Microsc. Microanal. 18, 612–620, 2012
doi:10.1017/S1431927612000220

Microscopy AND Microanalysis

An Improved Visual Tracking Method in Scanning Electron Microscope

Changhai Ru,[1,2,*] Yong Zhang,[3] Haibo Huang,[2] and Tao Chen[2]

[1]*College of Automation, Harbin Engineering University, Harbin 150001, China*
[2]*Robotics and Microsystems Center, Soochow University, Jiangsu 215021, China*
[3]*Advanced Micro and Nanosystems Laboratory, University of Toronto, Ontario M5S3G8, Canada*

Abstract: Since their invention, nanomanipulation systems in scanning electron microscopes (SEMs) have provided researchers with an increasing ability to interact with objects at the nanoscale. However, most nanomanipulators that are capable of generating nanometer displacement operate in an open-loop without suitable feedback mechanisms. In this article, a robust and effective tracking framework for visual servoing applications is presented inside an SEM to achieve more precise tracking manipulation and measurement. A subpixel template matching tracking algorithm based on contour models in the SEM has been developed to improve the tracking accuracy. A feed-forward controller is integrated into the control system to improve the response time. Experimental results demonstrate that a subpixel tracking accuracy is realized. Furthermore, the robustness against clutter can be achieved even in a challenging tracking environment.

Key words: nanomanipulation, visual servo, SEM, subpixel tracking

Introduction

Since nanomanipulation was proposed with a scanning tunneling microscope by Eigler and Schweizer (1990), many nanomanipulation schemes and systems have been developed and nanomanipulation has become a common tool of nanotechnology. During nanomanipulation, sizes of the target objects are in the range from submillimeter down to a few nanometers. Significant progress in nanomanipulation, such as the manipulation of nanotubes, nanowires, and nanoparticles, has been achieved by the use of an atomic force microscope (AFM) (Falvo et al., 1999; Rubio et al., 2005; Dietzel et al., 2007; Prior et al., 2007). However, AFM cannot provide manipulation with real-time visual feedback during the manipulation because AFM images can only be acquired before or after the manipulation. Thus the manipulation of nano objects by an AFM cantilever has to be conducted in a "blind" manner and is inefficient and time-consuming. The integration of a nanorobotic station into the vacuum chamber of a scanning electron microscope (SEM) can solve this issue by the processing of SEM images (Sievers & Fatikow, 2006; Fukuda et al., 2009; Kratochvil et al., 2009; Suga et al., 2009). To perform more kinematically complex manipulation or free the researchers from the laborious manual work, better manipulation strategies are needed. However, until now most nanomanipulation research has been focused on manual strategies for characterizing nanostructures (Sierra et al., 2005). One major challenge of performing visual servo tasks inside an SEM is balancing the needs of image quality. The raster scanning used to create an SEM image necessitates lower frame rates than those available with optical cameras. The drawback of fast scanning is strong additional noise. As averaging and filtering are always very time-consuming image processing protocols, the tracking approaches must provide a high robustness against extra noise. Thus, methodologies must be developed to accommodate the limited frame rate and poor imaging quality of the SEM.

A number of tracking algorithms have been proposed, such as template matching (Hager & Belhumeur, 1998; Rosolen & King, 1998), feature-based active contours (Kass et al., 1988; Pressigout & Marchand, 2007), and model-based methods (Drummond & Cipolla, 2002; Greminger & Nelson, 2004). But only recently, several tracking algorithms were realized in scanning electron microscopy. Sievers and Fatikow (2006) and Fatikow et al. (2008) proposed a template matching and active contours for semiautomated visual servo tasks in SEM. Nelson also proposed a rigid model-based method for end-effector tracking in an SEM to aid in enabling more precise automated manipulation and measurement (Kratochvil et al., 2009). In these methods, the advantages of template matching tracking methods are simple implementation and high robustness against additive noise. The shortcoming lies in the inaccurate modeling of the target object. Since parts of the background also belong to the model, this approach is sensitive to clutter. Active contours and model-based methods for visual tracking have low robustness against additive noise, as strong edges are needed for feature detection. Another problem is that only an integer pixel displacement can be acquired using the above two tracking methods. The integer pixel resolution is far from that required in some real applications. To further improve tracking accuracy, a subpixel registration algorithm is used to improve the tracking accuracy. As one can note, tracking algorithms in SEM can be mainly divided into two groups: the template matching and

Received September 25, 2011; accepted January 17, 2012
*Corresponding author. E-mail: rchhai@gmail.com

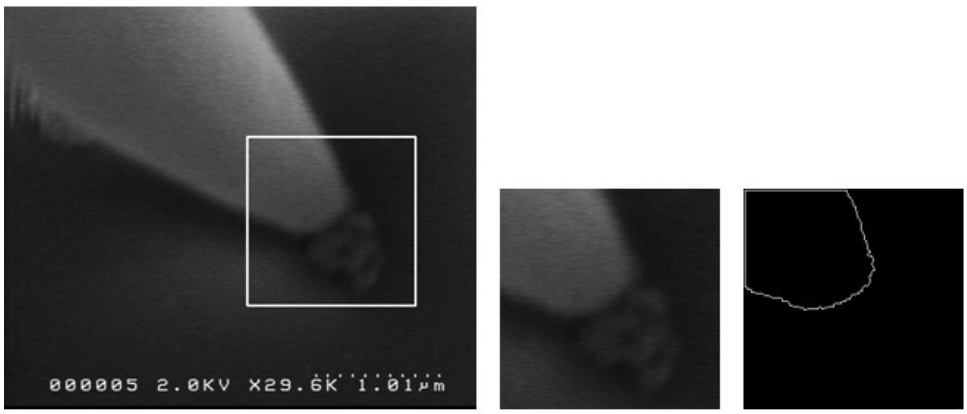

Figure 1. Analysis of contour template (the pixel size is 6.6 nm/pixel): (**a**) referenced image, (**b**) template image, and (**c**) contour template.

model/contour-based ones. Both have complementary advantages and drawbacks. The idea is then to integrate both approaches in the same process. This article addresses the problem of robust tracking of two-dimensional objects by closely integrating template and contour information. A subpixel template matching tracking algorithm based on a contour model in the SEM is proposed to improve the tracking accuracy and the robustness against additive noise and clutter during a nanomanipulation process. A feedforward controller is integrated into the control system to improve the response time.

