Efficient AFM based nanoparticle manipulation via sequential parallel pushing

Kangmin Xu, Arash Kalantari and Xiaoping Qian

***Abstract*—Atomic Force Microscopes (AFM) have become a useful tool not only for imaging at the nanoscale resolution, but also a useful tool for manipulating nanoscale objects in nanoscale device prototyping and for studying molecular and cellular mechanisms in biology.**

This paper presents a method, called *sequential parallel pushing (SPP)*, for efficient and automated nanoparticle manipulation. Instead of using tip scanning to fully locate the particle center, this method uses one scan line perpendicular to the pushing direction to determine the lateral coordinate of the particle center. The longitudinal position of the particle is inferred from the position where the tip loses contact with the particle through real-time analysis of vibration amplitude of the cantilever. The particle is then pushed from the determined lateral position along the current push direction toward the baseline of the target. This process is iterated until the particle reaches the target position.

Experimental results show that the SPP algorithm, when compared with simple target-oriented pushing algorithms, not only reduces the number of scan lines but also decreases the number of pushing iterations. Consequently, the manipulations time has been decreased up to 4 times in some cases. The SPP method has been successfully applied to fabricate designed nanoscale patterns that are made of gold ($10 \sim 15nm$ diameter) particles and of 170 latex ($50nm$ diameter) particles.

Keywords: Atomic force microscopy, Nanomanipulation, Nanorobotics, Nanofabrication, Nanoparticle.

I. INTRODUCTION

The Atomic Force Microscope (AFM), invented in 1986 [1], provides a means to access nanoscale objects with high-resolution images of topography and other sample characteristics. Its salient features such as atomic imaging resolution, requiring little to no sample preparation, being applicable to both conductive and non-conductive materials in an ambient environment (air, liquid or vacuum) has led to its broad usage both as an imaging tool and a manipulation tool in many fields, such as biology, chemical, material, and nano-electromechanical systems.

An AFM images a sample by rastering a tiny tip over the sample and by moderating the interaction force between the sample and the tip. This force has also been used to modify the substrate and manipulate the objects on it. Due to its general applicability and high resolution, AFM has become a promising tool to prototype nanoscale devices [2] [3] through tip based manipulation. Fabrication of patterns and arrays of nanoparticles with reported applications, e.g., data storage or nanodevice prototyping, has been a topic of interest for quite a while [4] [5] [6].

However, fundamental challenges still exist in tip based nanomanipulation. A key challenge is *the lack of real-time visual feedback*. Since the same tip is used to both image and manipulate the nanoscale objects, the visual information on the movement of the nano-objects is not available during the manipulation process. In order to verify the manipulation result, the workspace is re-imaged which sometimes takes up to several minutes. To improve the scanning efficiency, local scanning the manipulated object, rather than the entire workspace, has been generally followed [4] [7] [8]. To provide feedback during the manipulation, augmented reality systems have been developed where deflection signal (force) is displayed in real-time [9] [10] [11] [12].

Another challenge is *the spatial uncertainties*, caused by creep, thermal drift and hysteresis [13]. It can lead to positioning inaccuracy and result in the objects being easily missed by the tip [14] [15]. Methods for overcoming these spatial uncertainties, notably based on the Kalman filter [16] and landmarks [17], have been proposed. Also, a recent survey on nanomanipulation systems is available in [5].

The general process of forming a sample pattern is illustrated is Fig. 1. An initial image of the sample is obtained at the first step (Fig. 1.a). Next, pushing paths are planned based on the desired target positions (Fig. 1.b) and finally particles are pushed one-by-one using a manipulation algorithm. In this paper we will focus on developing a time efficient method for transporting each individual particle from a source position to a target position. The task of automatic path planning will be addressed elsewhere.

Typically manipulation of each particle requires the iteration of the following two essential steps: (1) full particle localization where the particle's center coordinates are fully determined and (2) target oriented pushing where the tip is moved from the determined particle center toward the target position. We term this approach as target oriented pushing (TOP) [4] [7] [8].

In this paper, our Sequential Parallel Pushing (SPP) algorithm uses two concepts to improve the time efficiency of the nanomanipulation process: partial localization of the particle center and parallel pushing. Note that the overall manipulation time is the sum of both particle pushing time and the

This work is supported by NSF Grant CMMI-0800912 and CNS-1035844).

K. Xu and A. Kalantari are Research Assistants with the Department of Mechanical, Materials and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL 60616. Email: kxu8@iit.edu, akalanta@iit.edu

X. Qian is an associate professor with the Department of Mechanical, Materials and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL 60616. Email: qian@iit.edu.

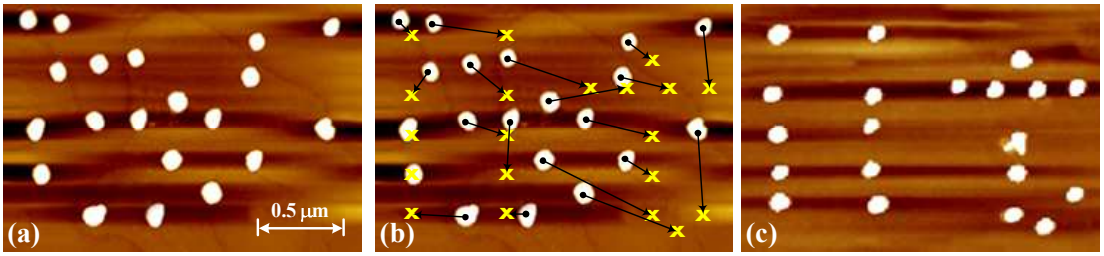

Fig. 1. A general AFM-based nano manipulation process. (a) Obtaining an Initial image, (b) Specifying target positions (shown by x) and planning pushing paths and (c)Obtaining the final result ("iit') after all particles are pushed.

scanning time used for particle localization. By using parallel pushing, we reduce the scanning time since particle's forward position can be inferred from the point where the tip-particle contact is lost. More specifically, (1) *Partial localization of the particle center*. Instead of using tip scanning to fully locate the particle center, this method uses scan lines in a single direction, perpendicular to the pushing direction, to determine the lateral coordinate of the particle center. This lateral coordinate is extracted from the topographical signal of the scan line. The longitudinal position of the particle is inferred from the position where the tip loses contact with the particle, through real-time analysis of vibration amplitude signal of the cantilever; (2) *Parallel* pushing. The particle is then pushed from the determined lateral position parallel to the initial pushing vector and toward the baseline of the target. If the lateral distance of the particle to current initial pushing line gets larger than a defined threshold, the particle center will be fully localized and the pushing direction is turned toward the target again.

