2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Mutation Analysis Models for Visual Servoing in Nanomanipulations

Jianguo Zhao, Bo Song, Ning Xi, and King Wai Chiu Lai

Abstract— **This paper has two purposes: investigating a featureless visual servoing approach based on mutation analysis and proposing a visual servo control method for nanomanipulations. For the first purpose, the featureless visual servoing method is needed because traditional visual servoing relies heavily on robust feature extraction and tracking, which are very difficult in natural environment. The mutation analysis based approach in this paper considers the image as a set, and designs a controller to make the distance between the initial and goal image sets converge to zero, thereby steering the initial image to the goal image. For the second purpose, atomic force microscopic (AFM) based nanomanipulations with subnanometer accuracy are very difficult because the position sensor cannot provide valuable feedback due to large noises at this precision level. We propose to use the images obtained by AFM and perform a visual servo control. This method, independent of external sensors, can directly perform control on the AFM end tip's position. The featureless controller is successfully validated on AFM images and the results suggest a potential precision enhancement for nanomanipulations.**

I. INTRODUCTION

Visual servoing uses vision information to control the motion of a mobile robot or a manipulator for various tasks such as navigation or manipulation. Consider the eye-in-hand configuration when the camera is rigidly attached to a robot or a manipulator. The goal for visual servoing is to design a control law, under which the mobile robot or manipulator can move, so that the current image will eventually converge to a specified goal image [1]. This problem is usually solved in vector space by minimizing the error $e(t)$ defined as $e(t) = s(t) - s^*$, where $s(t)$ and s^* are the current and desired position vectors of features from the image, respectively [1]. This approach, however, relies heavily on robust feature extraction and tracking. In fact, good features may be difficult to find and they may also become occluded during the movement in natural environment [2]. Furthermore, reliable feature tracking for visual servo in complex real environments is quite challenging [3]. To address these problems, artificial fiducial markers are employed for the experiments in most of visual servoing literature.

Instead of using fiducial markers, there is a trend to come up with featureless visual servoing techniques in recent years. In these techniques, the servo control is directly performed based on pixel intensity information in the image, and no explicit feature extraction and tracking is required. Several different featureless visual servoing methods are proposed in literature. In general, they can be classified into four categories based on how the error $e(t)$ is defined.

The first method [4], considering the whole image as a feature, performs principle component analysis on the original image to obtain an eigen space with reduced dimension. Then the control is performed to make the error in this eigen space converge to zero. This method, although not depending on explicit feature extraction, requires processing on the entire image; moreover, formal proof of convergence is not provided. The second method [5], similar to the first one, processes the original image with a spatial sampling function to derive a kernel measurement, and a controller is designed to make the measurement error converge to zero. But the controller is only obtained for a subset of rigid body motions instead of a general six degree-of-freedom motion. The third method tries to make the sum-of-squared-difference of intensities between two images converge to zero, and the matrix relating the camera motion to the motion of individual pixels is obtained through the optical flow equation. Two different implementations exist for this method. In [6], the optimization based control is performed without formal convergence proof, while in [7], [8], similar gradient control is designed to stabilize a helicopter but with a formal stability proof. It's also worth mentioning that the latter considers a dynamic model with force/torque input instead of a kinematic model with only velocity input. The last method employs the entropy from information theory to obtain the mutual information between two images and uses an optimization approach to maximize it [9]. This method is robust with respect to different lighting conditions.

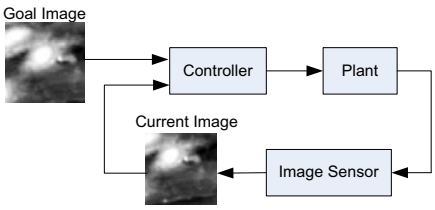


Fig. 1. The scheme for mutation analysis based visual servoing

The mutation analysis based visual servoing approach is another featureless method proposed earlier than all above methods [10]. The basic idea for this approach is shown in Fig. 1. Both the current and goal images are considered as sets, and the error between two images is defined as the distance between two sets. A controller is then designed to make the image set distance converge to zero. Nevertheless, since the space of sets does not have the linear structure of vector space, the traditional dynamics model in vector space

This work is partially supported by NSF Grant NO. IIS-0713346 and ONR Grant NOs. N00014-07-1-0935 and N00014-04-1-0799.