Subpixel Accuracy Template Matching Algorithm Based on a Contour Model

The template-matching method is widely used to search for an object in an image. This method uses an object image as a template image and moves the template pixel by pixel over the search image. There are a few problems with this conventional template matching method. Its position estimation accuracy is limited to one pixel, so there is a huge matching time cost when searching for a large image. Furthermore, it is not robust against object deformation and environmental change. As the image acquisition time depends on the image size, a region of interest (ROI) optimally adapted to the target object can be acquired much faster than a larger one. But a disadvantage of the conventional method is the inaccurate modeling of the target object because parts of the background also belong to the model as shown in Figure 1b. Thus, this approach is sensitive to clutter. Therefore, the conventional method must be extended to attain high accuracy at a subpixel level, cut down the matching time cost, and increase the robustness against object deformation and environmental change. Although a subpixel matching method will cost more time, the proposed method increases the matching accuracy. Moreover, the added time of subpixel matching can be neglected during a sampling period due to the lower frame rates.

Typically, a subset of $(2M+1)\times(2M+1)$ pixels from the template image is chosen to find its location in the target image to reduce the search time. Once the location of the subset in the target image is found, the displacements of the subset center can be determined. For the best estimation of the displacements, the following sum of squared difference (SSD) correlation criterion is commonly used.

SSD-Based Template Matching Tracking Algorithm

Let $I(\mathbf{x}, t)$ be the gray value at the location $\mathbf{x} = (x, y)^T$ in an image acquired at time t. Let the set $R = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ be the set of image locations that define a target region, and N is the number of pixels in the template image and $N = (2M+1)\times(2M+1)$. $\mathbf{I}(R, t) = (I(\mathbf{x}_1, t), I(\mathbf{x}_2, t), \ldots, I(\mathbf{x}_N, t))$ is a vector of the gray values of the target region. We refer to $\mathbf{I}(R, t_0)$ as the referenced template. It is the template to be tracked and t_0 is the initial time ($t = 0$). The motion of the tracked object induces changes in the position of the template in the image. We assume that these transformations can be perfectly modeled by a parametric motion model $\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}(t))$ where $\mathbf{x}$ denotes an image location and $\boldsymbol{\mu}(t) = (\mu_1(t), \mu_2(t), \ldots, \mu_n(t))$ denotes a set of parameters. We assume that $N > n$ and $\mathbf{f}$ are differentiable both in $\mathbf{x}$ and $\boldsymbol{\mu}$. $\boldsymbol{\mu}$ is defined as the motion parameter vector. The set of N image locations $(\mathbf{f}(\mathbf{x}_1; \boldsymbol{\mu}(t)), \mathbf{f}(\mathbf{x}_1; \boldsymbol{\mu}(t)), \ldots, \mathbf{f}(\mathbf{x}_1; \boldsymbol{\mu}(t)))$ is denoted by $\mathbf{f}(R; \boldsymbol{\mu}(t))$.

With these assumptions, tracking the object at time t means to compute $\boldsymbol{\mu}(t)$ such that

$$\mathbf{I}(\mathbf{f}(R; \boldsymbol{\mu}^*(t)), t) = \mathbf{I}(\mathbf{f}(R; \boldsymbol{\mu}^*(t_0)), t_0), \tag{1}$$

where $\boldsymbol{\mu}^*(t)$ is the ground truth value at time t. Suppose that a reference template is acquired at time t_0, and initially $\boldsymbol{\mu}^*(t_0) = \boldsymbol{\mu}(t_0) = 0$, we write $\boldsymbol{\mu}(t)$ the corresponding estimation of $\boldsymbol{\mu}^*(t)$. The motion parameter vector of the target region $\boldsymbol{\mu}(t)$ at time t_n can be estimated by minimizing the least-squares following function (Hager & Belhumeur, 1998).

$$SSD(\boldsymbol{\mu}) = \sum_{x \in R} [\mathbf{I}(\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}), t_n) - \mathbf{I}(\mathbf{x}, t_0)]^2, \tag{2}$$

where $\mathbf{I}(\mathbf{x}, t_0)$ is the intensity of point $\mathbf{x}$ in the template image, and $\mathbf{I}(\mathbf{f}(\mathbf{x}, \mathbf{u}), t_n)$ is the intensity of point $\mathbf{x}$ in the target region with motion parameters $\mathbf{u}$ at time t_n. In the case of pure translation, the allowed image motion is parameterized by the vector $\boldsymbol{\mu} = (u, v)^T$, giving

$$\mathbf{f}(\mathbf{x};\boldsymbol{\mu}) = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix}, \tag{3}$$

where u and v are the displacements between two subsets in the x and y directions, respectively. The increment $\Delta\boldsymbol{\mu}$ between two time instants can be expressed as

$$\Delta\boldsymbol{\mu} = -(\mathbf{M}_0^T\mathbf{M}_0)^{-1}\mathbf{M}_0^T[\mathbf{I}(\mathbf{f}(R;\boldsymbol{\mu}), t_n) - \mathbf{I}(R, t_0)],$$

where $\mathbf{M}_0$ is an off-line computed constant matrix depending on the template image gradient.

$$\mathbf{M}_0 = [\mathbf{I}_x(\mathbf{x}, t_0)|\mathbf{I}_y(\mathbf{x}, t_0)], \tag{4}$$

$$\mathbf{I}_x(\mathbf{x}, t_0) = \frac{\partial \mathbf{I}}{\partial x} = \begin{bmatrix} \mathbf{I}_x(\mathbf{x}_1, t_0) \\ \mathbf{I}_x(\mathbf{x}_2, t_0) \\ \vdots \\ \mathbf{I}_x(\mathbf{x}_N, t_0) \end{bmatrix},$$

$$\mathbf{I}_y(\mathbf{x}, t_0) = \frac{\partial \mathbf{I}}{\partial y} = \begin{bmatrix} \mathbf{I}_y(\mathbf{x}_1, t_0) \\ \mathbf{I}_y(\mathbf{x}_2, t_0) \\ \vdots \\ \mathbf{I}_y(\mathbf{x}_N, t_0) \end{bmatrix}. \tag{5}$$

