weinheim.
- [11] Schaefer, D. M., Reifengerger, R., Patil, A., and Andres, R. P., 1995, "Fabrication of Two-Dimensional Arrays of Nanometer-Size Clusters With the Atomic Force Microscope," *Appl. Phys. Lett.*, **66**, pp. 1012–1014.
- [12] Sheehan, P. E., and Lieber, C. M., 1996, "Nanotribology and Nanofabrication of MoO₃ Structures by Atomic Force Microscopy," *Science*, **272**(5265), pp. 1158–1161.
- [13] Guthold, M., Falvo, M. R., Matthews, W. G., and Paulson, S., 2000, "Controlled Manipulation of Molecular Samples With the Nanomanipulator," *IEEE/ASME Trans. Mechatron.*, **5**(2), pp. 189–198.
- [14] Hashimoto, H., and Sitti, M., 2000, "Controlled Pushing of Nanoparticles: Modeling and Experiments," *IEEE/ASME Trans. Mechatron.*, **5**(2), pp. 199–211.
- [15] Li, G., Xi, N., Yu, M., Fung, W. K., and Francisco, S., 2003, "Augmented Reality System for Real-Time Nanomanipulation," *Proceedings of the IEEE International Conference on Nanotechnology*, Vol. 1, pp. 64–67.
- [16] Hansen, L. T., Kühle, A., Sørensen, A. H., Bohr, J., and Lindelof, P. E., 1998, "A Technique for Positioning Nanoparticles Using an Atomic Force Microscope," *Nanotechnology*, **9**, pp. 337–342.
- [17] Requicha, A. A. G., Baur, C., and Bugacov, A., 1998, "Nanorobotic Assembly of Two-Dimensional Structures," *IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3368–3374.
- [18] Requicha, A. A. G., 1999, "Nanoparticle Patterns," *J. Nanopart. Res.*, **1**, pp. 321–323.
- [19] Requicha, A. A. G., Meltzer, S., and Arce, P. F. T., 2001, "Manipulation of Nanoscale Components With the AFM: Principles and Applications," *Proceedings of the First IEEE International Conference on Nanotechnology*, Vol. 4, pp. 81–86.
- [20] Yang, Y., Dong, Z., and Qu, Y., 2008, "A Programmable AFM-Based Nanomanipulation Method Using Vibration-Mode Operation," *Proceedings of the Third IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, pp. 81–86.
- [21] Decossas, S., Mazen, F., Baron, T., Brmond, G., and Souifi, A., 2003, "Atomic Force Microscopy Nanomanipulation of Silicon Nanocrystals for Nanodevice Fabrication," *Nanotechnology*, **14**, pp. 1272–1278.
- [22] Ladjal, H., and Ferreira, A., 2008, "Semi-Automated Control of AFM-Based Micromanipulation Using Potential Fields," *Proceedings of the 17th World Congress International Federation of Automatic Control*, pp. 13731–13736.
- [23] Li, G., Xi, N., Yu, M., and Fung, W. K., 2004, "Development of Augmented Reality System for AFM-Based Nanomanipulation," *IEEE/ASME Trans. Mechatron.*, **9**, pp. 358–365.
- [24] Sitti, M., and Hashimoto, H., 1998, "Tele-Nanorobotics Using Atomic Force Microscopy," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 1729–1746.
- [25] Chen, H., Xi, N., and Li, G., 2006, "CAD-Guided Automated Nanoassembly Using Atomic Force Microscopy-Based Nanorobotics," *IEEE Trans. Autom. Sci. Eng.*, **3**, pp. 208–217.
- [26] Makaliwe, J. H., and Requicha, A. A. G., 2001, "Automatic Planning of Nanoparticle Assembly Tasks," *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, pp. 288–293.
- [27] Mokaberi, B., and Requicha, A. A. G., 2004, "Towards Automatic Nanomanipulation: Drift Compensation in Scanning Probe Microscopy," *University of Southern California Progress Report No. 1*.