In order for the SPP algorithm to work efficiently, the amount of lateral movement (caused by the particle pushed to the left or the right side of the pushing path after each push) should not exceed the forward movement. This is generally true for particle manipulation. Further, due to the stochastic nature of the lateral movement, i.e. sometimes particles are pushed to the left and other times to the right of the path, the resulting zigzag travel path naturally compensates the lateral movement. Compared to our earlier approach introduced in [18], instead of multiple scan lines, the SPP algorithm uses scan lines only in one direction perpendicular to the pushing path, to determine the lateral coordinate of the particle center.

The performance of TOP and our SPP method is compared both through simulation using a contact model between tip and particle [18] based on a set of simple assumptions and via experiments. The introduced manipulation algorithm has been implemented on an Agilent 5500 AFM. SPP method has been successfully applied to fabricate designed patterns made of latex ($50nm$ diameter) and gold ($10 \sim 15nm$ diameter) particles. Experimental comparison of this method with the target-oriented pushing methods demonstrates the superior efficiency (up to 4 times better) of the SPP method.

Based on the empirical results, two main advantages can be remarked for SPP algorithm: (1) since only partial localization of the particle is needed in SPP, it leads to fewer number of scan lines comparing to full localization and consequently the manipulation process would be more time efficient (2) SPP algorithm results in faster forward manipulation of particles when compared to simple TOP algorithms. This observation has been proven to be true based on a simple geometrical analysis. It has also been shown empirically that the path tip travels during local scanning results in a more stable and consequently more reliable reading of AFM signals comparing to TOP algorithms. This fact also helps to maintain a more stable tip-particle contact and therefore a larger forward travel of the particle.

In the remainder of this paper, the hardware platform of our manipulation system is introduced in next section. Details of the SPP algorithm is given in Section III. The SPP algorithm is analytically analyzed in Section IV and the performance of SPP algorithm is experimentally studied and compared with TOP algorithms in Section V. The SPP algorithm has been used to create complex patterns and the results are presented in Section VI. Finally the paper is concluded in Section VII.

II. Manipulation Platform

The SPP manipulation algorithm is implemented on a commercial AFM system. The original hardware platform and the features added to make the implementation of the SPP possible are presented in this section.

The manipulation hardware platform is shown in Fig. 2. It consists of an AFM (5500 Atomic Force Microscope, Agilent Technology Inc.), data-acquisition (DAQ) card (NI USB 6229 BNC, National Instruments) and a personal computer. The microscope, head electronics box, AC controller, and PicoScan controller constitute the AFM system. The signal access module, voltage divider, DAQ card and the computer comprise the real-time data acquisition system. The microscope is equipped with a piezoelectric scanner with an X-Y scan range of $90\mu m \times 90\mu m$ and a Z range of $8\mu m$.

The DAQ card can acquire six signal channels: amplitude, topography (Z piezoelectric), X piezoelectric, Y piezoelectric, deflection and the phase. In our implementation, we have just utilized the first four signals. To keep the AFM tip at a constant distance from the sample surface, a feedback control system tunes the Z voltage known as the topography signal. The voltages applied to adjust the X and Y coordinates of tip position are referred to as X and Y piezoelectric signals.

In the tapping mode and when the tip is not in contact with any object, the piezo motion along Z axis causes the cantilever to oscillate with high amplitude (typically greater than $20nm$). The tip vibration amplitude, monitored by the amplitude signal, is controlled via AC controller.

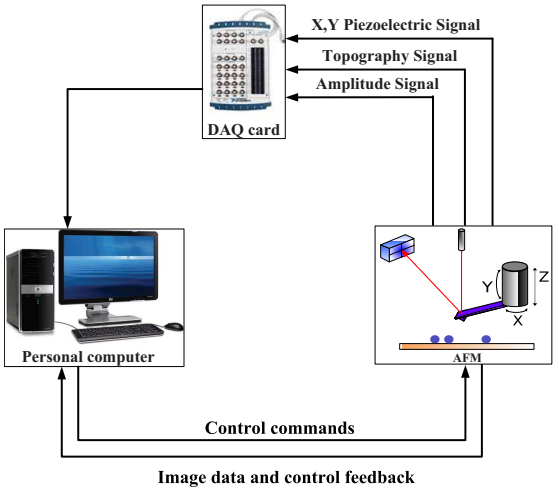

Fig. 2. Hardware platform for our nanomanipulation system

The imaging and manipulation software is developed on QT utilizing the AFM and DAQ application program interfaces (APIs). The AFM API allows the user to control the motion of the tip, e.g. move or withdraw the tip and set the operation parameters, e.g. the vibration amplitude set point. The original system hardware does not provide any real-time process information (e.g. amplitude, deflection and friction information). Thus a DAQ card and a signal access module have been added to the system to acquire real-time process information.

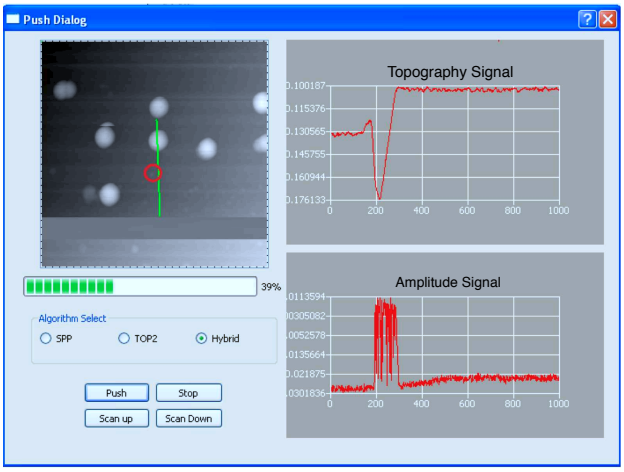

Fig. 3. Graphic User Interface of our nanomanipulation system

As scanning and manipulation processes are performed through AFM API, real time data is obtained by means of the DAQ card API. The program processes the acquired data and current position of the particle along with topography and amplitude signals are monitored on the program interface (Fig. 3).

III. Manipulation Algorithms and Time Efficiency

In this section, we first describe the usual target-oriented pushing (TOP) algorithm, then detail our sequential parallel-pushing (SPP) algorithm, and analyze the time efficiency of these algorithms.

A. Existing manipulation algorithms

Almost all manipulation algorithms proposed in previous research works follow the same procedure as the one described in Algorithm 1. The algorithm iterates till the distance of the particle to the target gets smaller than δ which is the termination condition. The only difference between them is in the local scanning subroutines.