Gradient-Based Subpixel Method

Because the minimal unit in a digital image is one pixel, the displacement calculated from equation (2) is an integer multiple of one pixel. To further improve the position estimation accuracy, some kind of subpixel registration algorithm should be used. In the following, a gradient-based subpixel method is utilized to obtain subpixel position estimation accuracy. The gradient-based method was first developed by Davis and Freeman (1998) as an optical flow method, which is used in digital image correlation. Based on the basic assumption of the gradient-based subpixel accuracy, the subset rigid body translation exists when the subset is small enough.

$$\mathbf{I}(\mathbf{f}(\mathbf{x};\boldsymbol{\mu}), t_n) = \mathbf{I}(\mathbf{x}, t_0), \tag{6}$$

where

$$\mathbf{f}(\mathbf{x};\boldsymbol{\mu}) = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u + \Delta x \\ v + \Delta y \end{bmatrix}. \tag{7}$$

In equation (7), Δx and Δy are the subpixel displacements. After neglecting the high-order terms, the first-order Taylor expansions of $\mathbf{I}(\mathbf{f}(\mathbf{x};\boldsymbol{\mu}), t_n)$ at time t_n yields

$$\begin{aligned} &I((x + u + \Delta x, y + v + \Delta y), t_n) \\ &\quad = I((x + u, y + v), t_n) \\ &\qquad + \Delta x \cdot I_x((x + u, y + v), t_n) \\ &\qquad + \Delta y \cdot I_y((x + u, y + v), t_n), \end{aligned} \tag{8}$$

where I_x and I_y are the first-order derivatives of the intensities in the rectified image with motion parameters $\boldsymbol{\mu}$ at time t_n. They can be calculated by convolution of gray with the mask of $[1/12, -8/12, 0, 8/12, -1/12]$ in the x and y directions, respectively. Its truncation error is $o(h4)$, much smaller than the Sobel and Prewitt gradient operators, which are commonly used in digital image processing to extract the image border. Equation (8) can be solved using the least-squares method, and the solution can be expressed in the following closed form:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sum_{x\in R} I_x^2 & \sum_{x\in R} I_x \cdot I_y \\ \sum_{x\in R} I_x \cdot I_y & \sum_{x\in R} I_y^2 \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{x\in R} (\mathbf{I}(\mathbf{x}, t_0) - \mathbf{I}(\mathbf{f}(\mathbf{x};\boldsymbol{\mu}), t_n)) \cdot I_x \\ \sum_{x\in R} (\mathbf{I}(\mathbf{x}, t_0) - \mathbf{I}(\mathbf{f}(\mathbf{x};\boldsymbol{\mu}), t_n)) \cdot I_y \end{bmatrix}. \tag{9}$$

The Template Matching Based Contour Model

The main drawback of the SEM is the low image quality due to strong additional noise when the frame rate is high in order to realize the visual tracking. Besides additive noise, two further sources of noise are important. The first one is gray-level fluctuation, which is a significant issue in scanning electron microscopy. Gray-level fluctuations occur due to electrostatic charge and due to variations of the alignment of target, electron beam, and secondary electron detector. Especially for pattern matching approaches without invariance against amplitude variations, the fluctuations will cause some trouble. The second source of noise is clutter caused by objects in the image background or foreground. The background objects often mask edges while objects in the image foreground can occlude the target object. Therefore, the tracking method in the SEM should satisfy the demand that is not only robust against the environment noise such as the additive noise, the gray-level fluctuations, and clutter, but also fast in image processing.

The advantages of template-matching-based tracking are simple implementation and high robustness against additive noise. But one of the main disadvantages is the inaccurate modeling of the target object, and the background objects often mask edges. Some small nontarget objects on the substrate surface are also included in the template image as shown in Figure 1b. So a number of operations have to be done on pixels in the template that do not contain really relevant information. Thus, the template-matching-based tracking method is sensitive to clutter. Moreover gray-level fluctuation may change the appearance of the object in such a way that it does not resemble the template enough to yield a reasonable match value. An improved template matching algorithm is proposed to solve the drawbacks, which uses the contour detected from template image as the template as shown in Figure 1c. Although contour templates will lose some other useful features of the manipulator tip, a limited set of template pixels is really characteristic for the target object, and the contour is considered to be the most characteristic part of a target object. With the manipulator tip contacting a sample, it is easy to identify the contour of the probe if the gray contrast between the probe and the sample is distinct. But it is difficult to obtain the full contour of the probe if the gray level between the probe and