Jianguo Zhao, Bo Song, Ning Xi, and King Wai Chiu Lai are with Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48824, USA. {zhaojia1, songbo}@msu.edu, {xin, kinglai}@egr.msu.edu

cannot be used; instead, the non-vector mutation analysis for set evolution should be employed [11], [12]. Therefore, this method provides a fundamentally different way to solve the featureless servoing problem.

The purpose of this paper is twofold. On one hand, we want to bring this mutation analysis based visual servoing method into the community's attention. Essentials of mutation analysis will be explained, and the controller for binary images in [10] will be extended to gray scale images. On the other hand, we propose a potential application for visual servoing: atomic force microscope (AFM) based nanomanipulation, and validate the mutation analysis based method with simulations on AFM images. Note that a literature review for AFM based nanomanipulations will be given in section IV.

The rest of paper is organized as follows. First of all, essential concepts in mutation analysis will be presented in section II. After that, the mutation equation model for visual servoing and the controller design are given in section III. Based on the controller, the potential application for nanomanipulation with visual servoing is presented in section IV, and the simulation results with AFM images are demonstrated in section V. Finally, the conclusions are given.

II. Mutation Analysis for Set Evolutions

Mutation analysis deals with the dynamics of sets instead of vectors. In this case, the linear structure of vector space does not exist because the addition and scalar multiplication of sets cannot be properly defined [12]. This nonvectorial characteristic requires a set of new tools which will be elaborated in this section.

A. Distance Between Sets

Although lack of linear structure, the space of sets can be a metric space if a proper metric is defined. The Hausdorff distance between two sets can be such a metric; indeed, it has been widely used to compare binary images [13]. We define the Hausdorff distance step by step. First of all, the distance between a point $x \in \mathbb{R}^n$ and a set $K \subset \mathbb{R}^n$ is defined as: $d_K(x) = \inf_{y \in K} ||y-x||$, where $||\cdot||$ is the Euclidean distance between two points. The projection of x to K is the set of points $y \in K$ defined by $\Pi_K(x) = \{y \in K : ||y-x|| = d_K(x)\}$. The distance from set X to set Y is defined as $d(X,Y) = \sup_{x \in X} d_Y(x)$, and the distance from Y to X is $d(Y,X) = \sup_{y \in Y} d_X(y)$. Generally, these two distances do not equal to each other and thus will not form a metric. But the Hausdorff distance defined as

$$dh(X,Y) = \max\{d(X,Y), d(Y,X))\} \tag{1}$$

is a metric. The gray scale images can be considered as a set of points in three dimensions because each pixel has two pixel index values and one intensity value. Therefore, the Hausdorff distance between two gray scale images X and Y is:

$$dh(X,Y) = \max\{\max_{x \in X} \min_{y \in Y} ||x-y||, \max_{y \in Y} \min_{x \in X} ||y-x||)\} \tag{2}$$

where $x, y \in \mathbb{N}^3$ are vectors formed by three natural numbers. Note that since the index values and intensity have different units, a proper rescale for the index values should be performed so that they have the same range as the intensities. The computational complexity for Hausdorff distance is large if the scanning search method is used, but it can be greatly reduced if other branch-and-bound approaches are used [14].

B. Tubes and Transitions

To describe the set dynamics, the evolution of a set, called tubes, should be defined at first. A tube $K(t) \subset \mathbb{R}^n$ is defined as [10]:

$$K(\cdot): \quad \mathbb{R}^+ \mapsto 2^{\mathbb{R}^n} \tag{3}$$

where $2^{\mathbb{R}^n}$ is the power set of $\mathbb{R}^n$ which is the collection of all subsets of $\mathbb{R}^n$. Let $\varphi : E \mapsto \mathbb{R}^n$ with $E \subset \mathbb{R}^n$ be a bounded Lipschitz function. Denote the set of all such functions as $\mathrm{BL}(E, \mathbb{R}^n)$. For ordinary differential equation (ODE) $\dot{x} = \varphi(x)$ with initial condition $x(0) = x_0$. The transition for φ is defined as:

