10

Sensor Referenced Real-Time Visual Feedback in Nanorobotic Manipulation and Assembly

Lianqing Liu, Yuechao Wang  
*Shenyang Institute of Automation, China*

Guangyong Li  
*University of Pittsburgh*

Ning Xi  
*Michigan State University*

10.1 Limitation of Augmented Reality System

Although the augmented reality system has greatly improved the efficiency and effectiveness of AFM based nanomanipulation through providing the real-time force and visual feedback to the operator, there are still several shortcomings that stand as roadblocks hindering its wide application.

The first problem is the poor reliability of the real-time visual feedback. Although the augmented reality system can provide real-time visual feedback to the operator, this kind of real-time visual display is not the true manipulation results, but a graphical simulation based on the behavior models.
Since any model may have differences with the true environment, especially in the nanoworld, microforces such as surface tension, van der Waals force, capillary force, and so on are very complex and interact with each other. It is difficult to get the accurate behavior model of nano-objects, which in turn results in a faulty visual display and a failed manipulation. As shown in Figure 10.1(a), two nanoparticles are manipulated along the arrow direction to form a triangle pattern. Figure 10.1(b) is the manipulated results displayed in the real-time visual feedback and it shows that the triangle pattern is formed very well. Figure 10.1(c) is the true manipulation result obtained from a new image scan. Through these pictures we can see that the real-time visual display cannot match the true manipulation result. Actually, this phenomenon is the common case in nanomanipulation with augmented reality systems especially for nanoparticles. Due to the complex interaction force among AFM tip, object, and substrate, a lot of uncertainties lead the tip to slip aside or slip over the objects easily during manipulation. This situation is extremely difficult to predict with a model. The modeling error leads to a faulty visual display and a failed manipulation. Thus it is very urgent to find a way to solve this problem and improve the reliability of the real-time visual feedback.

The second problem is the positioning error caused by the random drift. The random drift is caused by many factors such as the contraction and expansion of the mechanical system, temperature variety, humidity change, and so on [1–3]. Due to the random drift, nanomanipulation can be likened to using AFM to manipulate a nano-object that is in a conveyor. Random drift accompanies the manipulation process all the time. Figure 10.2 shows...
a sequential image scan with a time interval of 5 minutes. Even though the scanning parameters keep constant, there is still a obvious drift between the AFM tip and the sample. Since all the AFM based nanomanipulation is performed based on the previously scanned image, the random drift will cause a position error between the current manipulation coordinate and the true environment. Furthermore, if the random drift is larger than the size of the nano-object, the AFM tip may not be able to touch the object at all during manipulation. As shown in Figure 10.3, the dashed line in (b) shows the desired push trajectory of the nanoparticle. The visual feedback shows that the particle has been pushed away by the AFM tip. But due to the random drift, there was a positioning error between the display coordinate and the manipulation coordinate. Figure 10.3(c) is the true manipulation result from a new image scan. Comparing (b) with (c), we can see there is a rightwards drift during manipulation, which causes the tip to shave the particle instead of pushing it away. Thus it is urgent to compensate the positioning error caused
from random drift. The traditional ways to overcome this problem are inconvenient and inefficient, such as running image scan first for a couple hours before manipulation to eliminate the influence caused from the mechanical contraction and expansion, and controlling the manipulation environment strictly to get rid of the influence from temperature and humidity. Some research work has also been carried out to handle these problems by estimation and compensation [3, 4]. A Kalman filter is developed [3] to estimate and compensate the error caused by the thermal drift. Since this method is a model based compensation, whether the compensation is successful or not lies on the degree of the model’s accuracy; it is not easy and also takes time to obtain the accurate model parameters. A neural network is developed to compensate thermal drift in [4]. But many AFM images are required for training the network offline before manipulation. Even though this method works well, it is still not convenient because it takes a long time to get several images to train the network, and it is also a model based compensation method and has the similar problems as the Kalman filter based estimation. Moreover, these two methods only consider the influence caused by thermal drift, although the error caused by humidity change or other uncertainties is relatively small compared with the one caused by temperature, it is better to take care all of them when manipulation is performed in an arbitrary condition.