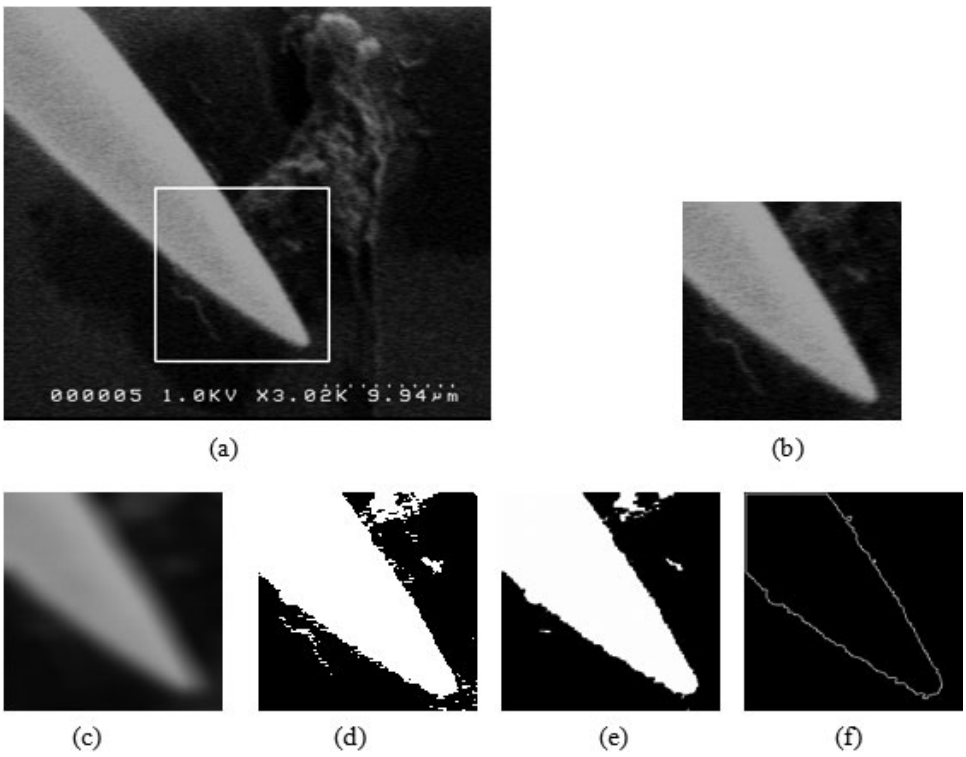

Figure 2. Image sequence of contour template identification (the pixel size is 66 nm/pixel): (**a**) original image, (**b**) ROI of contour template, (**c**) smooth image, (**d**) Otsu adaptive threshold, (**e**) eroded and dilated image, and (**f**) identified contour template.

the sample is similar. Force sensing in this case will be a more effective way to resolve such problems.

Realization of the Proposed Tracking Algorithm

The proposed subpixel template matching based on contour consists of two steps: first, identifying the contour template, which is shown in the first image in Figure 2a; then searching the object in a target-image-based subpixel template matching algorithm, which is done repeatedly for different target images. These steps will be described below.

Contour Template Identification

In the following experiments, a tungsten probe (Picoprobe T-4-5, GGB Industries, Inc., Naples, FL, USA) 200 nm in diameter is tracked based on the visual image in the SEM. As the image acquisition time depends on the image size, an ROI optimally adapted to the target object can be acquired much faster than a larger one. So an ROI surrounding the probe tip for tracking is identified in order to track the probe and obtain the real position of the probe tip. An identification algorithm is developed to obtain the contour template in the first image frame.

1. *Pointing an ROI including the probe tip:* Two simple methods are used to obtain the ROI. The first method is that the operator clicks the mouse near the probe tip, and a 200×200 ROI is obtained automatically by the PC at the magnification 3,000× as shown in Figure 2b. The other method is that the probe is moved along x or y axis at a constant step displacement, the unchanged background of each frame of the image is subtracted, and the moving probe can be identified. Typically, the ROI is found in less than 10 image frames. Commonly, the ROI can be changed in its dimension according to the size of the tracked object at a different magnification of the SEM, which will save image processing time. Larger ROI dimensions cost more processing time.
2. *Smooth:* Each image is first convolved with a low-pass Gaussian filter for noise suppression through utilizing a Gaussian function. The Gaussian blur is a type of image-blurring filter (which is also used for the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. The smoothed image is shown in Figure 2c.
3. *Adaptive threshold:* The Otsu adaptive threshold method (Otsu, 1979) is used to automatically perform the transformation from a gray-level image to a binary image in Figure 2d.
4. *Morphological operation:* A combination of the erosion and dilation operators is used to remove noise, isolate individual elements, and joint disparate elements after thresholding as shown in Figure 2e.
5. *Obtain the contour template:* Retrieve all of the contours from the binary images created by smoothing, thresholding, and morphological operations and then organize them into a two-level hierarchy, where the top-level boundaries are external boundaries of the components and the second level boundaries are boundaries of the external boundaries. The contours are sequences of points in the ROI. Commonly, the largest contour (external boundary) is considered as the contour template in Figure 2f.

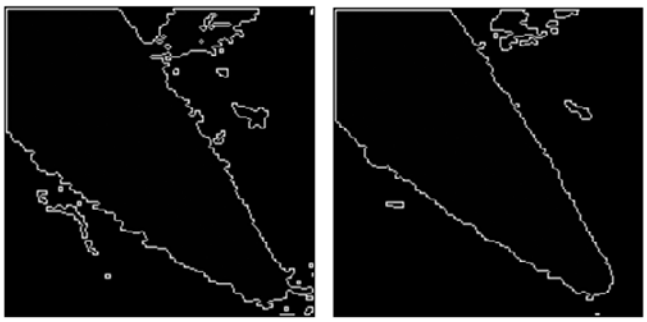

Figure 3. Contour edge detection: (**a**) detects weaker edges due to the lower threshold and (**b**) after adaptive threshold Canny edge detection.

Subpixel Template Matching Algorithm Based on the Contour Template

When the contour template is available, containing the pixels that are on the edge of (a part of) the object, we need to transform the gray-level target image that is retrieved from the SEM into an image with only contour, using a Canny edge detector with adaptive threshold. The Canny edge detector tracks the local maximum of an image gradient and eliminates weak edges by hysteresis thresholding using two constant threshold values. The high threshold T_h is used to detect strong edges, while the low threshold T_l detects weaker edges that are connected to strong edges. So a Canny edge detector using an inappropriate fixed threshold may miss some obvious edges or involve weak edges. Due to the effect of the gray fluctuation and the clutter caused by the image background, foreground, etc., the contour of the tracked object is confused as shown in Figure 3a and the original image is shown in Figure 2. An adaptive threshold algorithm for Canny edge detection (Zhi et al., 2005) is employed for tracking the contour of the target object. In the adaptive threshold method, the histogram $H(i)$ from the gradient magnitude of the image is first calculated, and the maximum value $H_{\max}$ is determined. The deviation of $H(i)$ from $H_{\max}$ is given by

$$\sigma_1 = \sqrt{\frac{\sum_{i=0}^{N} [H(i) - H_{\max}]^2}{N}}. \tag{10}$$

Accordingly, T_h is set to be $H_{\max} + \sigma_1$. The deviation σ_1' of $H(i)$ from $H_{\max}$ is recalculated to exclude those pixels with gradient magnitudes above T_h. Finally, the low threshold value is determined as $T_l = H_{\max} + \sigma_1'$. Figure 3b shows the result after Canny edge detection with adaptively determined threshold values. We can see that a clear contour is obtained.