$$T_\varphi(t, x_0) = \{x(t) : \dot{x} = \varphi(x), x(0) = x_0\} \tag{4}$$

where $x(t)$ is the solution to above ODE. In other words, the transition is nothing but the solution to a given ODE at time t. Note that a rigorous definition for transitions requires it to satisfy four conditions [12], but for simplicity, we use above definition which can be shown to satisfy those four conditions (Ch. 1, Example 3, [12]). The above definition can be extended when the initial condition is a set instead of a point:

$$T_\varphi(t, K_0) = \{x(t) : \dot{x} = \varphi(x), x(0) \in K_0\} \tag{5}$$

where K_0 is a set containing all possible initial conditions which implies $T_\varphi(t, K_0)$ is also a set. In fact, $T_\varphi(t, K_0)$ can be considered as a reachable set of points at time t generated by φ with the initial points in K_0. It can also be considered as a tube evolving from K_0 under the rule given by φ. Note that this definition can also be shown to satisfy the four conditions for transitions (Ch. 1, Example 4, [12]).

C. Mutation Equations

To describe the set dynamics, time derivative of a tube should be defined, but this is not trivial since the space of sets has only the structure of metric space. Fortunately, we can extend the derivative of a function in vector space to metric space based on transitions. For a function $f : \mathbb{R}^+ \mapsto \mathbb{R}^n$, the derivative is defined as $v = \lim_{\Delta t \to 0} [f(t+\Delta t) - f(t)]/\Delta t$. This can be rewritten in the first order approximation form as [12]:

$$\lim_{\Delta t \to 0} \frac{1}{\Delta t} ||f(t+\Delta t) - (f(t) + v\Delta t)|| = 0 \tag{6}$$

where $f(t) + v\Delta t$ can be considered as a new point in $\mathbb{R}^n$ obtained by starting from $f(t)$ and moving along the direction v after Δt time. Similarly, in the context of tube, transition $T_\varphi(\Delta t, K(t))$ can be considered as a new set in $\mathbb{R}^n$ obtained by starting from $K(t)$ and moving along the direction of φ after Δt time. Therefore, similar to Eq. (6), the derivative of a tube $K(t)$ should be the function φ satisfying:

$$\lim_{\Delta t \to 0^+} \frac{1}{\Delta t} dh(K(t+\Delta t), T_\varphi(\Delta t, K(t))) = 0 \tag{7}$$

where $K(t+\Delta t)$ is the set at time $t+\Delta t$ according to the tube $K(\cdot): \mathbb{R}^+ \mapsto 2^{\mathbb{R}^n}$. Note that for a given tube, there may be none or multiple φ satisfying Eq. (7); therefore, the mutation of a tube $K(t)$ is defined as the set, possibly empty, of all φ:

$$\mathring{K}(t) = \{\varphi(x) \in \mathrm{BL}(E, \mathbb{R}^n) : \text{Eq. (7) is satisfied}\} \quad (8)$$

Based on the mutation of a tube, the mutation equation describing set dynamics is defined as:

$$\varphi(x) \in \mathring{K}(t) \quad (9)$$

The mutation equation defined above has only x as its parameter. To add the control input to the equation, consider a map $\varphi : E \times U \mapsto \mathrm{BL}(E, \mathbb{R}^n)$ where U is the set of all possible controls u. Then the controlled mutation equation can be defined as:

$$\varphi(x(t), u(t)) \in \mathring{K}(t) \quad \text{with} \quad u(t) = \gamma(K(t)) \quad (10)$$

where $\gamma: 2^{\mathbb{R}^n} \mapsto U$ is the feedback map from current set $K(t)$ to the control input.

III. Mutation Analysis Based Visual Servoing

Based on mutation analysis, the visual servoing problem can be formulated as a stabilization problem: Given a goal image set $\hat{K}$ and an initial image set $K(0)$, design a feedback controller $u(t) = \gamma(K(t))$ based on current image set $K(t)$ such that $dh(K(t), \hat{K}) \to 0$ as $t \to \infty$.