From the above discussion we can see neither the poor reliability of visual display positioning error caused from random drift has been resolved very well. Any method that can solve these problems once and for all with high efficiency and effectiveness will lead to another breakthrough in AFM based nanomanipulation. In this chapter, based on our previously developed augmented reality system, a real-time fault detection and correction (RFDC) method will be proposed to approach this goal. The basic idea is using the real-time sensor information such as the interaction force as a reference to check the validity of the visual feedback through a Kalman filter. Once the fault display is detected, local scan, which can determine the position of the nano-object with several scanning lines, is performed to get the true manipulation result, then the display error is removed by updating the visual display with the true manipulation result. In addition, through monitoring the relative movement of the sample feature to the AFM tip with local scan, the positioning error caused from random drift can also be compensated. Since only several lines are scanned, local scan can be finished in several tens of milliseconds and is transparent to the operator. Therefore, the position errors caused by random drift or inaccurate models can be compensated without interrupting the operator’s manipulation process. By combining the RFDC method with our formerly developed augmented reality system, the reality of the
visual display is greatly improved. The experimental results of using the new developed system for manipulating some nanomaterials such as nanoparticles and nanorods show that not only random drift can be effectively detected and compensated, but the mismatch between the visual display and the true environments during manipulation can be real-time corrected, which in turn leads to a significantly improved efficiency and effectiveness of AFM based nanomanipulation.

10.2 The Augmented Reality System with Real-Time Fault Detection and Correction

On the base of the augmented reality system developed in the last chapter, a new RFDC enhanced system is built as shown in Figure 10.4. Compared with the original architecture in Figure 9.3 of the last chapter, two modules, Kalman filter and local scan, are newly integrated into the system. The Kalman filter is used to monitor the validity of the visual feedback; local scan is used to correct the display error once the faulty display is detected. Since interaction force is the key parameter for Kalman filter based fault detection, its accurate measurement is pivotal to whether the fault display can be detected or not. Thus, to achieve a better fault detection performance, active probe technology developed in the last chapter is used to obtain a high sensitive interaction forces.

![Figure 10.4](image.png)
Comparing the original architecture in Figure 9.3 and the new developed architecture in Figure 10.4, we also can see that in the original augmented reality system, the movie-like visual display is calculated based on the behavior models. There is not any feedback to check the validity of the visual display. Since any model may have differences with the true environment changes, the visual display may not be consistent with the true environment changes. As a result, a nanomanipulation may fail due to a fault display. With the new developed system architecture, Kalman filter and local scan schemes are newly implemented into the system to detect and correct the faulty visual display. That is, the visual feedback is generated and displayed in a close-loop way.

![Flowchart for the real-time fault detection and correction](image-url)
The flow chart of the new developed system is shown in Figure 10.5. In this scheme, an initial position error compensation is first performed to make the manipulation coordinate identically with the true environment. Since random drift happened between the whole image and AFM tip, the current random drift for all objects can be eliminated through manipulation coordinate compensation. After manipulation starts, a high sensitive interaction force is obtained from the active probe controller, based on the mismatch between the visual display and the true environment, which can be detected online by a Kalman filter. As the fault display is detected, an optimized scan pattern is also generated under the assistance of the Kalman filter, which starts a local scan to obtain the true manipulation results in a minimum time. Finally, the true position of nano-object is updated to the visual feedback interface to correct the faulty display. Since only several lines are scanned, local scan can be finished in tens of milliseconds. The visual feedback can be updated with a maxim frame frequency 20 Hz, and is very smooth. Thus, the whole process is transparent to the operator without any disturbing. Once the local scan is triggered, the feedback force will remain unchanged as the force value in the moment before local scan. To avoid the abrupt force change when switching back to the manipulation mode, a time sliding window is adopted to guarantee the force be changed in a smooth way.

From the above introduction we can see, in this new designed system, the visual feedback is not only updated relying on the behavior models, but also based on the real-time data from the RFDC process. In this way, the reality of the visual feedback in the virtual reality interface is greatly improved, which in turn facilitates the AFM based nanomanipulation.

### 10.3 Real-Time Random Drift Compensation with Local Scan

In the AFM based nanomanipulation, finishing a manipulation task often includes three steps: identifying objects, manipulation, and verifying the manipulation result with a new AFM image scan. During the process of identifying objects, the objects on the previously captured AFM image are identified and their positions are labeled. But due to the random drift, there will be a position error between the visual display coordinate and the true environment, thus the labeled positions may not represent the actual positions. This coordinate error should be compensated prior to manipulation starting, otherwise alarm signals will be triggered by the Kalman filter at the beginning of the manipulation for each object. Since the random drift happens between the whole image and the AFM tip, the initial coordinate error for each object
should be the same. Therefore, it can be compensated by detecting the random drift from any object in the AFM image. The value of the random drift can be calculated through the object’s real position and labeled position.