Within each image frame the template matching algorithm is calculated to locate the best match within a search window of 300 × 300 pixels, which is around the contour of the target object in the previous image. A larger search window will result in a longer search time, while a smaller search window may result in a failure of tracking the target object. Based on the subpixel template matching algorithm, the contour template is placed on all possible positions in the search window of the target image, computing a match value for every position. The match value now is based on the edge pixels in the contour template only, and the target image pixels that correspond to an edge pixel in the contour template are added. In most cases, there will be a unique subpixel position where the object has a maximum match value.

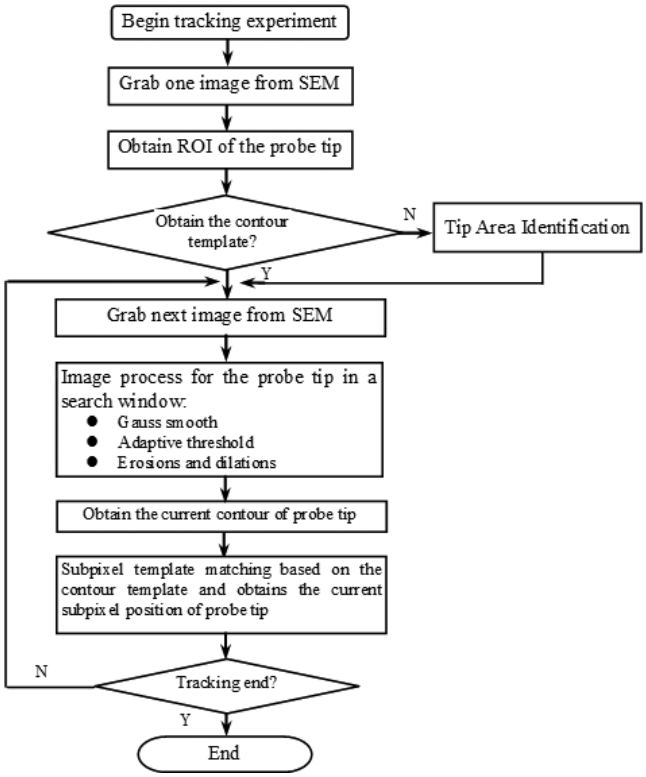

Figure 4. The flow chart of proposed tracking algorithm.

The whole process of the proposed tracking algorithm is described in Figure 4. First, the ROI of the probe tip is identified from a grabbed image, then the contour template is obtained. Next, the target object is tracked based on the subpixel template matching, which includes the following four steps: (1) grab one new image; (2) image process for the probe tip in a search window that includes the Gaussian smooth filter, adaptive threshold, and morphological operations; (3) obtain the current contour of the probe tip; and (4) apply the subpixel template matching to obtain the current position of the probe tip.

Tracking Experiments

Nanomanipulation System Setup

The nanomanipulation system based on a SEM is shown in Figure 5. The following experiments were performed with a Hitachi S-4000 scanning electron microscope (Hitachi, Ltd., Tokyo, Japan). In the following experiments, an image size

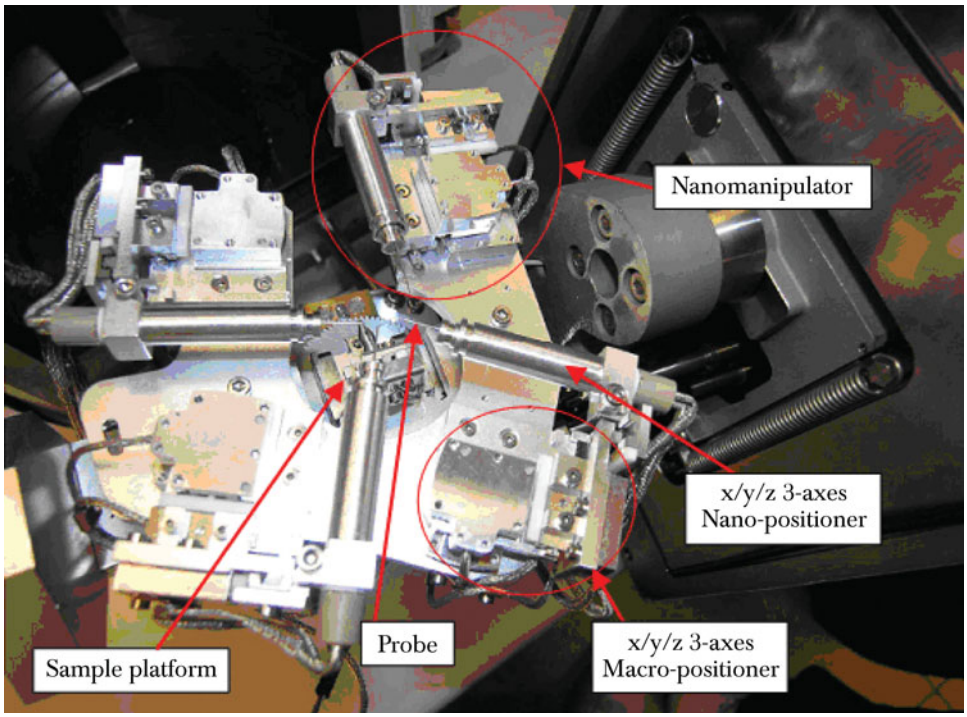

Figure 5. The nanomanipulation system in Hitachi SEM 4000.