Algorithm 1 Target Oriented Pushing (TOP)

1: $i = 0$
2: Get the start and target positions (P_0, P_f)
3: **while** $||P_i - P_f|| > \Delta$ **do**
4: $\quad i = i + 1$
5: $\quad$ Push along $P_i - P_f$
6: $\quad$ Local scan and get particle position$P_i(x_i, y_i)$
7: **end while**

Two major local scanning algorithms have been used in previous works, both of which are based on topography signals [7] [4] [8]. In first algorithm, TOP1, two perpendicular topography signals (one horizontal and one vertical signal) are acquired at every step as in Fig. 4. The horizontal scan line (S_i^h) at the ith iteration, passes through the point corresponding to the maximum topography of previous vertical scan line (S_{i-1}^v) and vice versa. The algorithm stops whenever the convergence threshold is met:

$$\delta_i = \|O_{i+1} - O_i\| < \delta_{threshold} \tag{1}$$

where the points O_{i+1} and O_i are given by:

$$O_{i+1} = [max(S_i^v), max(S_i^h)] \tag{2}$$
$$O_i = [max(S_{i-1}^v), max(S_{i-1}^h)] \tag{3}$$

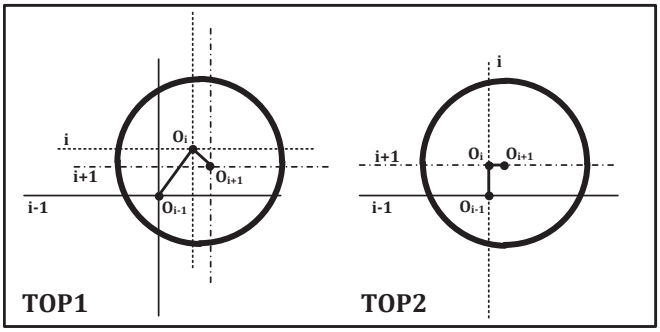

Fig. 4. Scan lines acquired in 3 consecutive iterations in current local scanning algorithms

TOP2 method uses the data of a single topography signal (S_i) at every iteration. This signal should pass through a point of maximum topography of previous signal (S_{i-1}). The orientation of scan line is also switched from horizontal to vertical or vice versa alternatively. The algorithm stops whenever the convergence condition is satisfied:

$$\delta_i = \|O_{i+1} - O_i\| < \delta_{threshold} \tag{4}$$

where the center points O_{i+1} and O_i are given by:

$$O_{i+1} = [max(S_{i+1})]O_i = [max(S_i)] \tag{5}$$

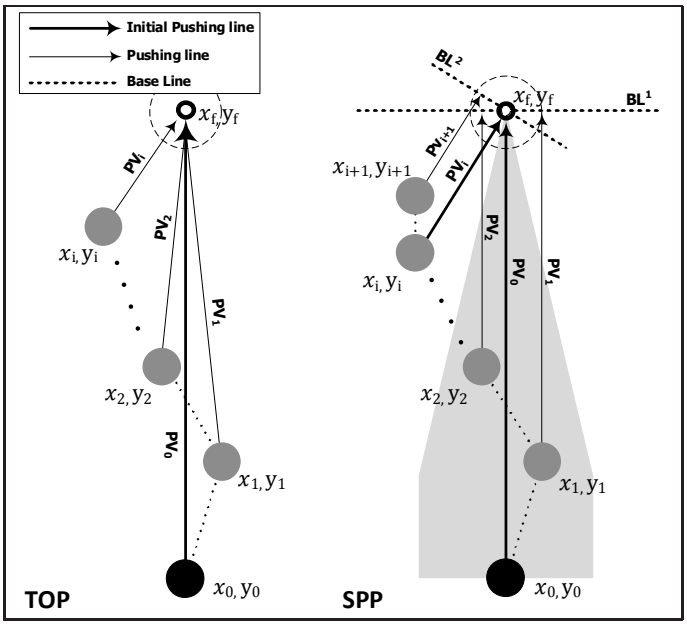

Fig. 5. The manipulation process of TOP and SPP algorithms. In TOP algorithm, particle is always pushed toward target while for SPP the particle is pushed parallel to the initial pushing line. As soon as particle leaves the current SPP domain the pushing line is turned toward target again.

B. Sequential Parallel Pushing (SPP) Algorithm

The main idea of SPP algorithm is to reduce the total manipulation time by making two modifications in both local scanning and pushing subroutines of TOP algorithms:

- Pushing direction is always parallel to the initial pushing line; therefore only one center coordinate of particle perpendicular to pushing direction is needed to be determined by local scanning.
- Following the tip-particle contact loss, the tip vibration amplitude signal is used to infer the particle forward position.

The manipulation process of SPP and TOP algorithms are compared in Fig. 5.

1) Parallel pushing: The parallel pushing can reduce the local scanning time cost by reducing the needed particle center coordinates to just one in a direction perpendicular to pushing line. Following the first push of particle from initial position to the target position, the particle center point is located and if it is not at the desired distance to the target, the pushing process will be iterated. The next pushing vector (PV_1) will be updated for SPP algorithm in a different manner comparing to TOP algorithms. This vector should be parallel to the initial pushing vector (PV_0), starting at current particle position (x_1, y_1) and ending at the intersection point of PV_1 and base line (BL_1), as depicted in Fig. 5.

2) Domain: Parallel pushing does not compensate the lateral position error. Although experimental results (demonstrated in section V) make clear how very common zigzag movement pattern of the particles limits the lateral movement, in some cases the particle may get away from initial pushing line increasingly. To limit the side error, a simple boundary is set for parallel pushing process such that as soon as the lateral distance of the particle to initial pushing line gets larger than a threshold, d_{max} , the parallel pushing direction is changed toward the target again. When the particle gets very close to the target, parallel pushing the particle toward the baseline and then to the goal would not be as efficient as pushing it directly to the target.As a result, in the vicinity of the target point this limit is tightened in proportion to the forward distance to the goal

$$|x - x_f| < \alpha |y - y_f| \tag{6}$$

The overall domain for applying parallel pushing can be written as following:

$$D = \{(x, y) : |\frac{x - x_f}{y - y_f}| < \alpha \cap \bar{\theta} | x - x_0| < d_{max}\} \tag{7}$$

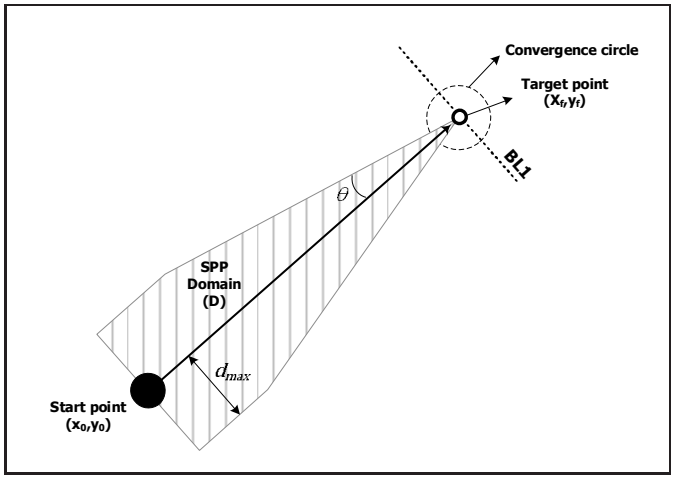

Fig. 6. SPP domain. Angle θ and maximum distance d_{max} are obtained emprically

The SPP domain is depicted in Fig. 6. The two parameters of domain are chosen ad $d_{max} = 100nm$ and $\alpha = \cos(\theta), \theta = 20^o$ based on the experiments. By setting θ equal to zero will transform algorithm to TOP algorithm. As soon as the particle leaves the parallel pushing domain (D), it's center point is fully localized using TOP2 local scanning method. Then parallel pushing direction is turned toward the target again and SPP domain is updated. The baseline (BL_2) should also be updated to a line perpendicular to the new pushing vector PL_{1+1} and passing through the goal position (x_f, y_f) . This process is iterated till particle is at desired distance with respect to the target (is within convergence circle).