The real position can be located by locally scanning several lines across the object. Figure 10.6 shows the mechanism of detecting position error during nano-particle and nanorod manipulation, respectively. The solid circle $O$ and rectangular $A$ represent the particle and the rod position on the AFM image before manipulation. Since there is random drift, the particle is actually at the position of the dashed circle $O_a$, and the rod is actually at the position of the dashed rectangular $B$. Different scan patterns are designed to find their real positions.

The location of a nanoparticle can be determined by its center position and radius. Since the radius of the nanoparticle has been obtained in the process of identifying objects before the manipulation, the local scan only needs to relocate the actual center of the nanoparticle. To get the actual center position of the particle, the local scan needs to scan at least two lines, one or more horizontal lines and one vertical line. First, as shown in the left of Figure 10.6, AFM scans along line $L_0$, which passes through point $O$, the displayed center of the particle on the image. If the particle is not found, then the scanning line moves up and down alternatively by a distance of $3R/2$ until the

Figure 10.6 The mechanism of local scan to detect the initial position error [8] (© IEEE 2008.)
A vertical line $V$, which goes through the midpoint between $P_1$ and $P_2$ and is perpendicular to the previous scanning line, is scanned to find the center of the particle. The vertical scanning line has two intersection points with the boundary of the particle, $Q_1$ and $Q_2$. The center of the nanoparticle $O_a$ is located at the midpoint between $Q_1$ and $Q_2$. The length of the scanning line $L$ is determined by the maximum random drift such that $L > 2(R + r_{max})$, where $r_{max}$ is the maximum random drift distance and is determined through experiments.

The similar mechanism can be used to find the actual location of the nano-rod. The location of a nanorod is represented by its width and two ends in local scan for convenience. The initial displayed rod width and ends are identified from a previously captured AFM image before the manipulation. Due to the random drift, the actual position is not exactly at $A$. The dashed rectangle $B$ represents the real rod position. To find this real position, local scan needs to scan at least three lines. First, as shown in the right of Figure 10.6, the AFM scans along line $L'_0$, which passes through the displayed center of the rod and perpendicular to the rod orientation. If the rod is not found, then the scanning line moves up and down alternatively by moving the scan line up and down by a distance of $L/4$ until two scan lines can locate the rod, where $L$ is the length of the rod. Each of the two scan lines has two intersection points with the boundary of the rod. For example $P'_{1}$ and $P'_{2}$ for $L'_0$, $P'_3$ and $P'_4$ for $L'_1$. Another line $V'''$, which goes from the midpoint between $P'_{1}$ and $P'_{2}$ to the midpoint between $P'_3$ and $P'_4$, is scanned to locate the two actual ends of the rod. The scanning line $V'''$ has two intersection points with the boundary of the rod, $Q'_{1}$ and $Q'_{2}$, which are the actual ends of the rod. The actual rod center will locate at the midpoint between $Q'_{1}$ and $Q'_{2}$. The orientation of the rod will be along the direction $Q'_{1}Q'_{2}$. The length of the scanning line $L'_0$ is determined by the maximum random drift such that $L_0 > d + r_{max}$, where $d$ is the width of the rod. The length of the scanning line $V'''$ is $2L$.

After the actual position is obtained, the random drift can be easily calculated through the difference between the displayed center position and the actual center position, such as

$$
\begin{align*}
    d_x &= O'^x_x - O^d_x \\
    d_y &= O'^y_y - O^d_y
\end{align*}
$$

(10.1)
where $d_x$ and $d_y$ delegate the random drift values in X and Y direction, respectively, $O'$ delegates the real position of the object center obtained through local scan, $O''$ represents the initial labeled position in the previous captured AFM image. For the nanorod, the rotation caused by the random drift can also be calculated from the actual orientation $Q_2Q_1$ and the initial orientation $W_2W_1$, such as

$$\alpha = \arctan W_2W_1 - \arctan Q_2Q_1$$

(10.2)

where $\alpha$ is the rotation angle caused by the random drift.