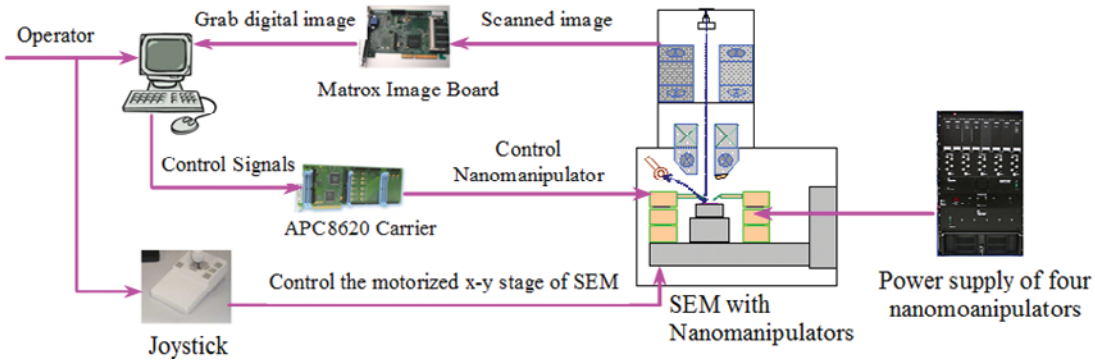

Figure 6. The control system architecture.

of 640 × 480 was selected to allow for a viewing area and the possibility for sampling at a rate of 13 Hz. The scanning image of the SEM is converted into a digital image by a Matrox image board (Matrox Electronic Systems Ltd., Dorval, Quebec, Canada). The Zyvex S100 nanomanipulation system is integrated into the vacuum chamber of an SEM. The setup mounted on a base plate contains four 3-degrees-of-freedom nanomanipulators. In each nanomanipulator, there are two scales of stepper motion: MACRO and NANO. With the MACRO motion, the macropositioner will move faster but will have a coarser, rougher motion. The nanopositioner can achieve ultraprecise motion in the NANO scale. The resolution of macropositioner with $x/y/z$ is 100 nm, and the maximum range of motion is 12 mm in the X and Z axes, and 10 mm in the Y axis. Resolution of the nanopositioner with $x/y/z$ is 5 nm, and the travel range is 50 μm in the X and Z axes, and 10 μm in the Y axis. An Industry Pack module on an APC8620 Carrier (produced by Acromag Inc., Wixom, MI, USA) is used to convert the control signals to drive the nanomanipulator, which has two 16-bit and one 12-bit D/A converters. Figure 6 shows the control system architecture currently used for contact detection experiments in the SEM. The x-y motorized stage of the SEM is controlled by a joystick and can move and adjust the nanomanipulators and sample platform within the vacuum chamber. The sample can be observed by using the image processing software running on the SEM PC and its joystick positioning controller. After adjusting the SEM image, the control signals produced by the computer are used to drive the three-axis macropositioner for the coarse positioning of the probe tip close to the substrate surface. Once the distance between the substrate and probe tip is in the stroke of the nanopositioner, the control is switched to the nanopostioner to realize the fine approach. All the control signals are converted by the APC8620 Carrier to move the nanomanipulator. The scanned image converted by the Matrox image board is obtained. The control software is written in VC++ 6.0 program.

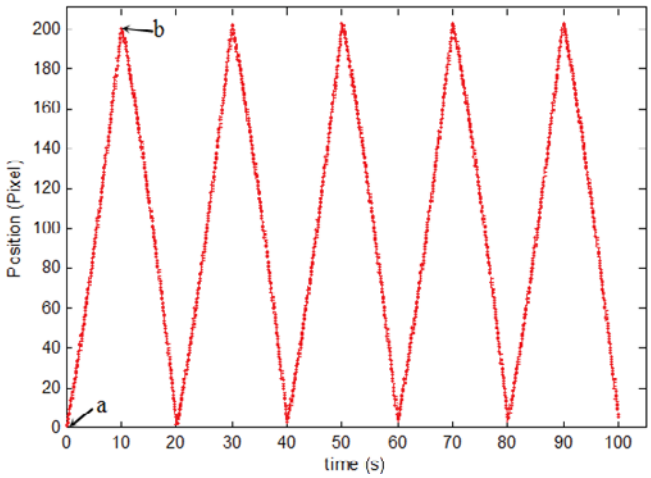

Figure 7. Tracking position curve of the probe tip based on the visual tracking algorithm.

Tracking Performance

To evaluate the repeatability of the visual tracking algorithm, a point is chosen, the probe tip detection is performed 100 times for that point, and the standard deviation and average error of the measurement are 0.2268 and 0.0412 pixels, respectively, at magnification 3,000×, and the pixel size is 66 nm/pixel. To evaluate the performance of the tracking algorithm, one nanomanipulator moves from current position "A" to desired position "B" with 200 pixels distance and repeats five times. The tracking positions of the probe tip based on the tracking algorithm are shown in Figure 7. The standard deviation accuracy in point B is 0.6255 (standard deviation 1σ) pixels. We can see that the tracking algorithm accomplishes the tracking motion of the probe tip with subpixel accuracy.

Visual Servo Control Incorporating a Feed-Forward Controller

One major challenge of performing visual servoing tasks inside an SEM is balancing the needs of image quality with the tracking speed. The sequential scanning used to create an SEM image results in lower frame rates than those available with optical cameras. Low frame rates of the SEM limit the response speed of the visual servo control system. This issue was previously addressed by selectively scanning a smaller ROI (Fatikow et al., 2007). Nevertheless, that method requires the installation of new hardware for accessing the scan controller of the SEM. Without altering a standard SEM configuration, we add a feed-forward controller in the visual control system, as shown in Figure 8a. The feed-forward controller contains a mathematical model accounting for the hysteresis of the piezoelectric nanomanipulator (Ru & Sun, 2005). The linear fit might be suitable for a coarse compensation of the piezo hysteresis, such that the probe can be quickly moved to the vicinity of a target position without depending on visual feedback. The applied hysteresis model based on linear fit is simple and the model parameters are obtained easily. But for obtaining a more