3) Localization: In order to push the particle parallel to the initial vector, lateral coordinate of the particle center should be obtained. A prerequisite for this process is having a rough estimation of the particle forward position. In existing algorithms this is reported to be inferred by acquiring topography signal

in an iterative manner [4] [8]. The tip scans both sides of the previous pushing vector until the existence of the particle is observed as a change in topography signal. For SPP algorithm, we have used tip amplitude signal to infer the particle forward position. The planned process is illustrated in Fig. 7. Note that our method for detecting the position where a particle is lost is similar to that in [19] except that, upon detection, SPP initiates only one lateral scan line afterwards to locate the particle instead of full local scan with both horizontal and vertical lines.

At time t_0, the AFM force feedback is turned off and tip starts moving toward the target. As soon as the tip touches the particle at time t_1 , its vibration amplitude drops to zero and the particle starts moving with tip. The tip loses its contact with the particle at time t_2 when it will again start vibrating and the amplitude signal will switch back to the initial value consequently. Estimating the time of contact loss, the forward position of the particle can be calculated from X and Y piezo signals.

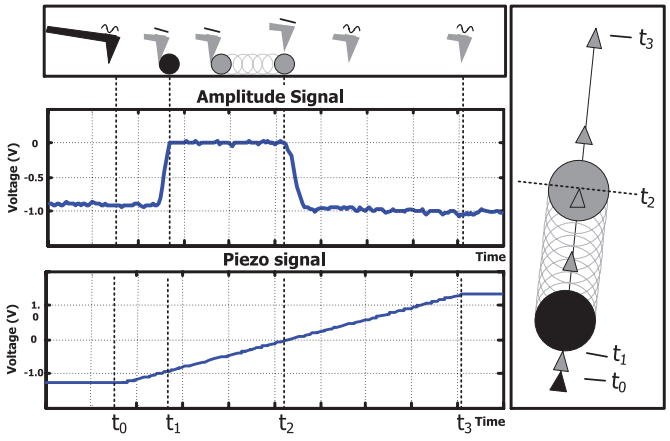

Fig. 7. Amplitude and topography signals during the pushing process.

Following estimation of forward position of the particle, the lateral coordinate will be estimated by acquiring topography signals at the inferred forward position at time . The topography scanning is repeated on the same line till the signal is stable which means:

$$\delta_i = \|O_{i+1} - O_i\| < \delta_{threshold} \tag{8}$$

where the center points O_{i+1} and O_i are given by:

$$O_{i+1} = max(S_{i+1})O_i = max(S_i) \tag{9}$$

Manipulation process using SPP method is summarized in Algorithm 2.

C. Time efficiency of manipulation algorithms

The total number of scan lines and pushing iterations are the key parameters that determine the total manipulation time. Assuming an ideal manipulation process, the time efficiency of introduced algorithms can be compared. The overall time of manipulation can be formulated as following:

$$t_t = t_{init} + \sum_{i=1}^{n}[(z_i + m_i)t_s + t_p] \tag{10}$$

Algorithm 2 Sequential Parallel Pushing

```
i = 0 (iteration counter);
Get the start and target positions (P_0, P_f);
while ||P_i − P_f|| > Δ do
    k = k + 1
    P_0^k = P_i
    while P_i in SPP Domain do
        i = i + 1
        Push along P_0^k − P_f from P_i up to BL^k
        SPP local scan and update P_i(x_i, y_i)
    end while
    Full local scanning and obtain exact P_i(x_i, y_i)
end while
```

where t_{init} is Initialization time, i is the iteration step, n is total number of pushing actions, m is the number of scan lines required to localize particle center at the iteration i, t_s is the time required for each scan line and t_p is the time required for each pushing action. Also, z_i is the number of scan lines required to find the forward position of the particle following each push.

As described earlier, the SPP algorithm utilizes the amplitude signal to locate the forward position of the particles. Hence the value of z_i for SPP algorithm is equal to zero.

In an ideal situation, the minimum and maximum number of required scan lines (m) for TOP1 and TOP2 algorithms can be estimated as $4 \leq m_{TOP1} \leq 6$, $2 \leq m_{TOP2} \leq 3$. Estimating this number for SPP algorithm is not that straightforward as this algorithm may need full scanning process at some steps, but in general the algorithm will find the center line by 2 scan lines. Therefore the expected value of total number of scan lines $(z_i + m_i)$ for SPP algorithm is expected to be the minimum.

This simple estimation of number of scan lines in ideal situation suggests that SPP can reduce the manipulation time to a great extent. It should also be noted that SPP performs a more rigorous local scanning in the required direction comparing to TOP2. Therefore we should expect to see fewer number of pushing iterations for SPP which can further enhance the time efficiency of the manipulation process. In the following section, the performance of described algorithms will be compared empirically and it has been shown that SPP can shorten the manipulation time up to 4 times when compared to two other algorithms.

IV. Algorithmic performance via simulation

To make the intrinsic characteristics of proposed pushing algorithm clear and to compare it with two other introduced algorithms, we model the travel distance for each push via a simple tip-particle contact model [18] based on a set of simple assumptions. The AFM tip is assumed to have a conical shape with semi-aperture angle θ, ending with a sphere of radius R_t. Particles are considered to be sphere with a radius of R_p as shown in Fig. 8. As such, we conduct a geometric analysis of particle movement from which we establish a stochastic model of the particle motion and the localization process. During

pushing, the tip can jump over the particle [8], which should be taken into consideration as part of uncertainty. Sliding happens as well that tip can bring particle to the destination without relative tip-particle motion. Local scan accuracy is assumed to be the same such that only the algorithms' pushing routine will effect the finalized pushing trajectory.