To solve this problem, the mutation equation should be first derived, which requires to determine $\varphi(x(t), u(t))$. In the visual servoing case, $x(t)$ will be the trajectory for individual pixel in the image. Therefore, we can denote $x = [x_1, x_2, x_3]^{\mathrm{T}}$ where x_1 and x_2 represent the pixel position and x_3 is the pixel intensity. The control input $u(t)$, if a kinematic model is considered, will be the camera's spatial velocity which consists of three linear velocity components and three angular velocity components; let $u(t) = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^{\mathrm{T}}$. Based on the definition of $\varphi(x(t), u(t))$, all we need to do is to find the relation $\dot{x}(t) = \varphi(x(t), u(t))$. In fact, $\dot{x}_1(t)$ and $\dot{x}_2(t)$ are related to $u(t)$ by the interaction matrix [1]. In what follows, $\varphi(x(t), u(t))$ will be obtained based on the interaction matrix.

Under invariant lighting condition, the projection of a 3D point in the environment onto the image plane will have a constant intensity; therefore, we have $\dot{x}_3(t) = 0$. Suppose the camera is calibrated, i.e., the intrinsic parameters such as skew factor or pixel aspect ratio are known. Then with the perspective projection model, any point in the image plane with coordinates x_1 and x_2 is related to the corresponding 3D point P in the environment with coordinates $P = [p_x, p_y, p_z]^{\mathrm{T}}$ by:

$$x_1 = \lambda p_x / p_z, \quad x_2 = \lambda p_y / p_z, \quad (11)$$

where λ is the focal length. Without loss of generality, we can assume $\lambda = 1$. By direct differentiation Eq. (11) and considering $\dot{x}_3 = 0$, we have:

$$\dot{x}(t) = Lu(t) \quad (12)$$

where

$$L = \begin{bmatrix} -1/p_z & 0 & x_1/p_z & x_1 x_2 & -(1+x_1^2) & x_2 \\ 0 & -1/p_z & x_2/p_z & 1+x_2^2 & -x_1 x_2 & -x_1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that L is obtained by adding a third row to the traditional interaction matrix to consider an extra intensity element x_3 in x. Similar to the usual visual servoing, the depth p_z should be determined, whose value can be directly estimated from the feature variations in different images [15]. In this paper, however, we consider p_z to be a constant in order to simplify the analysis. In this case, L only depends on x. Based on Eq. (12), we have $\varphi(x(t), u(t)) = L(x(t))u(t)$.

Then the tube $K(t)$ for image set evolution induced from $\varphi(x(t), u(t))$ is:

$$K(t) = T_\varphi(t, K_0) = \{x(t) : \dot{x} = \varphi(x(t), u(t)), x(0) \in K_0\} \quad (13)$$

where K_0 is the initial image set. With this $K(t)$, Eq. (7) is satisfied; therefore, the controlled mutation equation is:

$$L(x(t))u(t) \in \mathring{K}(t) \quad (14)$$

To simplify the notation, t will be omitted in the following discussion, and above equation is denoted as $L(x)u \in \mathring{K}$.

Because $\varphi(x,u) = L(x)u$ is linear in u, the same controller derived in [10] can be used. To obtain the controller, the following Lyapunov function candidate is defined [10]:

$$V(K) = \int_K d^2_{\hat{K}}(x)dx + \int_{\hat{K}} d^2_K(x)dx \quad (15)$$

Note that if we define the error as $e = dh(K, \hat{K})$, then the Lyapunov function candidate is usually defined as $e^2/2$. With this candidate, however, the possibility of differentiation is unknown. For the Lyapunov function candidate in Eq. (15), we can verify that if $V(K) \to 0$, then $e \to 0$. In fact, if $V(K) \to 0$, we have both $d_{\hat{K}}(x) \to 0$ with $x \in K$ and $d_K(x) \to 0$ with $x \in \hat{K}$ because the distance is nonnegative. Then $d(K, \hat{K}) \to 0$ and $d(\hat{K}, K) \to 0$. The definition of Hausdorff distance implies $e \to 0$.

Note that the Lyapunov function candidate in Eq. (15) is different from traditional Lyapunov function in vector space because it's a function of a time varying set. In fact, it is called shape Lyapunov function [16]. Based on the Lyapunov function candidate in Eq. (15), we can have the following theorem, whose proof is given in [10].