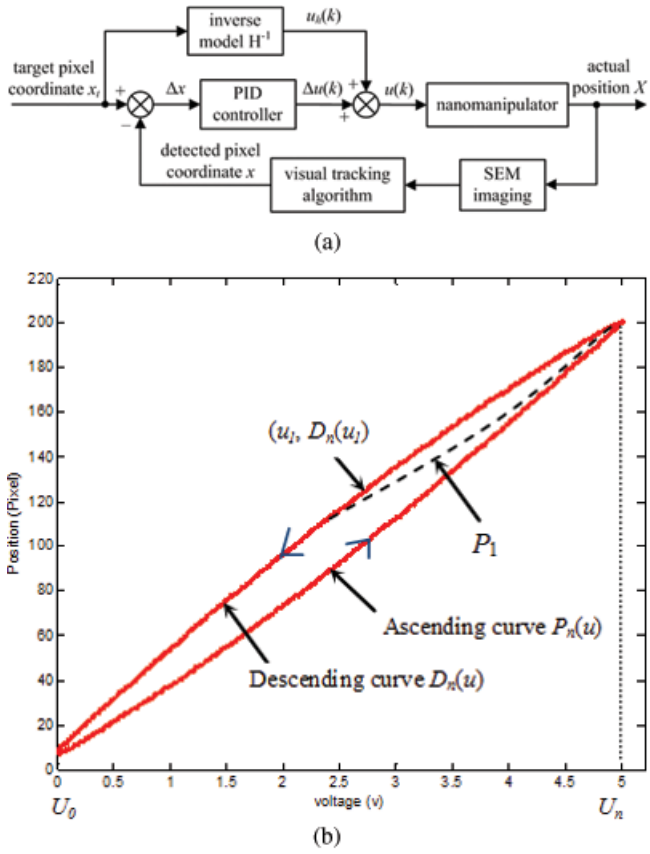

Figure 8. Feed-forward control method: (**a**) SEM image-based visual servo control system incorporating a feed-forward controller and (**b**) characterization of a piezoelectric actuator for feed-forward control within the visual servo control system.

accurate hysteresis model, some other models, e.g., PI, Preisach, are recommended. The proportional-integral-derivative (PID) controller in the image-based visual servo control system then brings the probe tip precisely to the target position.

To obtain the parameters in the mathematical model, voltages are incrementally applied to each axis of the actuator and then incrementally released with the corresponding displacements in the image frame recorded by the visual tracking algorithm. As an example, the characterization result of the x axis of the piezoelectric tube of one of our four nanomanipulators at the magnification of 3,000× is shown in Figure 8b. The applied voltage is denoted by u, with its lowest and highest values denoted by U_0 and U_n. The ascending and descending curves are denoted by functions $P_n(u)$ and $D_n(u)$, respectively, and are fitted with a fourth-order polynomial. If the piezoelectric actuator is at any point on the ascending (descending) curve and the applied voltage is monotonically increasing (decreasing), the output displacement is $P_n(u)(D_n(u))$. However, if the applied voltage starts to decrease (increase) when the actuator is on the ascending (descending) curve, the actuator will stray away from that curve due to hysteresis. In that case, a mathematical model is used to calculate the output displacement. When the voltage increases from $(u_1, D_n(u_1))$ on the

descending curve (Fig. 8b), the output displacement (dashed curve in Fig. 8b) is

$$P_1(u) = k \cdot P_n(m \cdot (u - U_n) + U_n) + D_n(U_1) - k \cdot P_n(U_0), \quad (11)$$

where $k = (D_n(U_n) - D_n(u_1))/(D_n(U_n) - D_n(U_0))$ and $m = (U_n - u_1)/(U_n - U_0)$. When the actuator is on the descending curve and a certain displacement is desired, the corresponding applied voltage is calculated via the Newton-Raphson method to invert $D_n(u)$ or the mathematical model, as the output of the feed-forward controller. A similar mathematical model is used for the scenario where the applied voltage starts to decrease while the actuator is on the ascending curve to calculate the output of the feed-forward controller.

For performance evaluation, a step signal (the desired displacement in pixels) was input to the visual servo control system. The responses of the system to a 100-pixel-step input with and without using the feed-forward controller are shown in Figure 9. Figure 9a shows the step response of vision servo PID feedback controller without using the feed-forward controller, and the response time during the OA process is about 5.23 s for 100 pixel step at magnification 3,000×. Figure 9b shows the step response of the vision servo PID feedback controller with feed-forward compensation. During the OB process, the nanomanipulator is driven based on the feed-forward controller and the consuming time is only 0.61 s. After that, the vision servo PID feedback controller is used to reduce the error and the desired position is reached at point C. The whole response time is less than 1.3 s. The response time of the vision servo PID feedback controller with feed-forward compensation is shorter and decreased to about 70%. The rate of convergence is improved and there is no overshot and oscillation. The vision servo PID feedback controller with feed-forward compensation widens the bandwidth of the system, linearizes the system behavior, and improves the positioning accuracy. As shown in Figure 8c, the maximum positioning error is less than ±1 pixel; the mean (m) and standard deviation (1σ) of visual servo algorithm are 100.059 pixels and 0.4584 pixels.

Conclusion

This article presents a robust and effective tracking framework for visual servoing applications inside an SEM to aid in enabling more precise tracking manipulations and measurements. A subpixel template matching tracking algorithm based on a contour model in the SEM is developed to improve the tracking accuracy and the robustness against additive noise and clutter during a nanomanipulation process. A visual servo control system with a feed-forward controller is built for closed-loop control of multiple nanomanipulators, which demonstrates significant improvement in response time in comparison to the feed-forward-controller-free scenario. The experimental results show that a subpixel track-