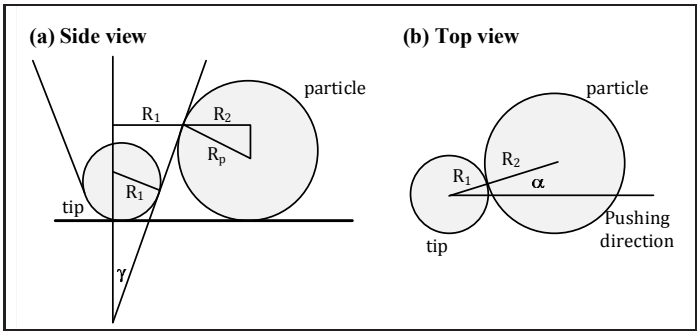

Fig. 8. (a) Side view and (b) top view of a conical AFM tip colliding with a spherical particle.

Based on the tip-particle contact model, the displacement of the particle in forward direction (pushing direction) and lateral direction are given by

$$\Delta y = R\int_{\alpha_0}^{\alpha}\frac{\cos^2\alpha}{\sin\alpha}d\alpha$$
$$= R(\cos\alpha - \cos\alpha_0 + \log\tan\frac{\alpha}{2} - \log\tan\frac{\alpha_0}{2}) \quad (11)$$

$$\Delta x = R\sin\alpha \quad (12)$$

where R is the distance between tip center and particle center on contact plane. α shows the direction particle laterally moves from the pushing line. α_0 is the initial angel of α. If tip does not jump over the particle during pushing process, final angel should be $\pi/2$. So a normal distribution is assumed to simulate the jump-over case between α_0 and $\pi/2$, the contact will be lost on α.

$$\alpha \sim \mathcal{N}(\frac{\pi/2-\alpha_0}{2}, \frac{\pi/2-\alpha_0}{3}) \quad (13)$$

The 2D distribution of the intermediate particle positions simulated by the model are demonstrated in Fig. 9. The pushing lenght is set to be 1000nm and the manipulation process is repeated 100 times. Based on results obtained, two points can be remarked: (1) The distribution of way points in TOPs is concentrated in a small range, e.g. $x \in (-130, 130)$ versus $x \in (-180, 180)$ in SPP. (2) In convergened area near the target, TOPs are uniformly distributed while position SPP mostly lies along the baseline. Thus parallel pushing will result in larger lateral movement than TOP during manipulation, taking time to compensate lateral movement when out of domain while TOPs compensate the lateral movement all the time. Under the same local scan accuracy assumption, the average forward distance is the same.

Fig. 10 shows that with the increase of the angle defined in the domain of SPP, the number of pushing iterations also increases in terms of two cases simulated based on experimental results in Section V. The average number of pushing iterations

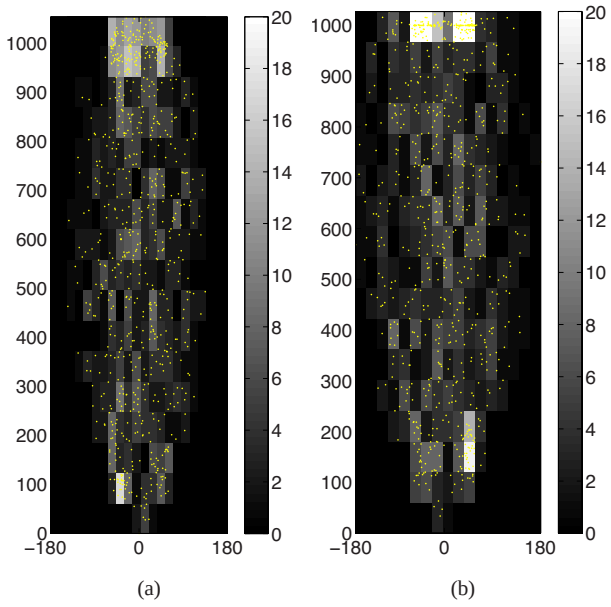

Fig. 9. 2D distribution of intermediate particle positions generated from (a) TOPs (b) SPP

of SPP is always slightly bigger than TOPs. Although it may push longer at early stages of manipulation, but the increased lateral movement slows down the pushing speed resulting in more pushings. The maximum number of pushing iterations of SPP can be up to 8% more than TOP.

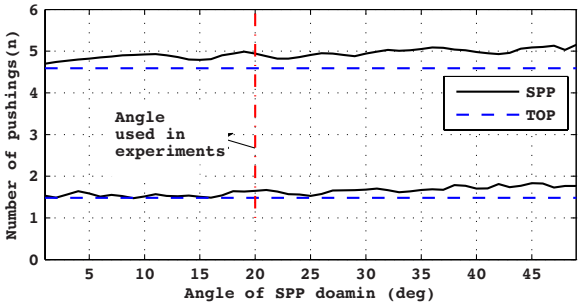

Fig. 10. Simulated number of pushes for SPP and TOPs

In summary, a tip-particle contact model is built to simulate the pushing path of algorithms. The model verifies that although the parallel pushing concept used in SPP results in longer forward travel distance than TOP, the lateral movement causes the overall pushing iterations to be more. The model demonstrates the fact that in terms of number of pushing iterations, SPP always behaves worse than TOP up to 8% because it lacks compensation of the lateral movement. However, this drawback can be made up by its salient advantage of saving half time during local scan. Experimental results confirm this is indeed the case.

V. SPP Implementation and Experimental Comparison

To investigate the performance of the proposed algorithm, it has been implemented on the platform introduced in section

II. The algorithm was applied to Latex nanoparticles with a radius of $50nm$ deposited on a silicon substrate to push them for a distance of $1\mu m$ in air environment. Three different local scanning convergence thresholds of $5nm$, $15nm$ and $25nm$ were used to observe the effect of this parameter on the performance of the algorithm. Each experiment was repeated 10 times and the same procedure was replicated using TOP1 and TOP2 algorithms to compare the performance of different algorithms.

Note that many process parameters that might affect the manipulation process (for instance particle size, substrate roughness and tip condition) which can affect the experimental results. We here report the experiments performed on two different sets of particles. In one set of experiments the particles could be moved to the final position easier and the average number of pushing iterations is less than 2 times. But in the other set manipulation processes required more iterations to successfully deliver the particle to the target position. Besides, for the second set we could not use the local scanning accuracy of $5nm$. The reason was that the local scanning process required relatively many more scan lines to converge and the particle was replaced during this process which resulted in failure of manipulation process eventually. Note that, in our experience, the SPP method has led to very good success rate in manipulating both latex and gold particles on the silicon substrate. In latex nano particle experiments, there is less than 5% chance that a particle cannot be moved due to strong bond between particle and substrate. We have not further investigated the nature of this bond, it could be due to the van der Waals force, capillary force or even chemical bonds. In gold nanoparticle experiments, all gold nanoparticles can be manipulated successfully by SPP.

Time and position data of all intermediate points were recorded during pushing process. The results will be analyzed here in time and position data subsections. Time data shows that SPP takes the minimum manipulation time. From position data it has been inferred that SPP results in a faster forward motion; therefore the forward motion of the particles has been analyzed and two potential reasons for this observation has been discussed.