Theorem 1 [10]: For the system $L(x)u \in \mathring{K}(t)$ with $x \in \mathbb{R}^m$, $L(x) \in \mathbb{R}^{m \times n}$, $u \in \mathbb{R}^n$, and $K(t) \subset \mathbb{R}^m$, the following controller can locally exponentially stabilize it at $\hat{K}$:

$$u(t) = \gamma(K) = -\alpha A(K)^+ V(K) \quad (16)$$

where $\alpha > 0$ is a gain factor and $A(K)^+$ is the Moore-Penrose pseudoinverse of $A(K) \in \mathbb{R}^{1 \times n}$ defined by:

$$A(K) = \int_K d^2_{\hat{K}}(x) \sum_{i=1}^{m} \frac{\partial L_i}{\partial x_i} dx + 2\int_K [x - \Pi_{\hat{K}}(x)]^{\mathrm{T}} L(x) dx - 2\int_{\hat{K}} [x - \Pi_K(x)]^{\mathrm{T}} L(\Pi_K(x)) dx$$

where $L_i\,(i = 1, 2, \ldots, m)$ are the m row vectors in matrix L, and $\partial L_i / \partial x_i$ is also a row vector with the same dimension.

IV. Application to Nanomanipulations

Atomic Force Microscopy (AFM) is one of the primary tools for imaging and manipulating matters at nanoscale. As shown in Fig. 2, the AFM consists of a piezo-actuated cantilever with a sharp tip at its end. Once the tip interacts with the sample surface, the atomic force will cause a deflection of the cantilever which will be recorded by a position sensitive device (PSD). The amplitude of deflection represents the height of sample point. In this way, the topography of the sample surface can be obtained [17].

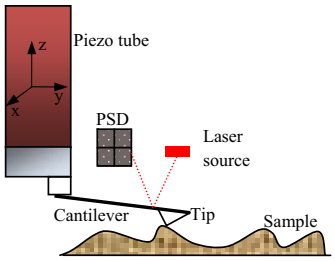
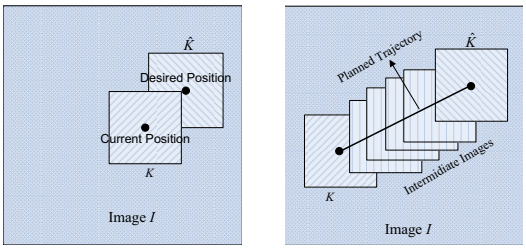


Fig. 2. Principle of Atomic Force Microscopy

Traditionally, AFM is used as a high resolution imaging tool for sample topography interrogating. In recent years, by considering the cantilever as a micro robotic arm, the AFM is used for manipulation to mechanically push, pull, or cut structures at nanoscale [18]. Due to the invisibility of the nano world, the manipulation is usually achieved in a scan-manipulation-scan fashion. First of all, an area of interest is scanned and the resulting image is used to identify the objects intended to be manipulated. Based on the location of the object in the image, a control signal is designed and applied to the AFM, leading to an open loop movement for the cantilever. Finally, the area of interest is scanned again to verify the manipulation results.

According to above process, the fundamental requirement for successful nanomanipulation is precise point to point motion, i.e., the horizontal position for the AFM tip should be controlled precisely enough. Nevertheless, the movement induced vibration, low frequency creep effect, thermal drift, and inherent hysteresis for piezoactuator make the desired position accuracy difficult to achieve [19]. Although position sensors can be added to provide state feedback, they can only measure pizeotube's position instead of tip's position; the large sensor noise at such precision level also makes the sensor incapable of providing valuable feedback. Therefore, sensor deficiencies exclude the use of standard feedback control methods.

To address these problems, the model inversion based feedforward compensation is always used [20]. Although this method provides satisfactory results in some cases, the identification process is quite complicated, and the performance relies on the accuracy of the forward model [21]. Several other methods such as combined feedforward/feedback and H_∞ can also be employed to enhance the performance [22].



(a) First case (b) Second case

Fig. 3. Illustration of two cases for visual servoing based nanomanipulation

Visual servoing provides another approach for manipulation in the nano world. In fact, the AFM can be considered as a single pixel image sensor with two translational degree of freedoms, and an image is obtained by sequentially moving the AFM tip on the sample surface. Based on this idea, the visual servoing based nanomanipulation can be implemented in two cases. For both cases, suppose an area of interest is scanned and an image I is obtained, from which the desire image set can be specified.