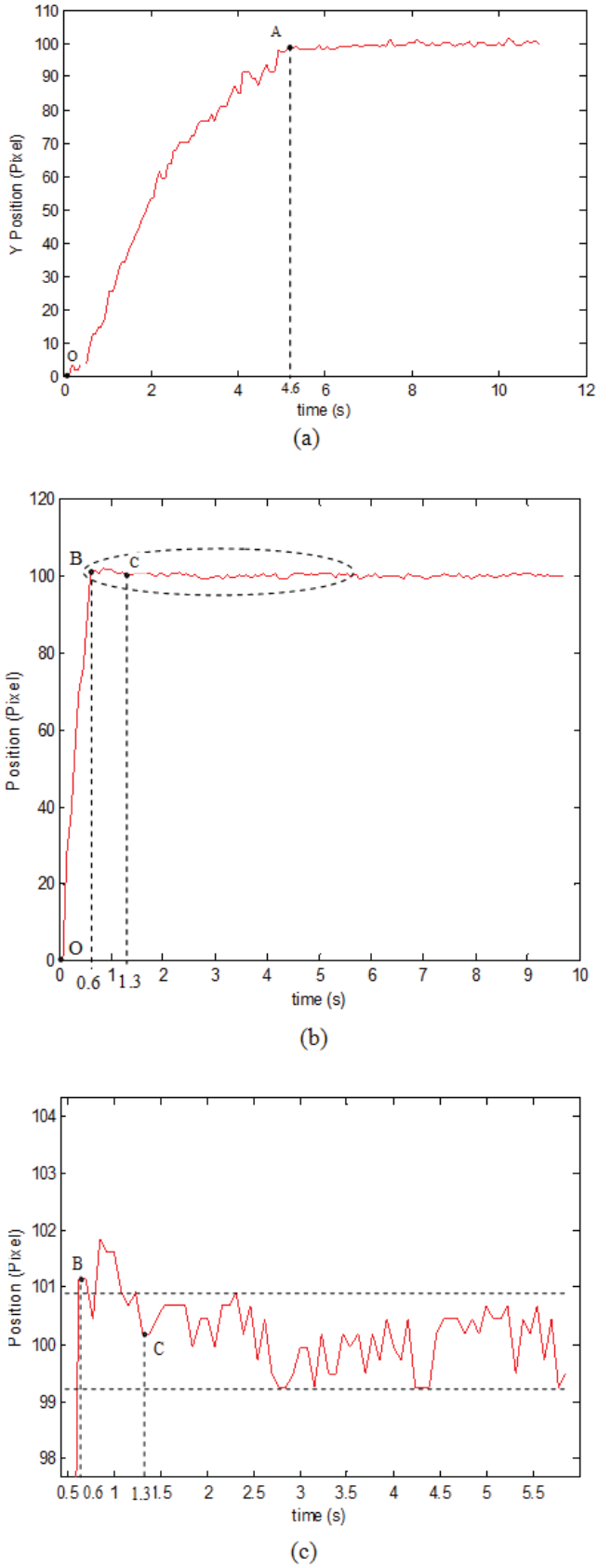

Figure 9. Step responses of visual servo control system for one axis of piezoelectric tube (the pixel size is 66 nm/pixel): (**a**) without feed-forward compensation, (**b**) with feed-forward compensation, and (**c**) local enlargement.

ing accuracy is obtained, and the robustness against clutter is achieved in a challenging tracking environment.

Acknowledgment

The authors gratefully acknowledge the contribution of the National Natural Science Foundation of China (No. 61174087) and Foundation for University Key Teacher of Heilongjiang Province of China.

References

Davis, C.Q. & Freeman, D.M. (1998). Statistics of subpixel registration algorithms based on spatiotemporal gradients or block matching. *Opt Eng* **37**, 1290–1298.

Dietzel, D., Mönninghoff, T., Jansen, L., Fuchs, H., Ritter, C., Schwarz, U.D. & Schirmeisen, A. (2007). Interfacial friction obtained by lateral manipulation of nanoparticles using atomic force microscopy techniques. *J Appl Phys* **102**, 084306.

Drummond, T. & Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Trans Pattern Anal* **24**, 932–946.

Eigler, D.M. & Schweizer, E.K. (1990). Positioning single atoms with a scanning tunnelling microscope. *Nature* **344**, 524–526.

Falvo, M.R., Taylor, R.M., Helser, A., Chi, V., Brooks, F.P., Washburn, S. & Superne, R. (1999). Nanometre-scale rolling and sliding of carbon nanotubes. *Nature* **397**, 236–238.

Fatikow, S., Eichhorn, V., Stolle, C. & Sievers, T. (2008). Development and control of a versatile nanohandling robot cell. *Mechatronics* **18**, 370–380.

Fatikow, S., Wich, T., Hulsen, H., Sievers, T. & Jahnisch, M. (2007). Microrobot system for automatic nanohandling inside a scanning electron microscope. *IEEE/ASME Trans Mechatron* **12**, 244–252.

Fukuda, T., Nakajima, M., Liu, P. & ElShimy, H. (2009). Nanofabrication, nanoinstrumentation and nanoassembly by nanorobotic manipulation. *Int J Robotics Res* **28**, 537–547.

Greminger, M.A. & Nelson, B.J. (2004). Vision-based force measurement. *IEEE Trans Pattern Anal* **26**, 290–298.

Hager, G.D. & Belhumeur, P.N. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans Pattern Anal* **20**(10), 1025–1039.

Kass, M., Witkin, A. & Terzopoulos, D. (1988). Snakes: Active contour models. *Int J Comput Vision* **1**, 321–331.

Kratochvil, B.E., Dong, L.X. & Nelson, B.J. (2009). Real-time rigid-body visual tracking in a scanning electron microscope. *Int J Robot Res* **28**, 498–511.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* **9**, 962–966.

Pressigout, M. & Marchand, E. (2007). Real-time hybrid tracking using edge and texture information. *Int J Robotics Res* **26**, 689–713.

Prior, M., Makarovski, A. & Finkelstein, G. (2007). Low-temperature conductive tip atomic force microscope for carbon nanotube probing and manipulation. *Appl Phys Lett* **91**, 053112.

Rosolen, G. & King, W. (1998). An automated image alignment system for the scanning electron microscope. *Scanning* **20**, 495–500.

Ru, C. & Sun, L. (2005). Improving positioning accuracy of piezoelectric actuators by feed-forward hysteresis compensation based on a new mathematical model. *Rev Sci Instrum* **76**, 095111.

Rubio, F.J., Heckl, W.M. & Stark, R.W. (2005). Nanomanipulation by atomic force microscopy. *Adv Eng Mater* **7**, 193–196.

Sierra, D.P., Weir, N.A. & Jones, J.F. (2005). A review of research in the field of nanorobotics. Technical Report SAND2005, 6808, Sandia National Laboratories.

Sievers, T. & Fatikow, S. (2006). Real-time object tracking for the robot-based nanohandling in a scanning electron microscope. *J Micromechatronics* **18**, 267–284.

Suga, H., Naitoh, Y., Tanaka, M., Horikawa, M., Kobori, H. & Shimizu, T. (2009). Nanomanipulation of single nanoparticle using a carbon nanotube probe in a scanning electron microscope. *Appl Phys Exp* **2**, 055004.

Zhi, W., Li, Q., Zhong, S. & He, S. (2005). Fast adaptive threshold for the canny edge detector. *Proc. SPIE* **6044**, 60441.