A. Time data

Details of time data for each algorithm are given in Table I. Evidently, in all experiments SPP has the minimum manipulation time. This time is up to 4 times less that TOP1 algorithm for $\delta = 5nm$. Taking advantage of parallel pushing concept, SPP algorithm requires almost the least number of scan lines in all experiments. The amplitude signal obtained from our experimental platform was not accurate enough. As a result in some cases it took extra scan lines in addition to amplitude signal to locate the particle. This fact has affected the average number of scan lines required by SPP and the experimental results do not fully reflect the advantage of using amplitude signal to estimate forward position of the particle.

The experimental results also show that as δ increases (the local scanning accuracy decreases) the number of required scan lines $(\bar{m})$ decreases. Comparing the average number of pushing iterations $(\bar{n})$, TOP1 has a slightly better performance than TOP2. This is what we expected because of the more thorough local scanning process of TOP1 algorithm. For SPP, $(\bar{n})$ is even better than that for TOP1.

TABLE I
TIME DATA OF TOP AND SPP ALGORITHMS IMPLEMENTED TO PUSH A LATEX NANOPARTICLE FOR A DISTANCE OF $1\mu m$. δ IS THE LOCAL SCANNING THRESHOLD, $\bar{m}$ IS THE AVERAGE NUMBER OF SCAN LINES, $\bar{n}$ IS THE AVERAGE NUMBER OF PUSHING ITERATIONS AND $t_t(s)$ IS THE AVERAGE TOTAL TIME REQUIRED FOR MANIPULATION

		SET I			SET II		
$\delta(nm)$	Algo.	$\bar{m}$	$\bar{n}$	$t_t(s)$	$\bar{m}$	$\bar{n}$	$t_t(s)$
5	TOP1	16.7	4.9	80.0			
	TOP2	9.5	4.9	48.6	-		
	SPP	6.2	4.3	20.2			
15	TOP1	6.5	4.4	33.1	6.7	1.8	39.9
	TOP2	3.5	4.9	23.0	4.6	1.4	26.6
	SPP	3.5	4.3	20.1	4.4	1.3	21.3
25	TOP1	5.1	4.4	27.0	6.2	2.1	41.8
	TOP2	3.3	4.7	21.2	3.8	1.5	22.3
	SPP	3.4	3.8	17.4	3.1	1.5	20.7

B. Position Data

The forward and lateral motion of the particles during manipulation process has been analyzed here.

1) Forward motion: The average forward movement of the particle $(\bar{D})$ is summarized in Table II for all three algorithms and two sets of data. According to the obtained data, SPP has the biggest $\bar{D}$. This means using the SPP pushing algorithm particles approach faster to the goal comparing to TOP1 and TOP2 algorithms.

TABLE II
AVERAGE FORWARD MOVEMENT OF THE PARTICLE

$\delta(nm)$	Algo.	SET I	SET II
5	TOP1	229.3	-
	TOP2	219.2	-
	SPP	250.7	-
15	TOP1	429.5	928
	TOP2	351.1	911
	SPP	492.0	953
25	TOP1	411.9	880
	TOP2	373.0	861
	SPP	414.0	952

The accumulative plot of particle travel way points obtained from all ten experiments of first set is depicted in Fig. 11. It can be seen that for SPP algorithm, fewer way points are located along the path and they are concentrated close to the target position while these points are evenly distributed around the entire pushing course for TOP1 and TOP2 algorithms.

There are two main potential reasons for SPP being faster in forward motion. Firstly, the path that tip travels during local scanning process is shorter and with fewer changes in direction comparing to two other algorithms. This fact has been empirically proven to improve the stability of measurements done by SPP local scanning algorithm in next part. Second of all, by a simple geometrical analysis it has been shown that the parallel pushing method of SPP algorithm justifies the faster forward speed in part.

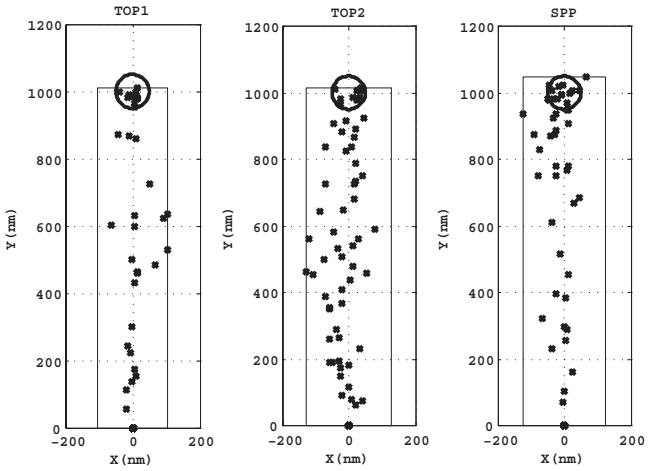

Fig. 11. Accumulative particle travel way points of all ten experiments plotted in $X-Yplane$. The local scanning threshold has been $\delta = 15nm$. Each point shows the position of the particle after a push. The circles illustrate the convergence boundary of manipulation ($r = 50nm$ in this case). The rectangles around points show the observed boundary of travel way points

a) Tip travel path: Pushing the particle closer to the centerline reduces the chance of tip-particle contact loss. Therefore the local scanning accuracy directly affects the length particle travels with the tip. It has been shown here that the SPP local scanning process results in reading more stable signals from AFM. This fact can be a potential reason for SPP to push the particle toward the goal faster.

The paths tip goes through during local scanning algorithms are compared in Fig. 12. For SPP, the scanning lines are repeated trace and retrace lines since it only needs one coordinate of the particle. For TOP, cross-scanning lines are needed to fully localize the particle. The changes in direction of tip path cause disturbances due to piezoelectric actuator nonlinearities [4] which leads to imperfection in local scanning algorithm.

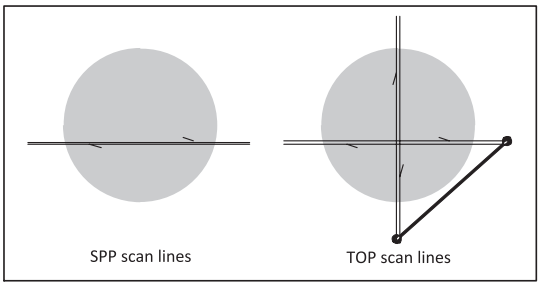

Fig. 12. Tip travel path for SPP and TOP local scanning algorithms. In SPP tip scans over the same line repeatedly while for TOP algorithm the tip performs cross scans.

To make this phenomenon explicit, both SPP and TOP local scanning procedures were used 10 times repeatedly to localize position of 5 different particles. The same experiment was repeated 20 times for each particle. The average standard deviations (σ) of estimated positions from 20 experiments for each of 5 particles are given in Table III.