Since random drift happened between the whole image and the AFM tip, the position error from the local scan existed between each object and the AFM tip. Thus, instead of updating the visual display image with the actual position of the single object, the random drift was compensated by multiplying the motion command of the AFM tip with a transform matrix, such as

$$
\begin{bmatrix}
P_x' \\
P_y'
\end{bmatrix} =
\begin{bmatrix}
\cos \alpha & \sin \alpha & d_x \\
-\sin \alpha & \cos \alpha & d_y
\end{bmatrix}
\begin{bmatrix}
P_x \\
P_y \\
1
\end{bmatrix}
$$

(10.3)

where $P_x$ and $P_y$ are the position command generated in the initial display coordinate. $P_x'$ and $P_y'$ are the true position command sent to the AFM tip after the random drift compensation. By intermittently executing this process before each step of manipulation, the position error caused by the random drift can be effectively eliminated.

Here we give two kinds of local scan patterns for determining the true positions of nanoparticle and nanorod, respectively. Especially, once the true position of a nanorod is determined, based on its displayed positions in the visual feedback, the translation drift and the rotation drift can be compensated simultaneously. But once the nanorod is too slim to form two intersections with the first two scanning lines, the orientation of the nanorod cannot be determined. In this case we have to change the local scan pattern for determining the rotation drift. In our previous work [5], a local scan pattern for a DNA molecule was developed, and the position and orientation of the DNA molecule can be determined uniquely. Then we can use graphical matching method to calculate both the translation and rotation drift in a similar way as used for nanorod. Since for most of the nanorod, the intersections are possible to be detected, we won’t address this extreme situation too much.
10.4 On-Line Fault Detection and Correction

As introduced in Section 10.1, the movie-like visual feedback in the visual reality environment does not show the real changes of the nano-environment, but rather is calculated from the behavior models. Since any model may have differences with the true environment changes, the modeling errors will result in a fault display and lead to a failed manipulation. To solve this problem, a Kalman filter and local scan scheme are developed in this section to detect and correct the fault display in real-time. Here the mechanism of the local scan is similar with the one in Section 10.3, but the scan pattern is optimized based on the estimation of the Kalman filter.

10.4.1 Kalman Filter Based Fault Display Detection

The object’s dynamic model is necessary to detect the error between the visual feedback and the real time changes in the true environment by using Kalman filtering techniques. To make our point more clear, an example of manipulating a nanoparticle will be used to explain this method. This method can be expanded to other tapes’ objects by adopting corresponding dynamic models. To simplify the analysis, the motion of tip and objects is decomposed into two directions X and Y. The dynamic model of the nanoparticle in X direction during manipulation can be expressed as

\[
\ddot{x} + k\dot{x} = \lambda F_x
\]  

(10.4)

Here \(x\) is the relative displacement of the particle to the pushing start point in X direction, \(F_x\) is the interaction force between the particle and the AFM tip measured from the active probe controller, \(k\) is the damping coefficient of the tip-object dynamic system, \(\lambda\) is the proportional gain for adjusting the sensed interaction force. \(k\) and \(\lambda\) can be determined through experiments. The similar equation can be written for Y direction. The state space representation of the nanoparticle in X direction is

\[
\begin{align*}
\dot{x} & = v_x \\
\dot{v}_x & = \lambda F_x - kv_x
\end{align*}
\]  

(10.5)

Here \(x\), \(v_x\) are the displacement and velocity of the particle in X direction, respectively.
The corresponding discrete-time equations in X and Y direction with a sampling period $T$ can be retrieved from (10.5) as

$$
\begin{align*}
X_{k+1} &= AX_k + BU_k \\
Z_k &= CX_k
\end{align*}
$$

where $X_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$, $U_k = [0,0, f^x_k, f^y_k]^T$, here $f^x_k$ and $f^y_k$ is the interaction force in X and Y direction, respectively. $Z_k$ is the states measurement from the visual display, the observation matrixes $C$ is a 4-by-4 unit matrix, state transition matrix $A$ and control input matrix $B$ are given as

$$
A = 
\begin{bmatrix}
1 & 0 & \frac{1}{k} (1 - e^{-kT}) & 0 \\
0 & 1 & 0 & \frac{1}{k} (1 - e^{-kT}) \\
0 & 0 & e^{-kT} & 0 \\
0 & 0 & 0 & e^{-kT}
\end{bmatrix}
$$

$$
B = 
\begin{bmatrix}
0 & 0 & \frac{\lambda}{k} (T - \frac{1}{k} + \frac{1}{k} e^{-kT}) & 0 \\
0 & 0 & 0 & \frac{\lambda}{k} (T - \frac{1}{k} + \frac{1}{k} e^{-kT}) \\
0 & 0 & \frac{\lambda}{k} (1 - e^{-kT}) & 0 \\
0 & 0 & 0 & \frac{\lambda}{k} (1 - e^{-kT})
\end{bmatrix}
$$