The first case is when the desired AFM tip position is sufficiently close to the current tip position. The goal image $\hat{K}$, containing the desired tip position in the image, can be specified in image I. The current image K is obtained by using the AFM for a local scan around its current tip position [23]. Once K is obtained, the current tip position can be obtained by comparing K with I using image registration methods such as template matching. This case is shown in Fig. 3(a), where the current and desired positions are shown as solid circles and the corresponding images are shown as squares. Then the controller in Eq. (16) can be used to steer the AFM tip to the desired position. Note that during the implementation, when the tip arrives an updated position, a local scan should be performed to obtain an updated image.

The second case corresponds to when the desired AFM tip position is far away from the current position. In this case, a series of intermediate images, with their image centers along a planned trajectory between the current and desired positions, can be used as the intermediate goal images. The consecutive intermediate images should be sufficiently close so that the same approach for the first case can be employed. This case is shown in Fig. 3(b), where four intermediate images along a straight line trajectory are shown.

The nanomanipulation discussed above can be considered as a special case with only two translational velocity components v_x and v_y as the inputs. Without loss of generality, assume $p_z = 1$, then we can obtain $\varphi(x,u)$ from Eq. (12) as:

$$\varphi(x,u) = Lu$$

where

$$L = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}$$

is a constant matrix. In this case, $A(K)$ in the controller given

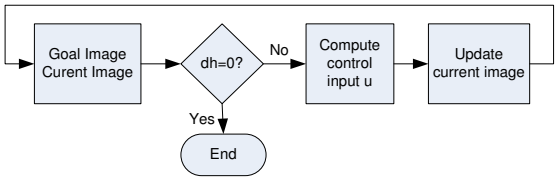


Fig. 4. Simulation Flow chart

by Eq. (16) is simplified to:

$$A(K) = 2\{\int_K [x - \Pi_{\hat{K}}(x)]^{\mathrm{T}} dx - \int_{\hat{K}} [x - \Pi_K(x)]^{\mathrm{T}} dx\} L$$

It should be noted that the scanning images have been used to obtain the parameters for the forward model [20] and compensate the dynamic effects [24]. However, to control directly with the image information is, to the best of our knowledge, never considered before. On the other hand, our approach is very similar to the vision guided robot navigation based on image memory [25]; however, our approach is different because no feature is used.

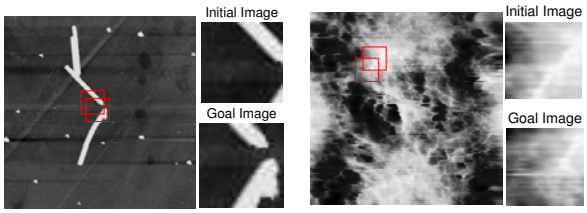
V. Simulation Results

In this section, we validate our controller using AFM images for the above two cases. For the first case, the simulation flow chart is shown in Fig. 4. First of all, a goal image $\hat{K}$ and a current image K, sufficiently close to each other, will be chosen from a given whole image I. Note that in real AFM implementation, both $\hat{K}$ and its position in I are specified, but K is obtained by a local scan and its position is derived by a image registration using K and I. In our simulation, it's assumed that both K and its position are known in advance. Based on $\hat{K}$ and K, the control input u is calculated, which is used to update the position for K. Although the control signal is the translational velocity v_x and v_y, it can be considered as displacements if the time intervals for each iteration are the same. Therefore, an updated K is obtained by moving K in I with control input as the displacements. Note that in AFM implementation, we should move the AFM tip to the updated position and perform a local scan to update K. Using the updated K, the same process is performed again to obtain a new control signal. The program terminates until $dh(K,\hat{K}) = 0$.

The simulation is performed based on two gray scale images with size 256×256 pixels as shown in Fig. 5. The image in Fig. 5(a) has a scan size $25\mu m$ and is scanned for nanowire manipulation, while the image in Fig. 5(b) has a scan size $32\mu m$ and is scanned for hacat cell manipulation. The initial and goal images, with size 30×30 pixels, are labeled in original image and also enlarged on the right.