The results show a smaller σ for SPP algorithm in all 5 cases. This implies that the center coordinate obtained using SPP algorithm is more stable than that obtained by TOP algorithm.

TABLE III
THE STANDARD DEVIATION (σ) OF ESTIMATED POTIONS OF PARTICLES FROM SPP AND TOP LOCAL SCANNING ALGORITHMS

Particle	σ (TOP)	σ (SPP)
1	6.2432	4.6324
2	5.8210	4.2781
3	6.2929	4.0113
4	6.2986	4.4459
5	6.5998	4.0363
6	6.2511	4.2808

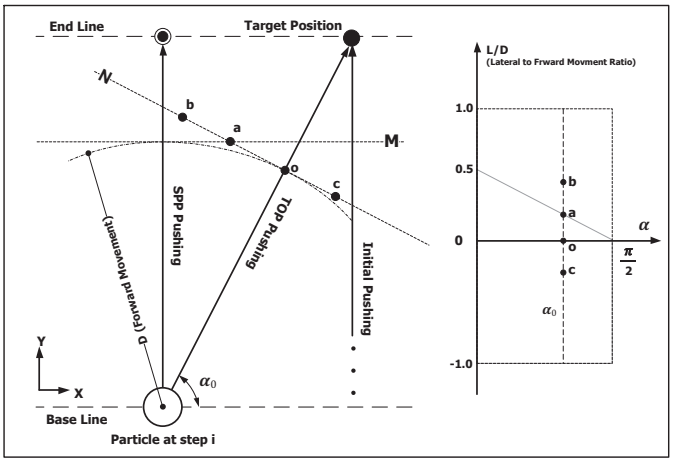

Fig. 13. Geometrical analysis of the pushing process

b) Geometrical analysis: Assuming the same local scanning accuracy and consequently the same forward traveling length (D) for SPP and TOP algorithms, the forward motion of the particle (in direction of initial pushing) is compared in Fig. 13. Pushing the particle from *i*th step it can end up on any point on line M for SPP algorithm or line N for TOP algorithms.

The plot on the right hand compares the distribution of the resulting position versus initial pushing angle, α_0 . It can be concluded that as the particle gets more away from initial pushing line, the possibility of particle ending up closer to the base line is higher in case of SPP. This geometrical description justifies the observation of faster motion of particle in case of SPP in part.

2) Lateral motion: Since the particle is just pushed forward in SPP algorithm, the lateral movement may get large as the particle approaches to the target. To observe how the particle moves laterally while being pushed by SPP algorithm, the particle travel path for all ten experiments of set one where $\delta = 15nm$ is depicted in Fig. 14. The initial position of the particle has been at $P_0(0,0)$ and the target point is $P_f(0,1000)$. Each circle shows a way point of the travel path. The travel paths in Fig. 14 (c),(d) and (e) are the worse cases that can happen in which the lateral movement is always towards one side. But paths of Fig. 14 (a), (b) , (f) and (h) show that it is also possible that lateral movement in consecutive steps compensate each other by forming a zigzag pattern.

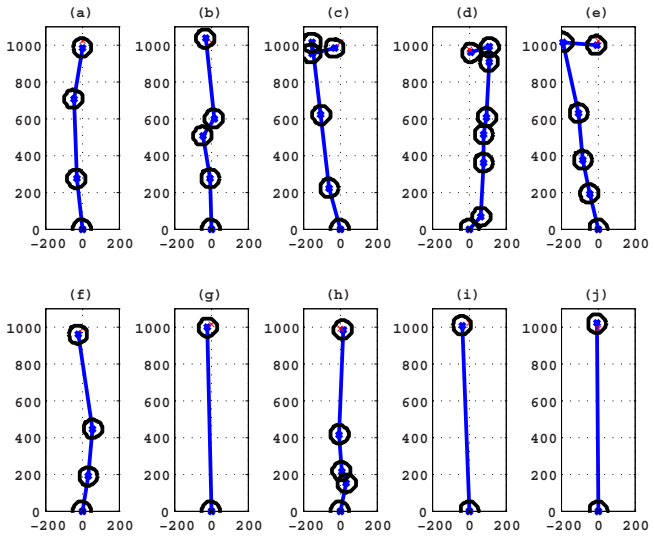

Fig. 14. Particle travel path for all ten experiments of set one for $\delta = 15nm$. The initial position of the particle has been at $P_0(0,0)$ and the target point is $P_f(0,1000)$. The termination condition is $50nm$.

VI. Complex Pattern Fabrication

The SPP method has been used to fabricate different patterns with particles of gold and latex and the particle numbers ranging from twenty to one-hundred seventy. The manipulation process is done in an operation window $2\mu m \times 2\mu m$ but the whole workspace can be significantly larger depending on the size of the pattern. During the whole process, AFM is set to the non-contact mode. The details of manipulation process, including the number and types of particles, pushing time, and workspace size, are given in Table IV. These examples demonstrate that SPP is a viable and efficient method for particle manipulation.

Note that each time a particle is moved to its target position, the sample is re-imaged to update the status of all the particles. The process of re-acquiring an image can take from a couple of minutes up to tens of minutes based on imaging resolution and workspace size. That is what the majority of time in Table IV is spent. We are currently working on automating the whole AFM-based nanopattern fabrication process which by minimizing the time required for rescanning will make the time efficiency of SPP algorithm over other algorithms clearer. Algorithms in [15], [13] can be used for this purpose.

TABLE IV
Specifications of three patterns fabricated using the SPP

Pattern	"iit"	"GOLD"	"IIT CAD LAB"
Workspace size$(\mu m)^2$	4×4	2.8×2.8	10×10
Particle type	Latex	Gold	Latex
Particle diameter (nm)	50	10	50
Number of particles	20	33	170
Time spent (hours)	8	14	40
Manipulation algorithm	SPP	SPP	SPP

VII. Conclusion

A new and more efficient nanomanipulation algorithm, called sequential parallel pushing (SPP), has been presented in this paper. This approach has two salient features: (1) Taking advantage of the parallel pushing idea, SPP algorithm requires just the lateral coordinate of the particle center during pushing. This leads to fewer scan line to locate particle and consequently improves algorithm time efficiency. Besides, real-time vibration amplitude of the cantilever is used to detect when the tip particle contact loss happens and thus the longitudinal coordinate of the particle is determined. (2) The more stable signals acquired during local scanning along with the geometrical nature of parallel pushing gives the SPP algorithm a higher forward speed in moving the particles when compared to TOP algorithms. This leads to SPP to require fewer number of pushing iterations to deliver the particle to the target position.

Although the lateral position of the particle is not compensated during pushing process, it has been shown that the very often zigzag pattern of the particle movement compensates the lateral movement of consecutive steps. The performance of SPP method was experimentally compared with two TOP methods and it was shown that SPP always takes the least manipulation time and it can reduce the manipulation time up to for times comparing to other algorithms. Both latex particles of $50nm$ diameter and gold particles of $10 \sim 15nm$ diameter have been successfully manipulated through this new method to form several different designed patterns. Three of these patterns were demonstrated here.