With the dynamic model of the object, it is now possible to test if the visual display agrees with the real time changes in the nano-environment or not. This is done by monitoring the state with a Kalman filter. An estimate $X_{k|k-1}$ and an associated covariance matrix $P_{k|k}$ are real-time calculated during manipulation. In the ideal case, without error the residual $e_k$ from the Kalman filter should be Gaussian and white. $e_k$ is defined as

$$
e_k = Z_k - CX_{k|k-1}
$$
When the visual display does not agree with the estimation, the innovation $e_k$ will not be Gaussian or white. The value of $e_k^T e_k$ will have an abrupt increase. $e_k^T e_k$ can be directly used to judge whether there is a display error by setting a threshold on it. But this way is less robust and false alarm signals are frequently triggered. To improve the robustness of the fault detection, the Mahalanobis distance of the innovation $e_k$ is calculated and used as a measurement to determine whether there is a fault display [9]. The Mahalanobis distance is defined as

$$M_t = e^T R_e^{-1} e$$

(10.8)

where $R_e$ is the associated covariance of $e_k$ and is defined as

$$R_e = E[(Z_k - CX_{k|k-1})(Z_k - CX_{k|k-1})^T]$$

(10.9)

Substituting $Y_k = CX_k + V_k$ into (10.9) (here $V_k$ is the measurement noise with the covariance $R$ and independent of $X_k$) (10.9) can be rewritten as

$$R_e = P_{k|k} + R$$

(10.10)

By substituting (10.10) into (10.8), the Mahalanobis distance ($M_t$) can be real-time obtained during manipulation. A threshold on $M_t$ is set to give an alarm signal when a fault happens. Once the visual display loses its match with the real environment changes, $M_t$ will have an abrupt increase and overshoot the threshold. An alarm signal will be triggered for reporting the display error to the system.

### 10.4.2 Fault Display On-Line Correction

Once it is detected that the visual display does not match the real-time changes in the nano-environment, a local scan is needed to find the real manipulation result and correct this error. The local scan mechanism is similar to the one in Section 10.3. However, with the estimation of the Kalman filter, the scan pattern can be optimized to minimize the local scan time. The optimized scan pattern should first scan the area with the highest probability that the object can be found. Since the Kalman filter can provide an optimal estimation of the true position, the scan pattern is designed as shown in Figure 10.7. The solid circle $O$ represents the particle position in the visual display interface. The dotted circle $O_e$ represents the estimation position from the
Kalman filter. In an ideal case, the particle’s real position should agree with the estimated position the Kalman filter. Due to environment noise or dynamic modeling errors, the real position of the particle may not be at point $O_e$, but at point $O_r$. The scan pattern will first go along line $L_0$, which passes through the displayed center $O$ and the estimating center $O_e$. If the particle is not found, then the scanning line moves up and down along the direction perpendicular to line $L_0$ alternatively by a distance of $3/2R$ until the particle is found, where $R$ is the radius of the particle. Once the particle is found, the scanning line forms two intersection points with the boundary of the particle, $P_1$ and $P_2$. Another line $V$, which goes through the midpoint between $P_1$ and $P_2$ and is perpendicular to the previous scanning line, is scanned to get the actual center $O_r$ of the particle. The last scan line has two intersection points with the boundary of the particle, $Q_1$ and $Q_2$. The final actual center of the nanoparticle $O_r$ is located at the midpoint between $Q_1$ and $Q_2$. By first scanning the estimated position of the Kalman filter, the object can often be found by the first scan line, which shortens the local scan time, as well as smooths the visual feedback.

10.5 Implementation and Experimental Results

The experiments were performed in an arbitrary condition. The experimental system mainly consists of a Bioscope AFM (Vecco Inc., Santa Barbara, CA)
with a scanner which has a maximum XY scan range of 90 μm × 90 μm and a Z range of 5 μm. Some peripheral devices including a haptic device (Phantom, Sensable Company, Woburn, MA), a Multifunction Data Acquisition (DAQ) cards NI PCI-6036E (National Instruments), and three computers.