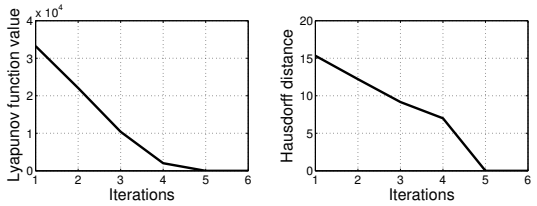
For the nanowire image, the pixel distance between initial and goal images is ten pixels horizontally and seven pixels vertically. The simulation results for this image are shown in Fig. 6(a), where both values for Lyapunov function and Hausdorff distance for all the iterations are shown. From the figure, both values decrease monotonically, and after five iterations, the current image coincides with the goal image.

For the hacat cell image, the pixel distance between initial and goal images is ten pixels horizontally and fifteen pixels vertically. The simulation results shown in Fig. 6(b) demonstrate that the control goal can be achieved. Although the Lyapunov function value decreases, the Hausdorff distance increases at the third iteration. This is because the decrease of Lyapunov function cannot guarantee the decrease of Hausdorff distance, but if the Lyapunov function approaches zero, the Hausdorff distance must also approach zero as discussed in section III.

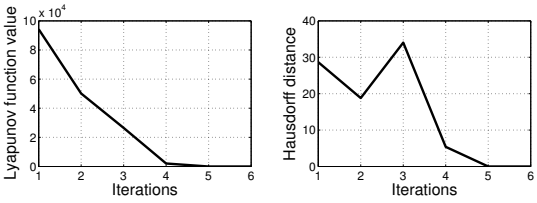


(a) Nanowire AFM images (b) Hacat cell AFM images

Fig. 5. Simulation images for the first case



(a) Results for nanowire AFM images



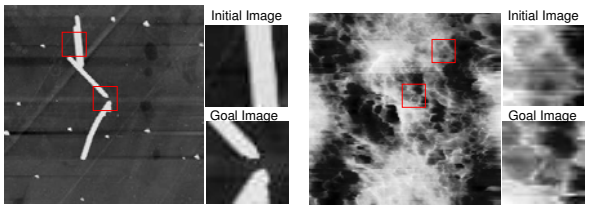
(b) Results for hacat cell AFM images

Fig. 6. Simulation results for the first case

For the second case when the initial image and goal image are far away from each other, a straight line trajectory is used to generate the intermediate images. To guarantee the convergence at those intermediate goal images, the pixel distance between two consecutive images is chosen to be four pixels vertically, while the horizontal pixel distance is obtained from the equation of the line trajectory. The same images, as shown in Fig. 7, are used for the simulation, where the initial and goals images are also shown. For both simulations, the pixel distance between the initial and goal images is 40 pixels horizontally and 60 pixels vertically. Therefore, the number of intermediate goal images is 14.

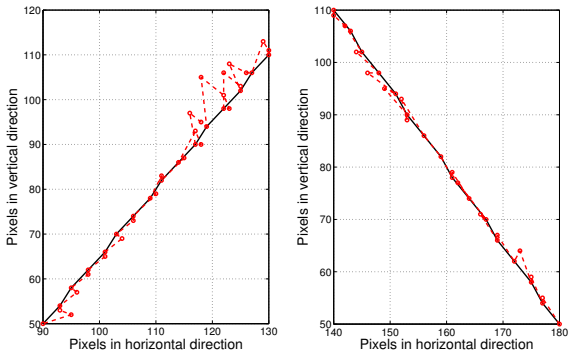
The simulation results for both images are shown in Fig. 8, where the center positions in pixels for the desired trajectory and the true trajectory are plotted. The end points for the black solid lines are the center positions for the intermediate

goal images, while the end points, shown as small circles in the figure, for the red dashed lines are the center positions for the images obtained by servo control. From the figure, we can see that although the true positions may deviate from the desired positions sometimes, the final goal position can be achieved. For the nanowire image, a total number of 38 iterations is performed to arrive the goal image, which means an average of 2.5 iterations from one intermediate goal image to the next one, while for the cell image, the total number is 29 and the average number is thus 1.9.