Ongoing research aims to extend this method to automated manipulation of a large number of particles.

References

[1] G. Binnig, C. F. Quate, and C. Gerber, "Atomic force microscope," *Phys. Rev. Lett.*, vol. 56, no. 9, pp. 930–933, Mar 1986.

[2] L. Zheng and J. Brody, "Electronic manipulation of dna,proteins and nanoparticles for potential circuit assembly," *Biosensors and Bioelectronics*, vol. 20, pp. 606–619, 2004.

[3] A. N. Watkins, J. L. Ingram, and J. D. Jordan, "Single wall carbon nanotube-based structural health sensing materials," *NSTI-Nanotect*, vol. 3, pp. 11–15, 2004.

[4] A. Requicha, D. Arbuckle, B. Mokaberi, and J. Yun, "Algorithms and software for nanomanipulation with atomic force microscopes," *The International Journal of Robotics Research*, vol. 28, no. 4, p. 512, 2009.

[5] A. A. G. Requicha, *Nanotechnology, Volume 3: Information Technology*. Reading,MA: Weinheim, Germany: Wiley-VCH, 2008.

[6] S. Maier, M. Brongersma, P. Kik, S. Meltzer, A. A. G. Requicha, and H. Atwater, "Plasmonics - A Route to Nanoscale Optical Devices," *Advanced Materials*, vol. 13, pp. 1501–1505, 2001.

[7] L. Liu, N. Xi, J. Zhang, G. Li, Y. Wang, and Z. Dong, "System positioning error compensated by local scan in atomic force microscope based nanomanipulation," *Proceedings of the 3rd IEEE Int. Conf. On Nano/Micro Engineered and Molecular Systems*, pp. 1113–1118, 2008.

[8] C. D. Onal, O. Ozcan, and M. Sitti, "Automated 2-d nanoparticle manipulation with an atomic force," *IEEE International Conference on Robotics and Automation*, pp. 1814–1819, 2009.

[9] J. Zhang, N. Xi, G. Li, H. Chan, and U. Wejinya, "Adaptable end effector for atomic force microscopy based nanomanipulation," *IEEE transactions on nanotechnology*, vol. 5, no. 6, p. 628, 2006.

[10] G. Li, N. Xi, M. Yu, and W. K. Fung, "3d nanomanipulation using atomic force microscopy," *Proc. IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3642–3647, 2003.

[11] M. Sitti and H. Hashinoto, "Tele-nanorobotics using atomic force microscopy," in *Proc.of IEEE/RSJ int'l conf. on intelligent robots and system*. Victoria, Canada: Publisher Name, 1998, pp. 1729–1746.

[12] L. Liu, Y. Luo, N. Xi, Y. Wang, J. Zhang, and G. Li, "Sensor referenced real-time videolization of atomic force microscopy for nanomanipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 1, pp. 76–85, 2008.

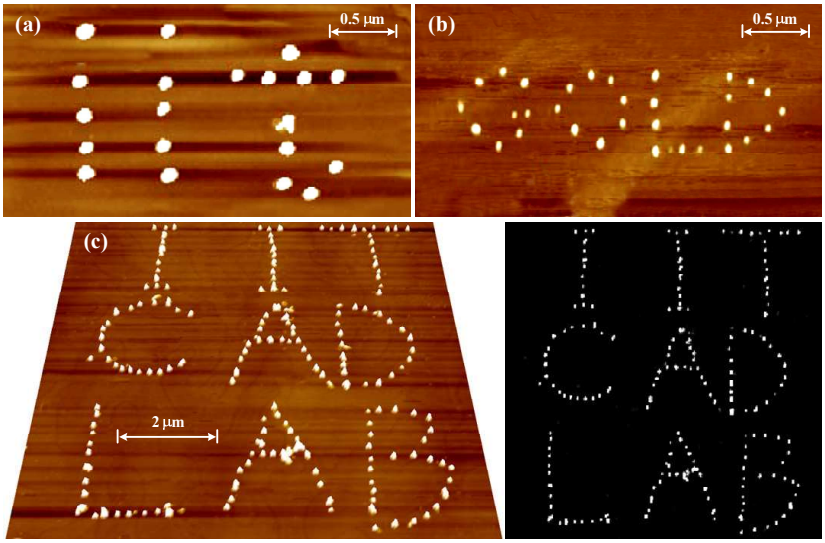

Fig. 15. Complex patterns fabricated using SPP. (a)“iit” pattern composed of 20 latex nanoparticles; (b) “GOLD” pattern composed of 33 gold nanoparticles(c); “IIT CAD LAB” pattern composed of 170 latex nanoparticles.

[13] B. Mokaberi and A. A. G. Requicha, “Compensation of scanner creep and hysteresis for afm nanominipulation,” *IEEE Transactions on Automation Science and Engineering*, vol. 5, pp. 197–206, 2008.

[14] B. A. Mantooth, Z. J. Donhauser, K. F. Kelly, and P. S. Weiss, “Cross-correction image tracking for drift correction and absorbate analysis,” *Review of scientific instruments*, vol. 73, no. 2, pp. 313–317, 2001.

[15] Z. Xu, X. Li, M. A. Sutton, and N. Li, “Drift and spatial distortion elimination in atomic force microscopy images by the digital image correction technique,” *J. Strain Analysis*, vol. 43, pp. 729–743, 2008.

[16] B. Mokaberi and A. A. G. Requicha, “Drift compensation for automatic nanomanipulation with scanning probe microscopes,” *IEEE Transactions on Automation Science and Engineering*, vol. 3, pp. 199–207, 2006.

[17] L. Liu, N. Xi, Y. Wang, Z. Dong, and U. Wejinya, “Landmark Based Sensing and Positioning in Robotic Nano Manipulation,” *Proceedings of the IEEE Int. Conf. On Robotics and Biomimetics*, pp. 37–42, 2008.

[18] A. Rao, E. Gnecco, D. Marchetto, K. Mougin, M. Sch ”onenberger, S. Valeri, and E. Meyer, “The analytical relations between particles and probe trajectories in atomic force microscope nanomanipulation,” *Nanotechnology*, vol. 20, p. 115706, 2009.

[19] H. Xie and S. Régnier, “High-efficiency automated nanomanipulation with parallel imaging/manipulation force microscopy,” *Nanotechnology, IEEE Transactions on*, no. 99, pp. 1–1, 2010.

[20] M. Tomitori and T. Arai, “Tip cleaning and sharpening processes for noncontact atomic force microscope in ultrahigh vacuum,” *Applied surface science*, vol. 140, no. 3-4, pp. 432–438, 1999.