The system setup is similar to the one in Figure 9.22. During manipulations, the operator uses the haptic device to input the tip position command and feel the real-time interaction force between the tip and the nano-objects. The real-time visual display provides the operator a locally updated dynamic AFM image that is retrieved from the behavior model and enhanced by the RFDC method. The real-time control module runs in a Linux computer mounted with a DAQ card, which an active probe controller, a Kalman filter, and a local scan controller. The whole system works as introduced in Section 10.2. To speed up local scan, the drive signals of the AFM scanner for local scan are directly outputted to the AFM controller through the DAQ card by the Linux PC. The value of the voltage applied to the piezo tube in Z direction, which represents the topography of the scanned surface, is recorded during local scan. With the topography data along the scanning line, the actual position of nano-objects can be determined as described in Section 10.3. By this way, the position error caused by random drift and model errors can be real-time detected and corrected.

10.5.1 Random Drift Compensation

The random drift generates rotation and translational errors between the manipulation coordinate and the true environment. This position error can be detected and compensated through a quick local scan before manipulation starts. As shown in Figure 10.8, a nanorod was deposited on the polycarbonate surface. Due to the random drift, the displayed location may not be the real location. A local scan was performed before manipulation to get the transform matrix for random drift compensation. The local scan pattern is shown as in Figure 10.9(a). The topography information along the scanning lines is shown in Figure 10.9(b). Here the different max height among the three scan lines is due to the data process for removing the offset and eliminating the tilt, but the relative height along the scanning line keeps constant, and the intersection points between the scanning line and the object can be identified without any influence. From the local scan result, the translational error and the rotation angle can be calculated with the method in Section 10.3. In this experiment, the rotation angle \( \alpha \) caused by the random drift is –0.07 radian, the translational drift are 72.4 nm in X-axis and 106.7 nm in
Y-axis, respectively. Then we substitute these parameters into the (10.3) and can get the following compensation equation:

\[
\begin{bmatrix}
P_x' \\
P_y' \\
1
\end{bmatrix} =
\begin{bmatrix}
0.999 & -0.001 & 72.4 \\
0.001 & 0.999 & 106.7 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
P_x \\
P_y \\
1
\end{bmatrix} \tag{10.11}
\]

Each motion command generated in the visual display image should be compensated by (10.11) before being sent to the AFM controller. The total time cost by local scan in this experiment is less than 80 ms, which is fast enough such that the whole process can be finished without being perceived by the operator. By intermittently repeating this process, the random drift error can be effectively eliminated on-line. From this experiment we also can see that the random drift mainly causes the translational error. The rotation error normally can be neglected during manipulation.

### 10.5.2 Modeling Error Detection and Correction

In this section, nanoparticles are manipulated to validate the effectiveness and efficiency of the RFDC method. A quick local scan has been performed to compensate the initial random drift before manipulation starts. Figure 10.10(a) shows a latex particle with diameter around 350 nm that was pushed along the arrow direction. Figure 10.11(a) shows the real-time interaction...
Figure 10.9  Local scan pattern and result; (a) the local scan pattern for nanorod; (b) the topography information along the scanning lines [8] (© IEEE 2008.)
force between the AFM tip and the particle during manipulation. Here the interaction force is calculated from the cantilever control signal and the horizontal output of the photodiode detector based on the models in [6] and [7]. The parameter $k$ and $l$ in (10.4) are with value 2.2 and 1, respectively. Based on the interaction force, the behavior state of the particle estimated from the Kalman filter is shown as the dashed line in Figure 10.11(b). The solid line in Figure 10.11(b) is the measured behavior state from the visual display interface. The corresponding residual from the Kalman filter is shown in Figure 10.11(c). Before $t = 5.8s$, the behavior state in the visual display agrees with the Kalman filter estimation. The residual value keeps near zero and is less than the threshold. After $t = 5.8s$, the residual has an abrupt increase and overshoots the threshold, which means the visual display does not agree with the true environment changes any more, and a fault display is detected. This phenomenon agrees with the change of the interaction force as shown in Figure 10.11(a), at $t = 5.8s$, there is a sudden drop of the interaction force, which means the tip may slip away from the particle. As a result, the estimated displacement from Kalman filter does not agree with the visual display any longer after $t = 5.8s$. Therefore, the residual sharply increased and an alarm signal is triggered for reporting this error to the system.