(a) Nanowire AFM images (b) Hacat cell AFM images

Fig. 7. Simulation images for the second case



(a) Results for nanowire images (b) Results for hacat cell images

Fig. 8. Simulation results for the second case

VI. Conclusions

In this paper, the framework of using mutational analysis for featureless visual servoing is reinvestigated. Different from existing featureless methods, this set evolution based approach considers the image as a set, and the controller is designed based on the distance between two image sets. The simplified controller with two translational degree of freedoms is validated using AFM images. If the initial image and goal image are close enough, the controller can successfully steer the initial image to the goal image. On the other hand, if they are far apart, a set of intermediate goal images will be obtained with consecutive images sufficiently close to each other. Then the initial image can also be steered to the goal image successfully. The method presented in this paper can be considered as a new approach for precise nanomanipulations, and the implementation on an AFM system will be performed in the future.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[2] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[3] E. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robot. Auton. Syst.*, vol. 52, no. 1, pp. 53–70, 2005.

[4] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. Journal of Computer Vision*, vol. 37, no. 1, pp. 7–20, 2000.

[5] V. Kallem, M. Dewan, J. P. Swensen, G. D. Hager, and N. J. Cowan, "Kernel-based visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, 2007, pp. 1975–1980.

[6] C. Collewet, E. Marchand, and F. Chaumette, "Visual servoing set free from image processing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, 2008, pp. 81–86.

[7] A. Censi, S. Han, S. B. Fuller, and R. M. Murray, "A bio-plausible design for visual attitude stabilization," in *Proc. IEEE Int. Conf. Decision and Control*, Shanghai, China, 2009, pp. 3513–3520.

[8] S. Han, A. Censi, A. D. Straw, and R. M. Murray, "A bio-plausible design for visual pose stabilization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Taipei, Taiwan, 2010, pp. 5679–5686.

[9] A. Dame and E. Marchand, "Improving mutual information-based visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, Alaska, USA, 2010, pp. 5531–5536.

[10] L. Doyen, "Mutational equations for shapes and vision-based control," *Journal of Mathematical Imaging and Vision*, vol. 5, no. 2, pp. 99–109, 1995.

[11] J. P. Aubin, *Mutational and Morphological Analysis: Tools for Shape Evolution and Morphogenesis*. Birkhäuser, 1998.

[12] T. Lorenz, *Mutation Analysis: A Joint Framework for Cauchy Problems in and Beyond Vector Spaces*. Springer, 2010.

[13] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[14] S. Nutanong, E. H. Jacox, and H. Samet, "An incremental hausdorff distance calculation algorithm," *Proc. VLDB Endow.*, vol. 4, pp. 506–517, May 2011.

[15] F. Chaumette and S. Hutchinson, "Visual servo control part II: Advanced approaches," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, 2007.

[16] L. Doyen, "Shape laypunov functions and stabilization of reachable tubes of control problems," *Journal of Mathematical Analysis and Applications*, vol. 184, no. 2, pp. 222–228, 1994.

[17] S. M. Salapaka and M. V. Salapaka, "Scanning probe microscopy," *IEEE Control System Magzine*, vol. 28, no. 2, pp. 65–83, 2008.

[18] A. A. G. Requicha, "Nanorobots, NEMS, and nanoassembly," *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1922–1933, 2003.

[19] D. Croft, G. Shed, and S. Devasia, "Creep, hysteresis, and vibration compensation for piezoactuators: Atomic force microscopy application," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 1, pp. 35–43, 2001.

[20] B. Mokaberi and A. A. G. Requicha, "Compensation of scanner creep and hysteresis for AFM nanomanipulation," *IEEE Trans. on Automation Science and Engineering*, vol. 5, no. 2, pp. 197–206, 2008.

[21] G. M. Clayton, S. Tien, K. K. Leang, Q. Zou, and S. Devasia, "A review of feedforward control approaches in nanopositioning for high-speed SPM," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 131, no. 6, p. 061101 (19 pages), 2009.

[22] D. Y. Abramovitch, S. B. Andersson, L. Y. Pao, and G. Schitter, "A tutorial on the mechanisms, dynamics, and control of atomic force microscopes," in *Proc. of American Control Conference*, New York City, USA, 2007, pp. 3488–3502.

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

[24] G. M. Clayton and S. Devasia, "Conditions for image-based identification of SPM-nanopositioner dynamics," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 4, pp. 405–413, 2009.

[25] A. Remazeilles and F. Chaumette, "Image-based robot navigation from an image memory," *Robot. Auton. Syst.*, vol. 55, no. 4, pp. 345–356, April 2007.