After the alarm signal was triggered, an optimized scan pattern was generated based on the estimation of the Kalman filter as shown in Figure

![Figure 10.10](image) Manipulating latex particles (diameter 350 nm) on polycarbonate surface with an operation range of 12 $\mu$m. (a) Pushing a particle along the arrow direction. (b) Manipulation result in the visual display. The right picture is the zoom-in image of the rectangle area in (b). The indent A represents the false display position. B represents the true position which is obtained on-line by local scan.
The topography information along the scanning lines is shown in Figure 10.12(b). From the topography information, the actual position of the nanoparticle can be identified and updated to the visual interface. As shown in Figure 10.10(b), the indent area A represents the particle location in the visual display. Area B shows the real particle position detected from the local scan. The difference between position A and position B demonstrates that the fault display caused by the modeling error or other uncertainties can be effectively detected by the Kalman filter and corrected through the local scan process.

As discussed in the first section of this chapter, the visual feedback losing match with the true environment is a common case in AFM based...
Figure 10.12  Local scan pattern and result. (a) The local scan pattern for nano-particles. (b) The topography information along the searching lines [8] (© IEEE 2008.)
manipulation. Figure 10.1 and Figure 10.10 both demonstrated this case. This situation is difficult to predict with behavior models, which often result in a fault visual feedback and leads to a failed manipulation. Without the RFDC method, a new image scan is repeatedly needed to correct the display errors and verify the manipulation result for building some complicated nanostructures, which is time-consuming and inefficient. After incorporating the RFDC method, the AFM based nanomanipulation is greatly increased. Not only can the fault display be real-time detected, but also it can be corrected on-line without interrupting the manipulation. The operator can immediately know the result after each step of manipulation. Figure 10.13 shows different patterns built by manipulating nanoparticles under the assistance of RFDC. The real-time AFM image is displayed in the augmented reality interface as shown in Figure 10.13(I-b), (II-b). A new scanned image after manipulation is shown in Figure 10.13(I-c), (II-c). The two sets of pictures

![Figure 10.13](image)

**Figure 10.13** Manipulating nanoparticles with diameter 350 nm on polycarbonate surface. The operation range is 12 μm × 12 μm in I and 13 μm × 13 μm in II. (a) Image of latex particles on a polycarbonate surface before manipulation. (b) The image of manipulation result displayed in the real-time visual feedback interface. (c) A new scanned AFM image after final manipulation [8] (© IEEE 2008.)
show that the final results match the display well in the visual feedback interface. The reason that the diameter of the particle in Figure 10.13(II-c) looks bigger than Figure 10.13(II-a) is due to the tip effect since the tip has been contaminated during manipulation. Figure 10.14 shows another example of a nanorod with diameter 170 nm being manipulated under the assistance of RFDC; the display result matches the new image scan too. All of these experiments were performed continuously in a couple of minutes. Without the assistance of RFDC it often took several tens of minutes to form a pattern as shown in Figure 10.13 due to repetitively scanning new AFM images. From this experimental study, it can be seen that assembly of nanostructures using an RFDC enhanced augmented reality system becomes very straightforward and highly efficient.

### 10.6 Conclusion

It is well known that the main difficulty of nanomanipulation using AFM is the lack of real-time visual feedback. Although this problem is partly resolved through the augmented reality technology, the problems of random drift and modeling errors in the manipulation still stand as a roadblocks that hinder the efficiency and effectiveness of AFM based nanomanipulation. As a result, new AFM image scan is repeatedly needed to finish the manipulation task successfully. This chapter introduced a RFDC method to solve this
problem. By adopting active probe technology, the AFM tip works as an end effector as well as a high sensitive force sensor during manipulation. With the real-time interaction force as a reference, the validity of the visual display is monitored on-line. The random drift and the modeling errors can also be real-time compensated without interrupting manipulation through local scan. The manipulation is always carried out with the visual feedback of the true environment changes, through which the efficiency and effectiveness of the AFM based nanomanipulation are greatly improved.

The developed method also provides a facilitated “bottom-up” technology for nanomanufacture. Since building nonsymmetric nanostructures using a “top-down” method has been hindered by the limitations of the optical lithography, it is necessary to develop an efficient technology for building irregular and nonsymmetric nanostructures. The developed method provides such a facilitated technology and has a lot of potential applications in manufacturing new revolutionary products under nanoscale, which will bring breakthroughs to the modern society in the future. In addition, the large immunity to random drift and modeling error of this method also facilitates the automatic nanomanipulation, which in turn makes it possible to build more complex nanostructures.

References


