Accurate 3D-Positioning in a SEM through Robot Calibration

Valérian Guelpa, Andrey V. Kudryavtsev, Nadine Le-Fort Piat, Sounkalo Dembélé
Automation and Micro-Mechatronics Systems Department
FEMTO-ST Institute, UMR CNRS 6174, ENSMM, University of Bourgogne-Franche-Comté, Besancon, France
valerian.guelpa@femto-st.fr, sounkalo.dembele@femto-st.fr

***Abstract*—With growing trend of miniaturization, new challenges in microrobotics have appeared. In particular, the complexity of microworld comes from the fact that visual sensors such as Scanning Electron Microscope have a very different principles of image formation in contrast with classical cameras, and their field of view stay very limited. Moreover, usually, the kinematic model of the robots used are not well defined. The consequence of both properties is that even a small movement of the robot arm leads to a huge object displacement comparing to the size of the viewed area.**

This paper develops a procedure allowing to perform object rotation while keeping it at the same 3D-position in open loop. Such performance is achieved by a method of robot calibration based on visual servoing and autofocus inside SEM. This kind of properties is required for manipulation and 3D reconstruction inside SEM.

***Index Terms*—Calibration, SEM, positioning, autofocus**

I. Introduction

Vision is a powerful tool to implement robotic tasks at micro-scale, such as micro-manipulation or characterisation. However the need of a high magnification induce difficulties to constitute sequences of images. In order to acquire it, a user may have to spend hours of work which is mostly due to the problem of maintaining the object in the field of view and in focus. Consider the following example: the sample is about 100 µm size and it is located somewhere on the robotic stage (e.g. a robotic arm), several centimeters from the rotation center. Even if an in-plane rotation of 3 degrees is applied, the object may be millimeters away from the initial position. Hence, an operator needs to relocate the object and the situation repeats itself for every new image. Moreover, the high magnification involve to mostly consider affine cameras, among which Scanning Electron Microscopes (SEM).

Indeed, a distortion-free parallel projection model (affine camera) may be used for description of image formation in SEM [1]. SEM image is formed by gathering the electrons reflected from the surface of the sample. Such images of a sample with an arbitrary topography are of excellent quality and high resolution, provided that it has a very big depth of field that may achieve hundreds of micrometers. However, as with every affine camera, the depth coordinate is lost during the image formation, because an affine camera model assumes that all projection rays are parallel to each other. Moreover, the projection of the object is independent of translations in the plane perpendicular to the optical axis. Thus, even if the use of SEM opens up great opportunity to do microrobotic task, new methods have to be developed to exploit its particularities.

Firstly, the focus information have to be considered. When an object performs an in-plane rotation, it leaves the field of view which can be compensated by in-plane translations. However, in the case of more complex out-of-plane rotation, the displacement of object center is characterized by three coordinates, adding translation along the optical axis. As it was mentioned previously, this last translation is not taken into account by the projection process in case of an affine camera. Yet, it has to be considered as SEM has limited depth of field, an area in which the object stays in focus. Obviously, blurry images is not a convenient start point for positioning by vision. Therefore, it is important to *maintain* the object in focus while it is moving. We will refer to this problem as autofocus. However, such autofocus controls only one degree of freedom, which is the distance from camera to object, so it is important to combine it with a visual servoing to control the position of the object along the two other directions.

Such method could theoretically be used for a real-time application, but could be not efficient, notably for fast or complex tasks. So the precise knowledge of the robot kinematic model is often required. In this paper, we propose to use point-link calibration inside SEM, the methods presented in Section II. Using such method, it is possible to accurately calibrate a robot, only thanks to internal sensors and the SEM image, by maintaining an object of interest in the same point in space (without considering its orientation) for some robot configurations. To do this, the Section III shows how to combine visual servoing and autofocus. Finally, Section IV illustrates the performances of the method, by the observation of the micrometric displacements of an object during a rotational open-loop displacement of the calibrated robot inside SEM.

II. Robot calibration

A. Robotic system used

The studied system is shown in Fig. 1. Inside a vacuum chamber of SEM, a serial-link manipulator, further referred as robot, is installed. As a usual robotic arm, it comprises a

This work has been supported by Rgion de Bourgogne-Franche-Comt, Labex ACTION project (cantract ANR-11-LABX-0001-01), Equipex ROBOTEX project (contract ANR-10-EQPX-44-01) and the cross border project CITHADEL of Interreg France-Switzerland program through the European regional development fund.

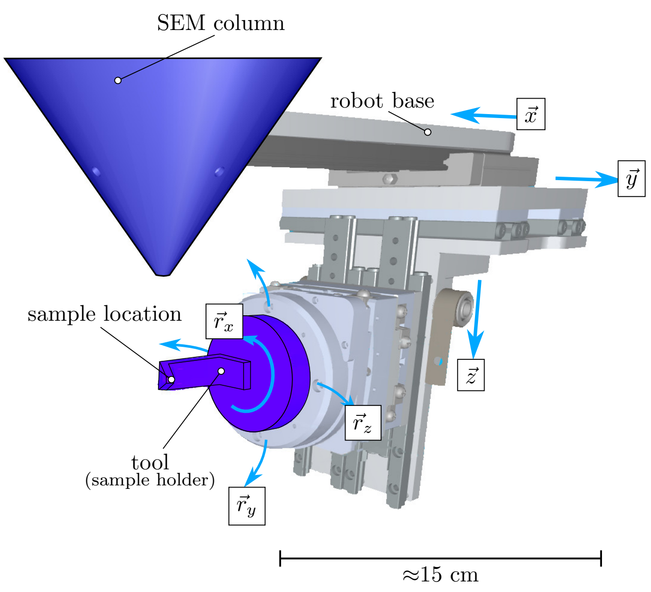

Fig. 1. View of the CAD models of the 3P3R robot and the SEM column, inside the SEM vacuum chamber.

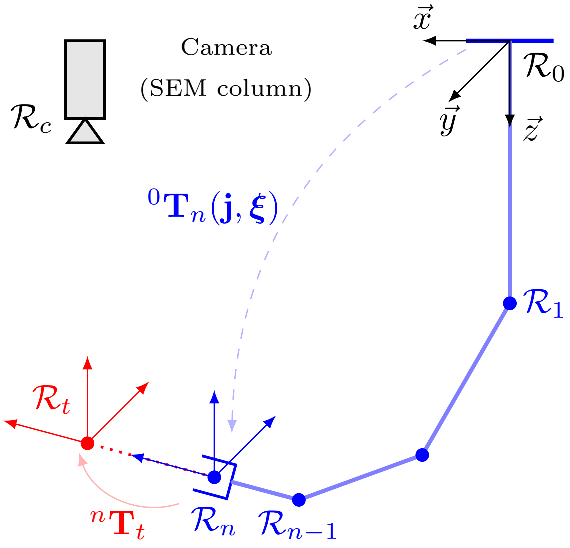

Fig. 2. Model of the robot installed inside SEM chamber.

chain of rigid links and joints. One joint j has one degree of freedom represented either as a translational movement for prismatic joints or rotational movement for revolute joints. In our configuration, the robot is PPPRRR or 3P3R, which means that it has three prismatic ($\vec{x}, \vec{y}, \vec{z}$) and three revolute joints. All joints are equipped with position sensors. Prismatic joints are actually three micropositioning stages mounted together. Regarding revolute joints, first two are implemented as a goniometer ($\vec{r}_y, \vec{r}_z$) and the last one is a classic rotation block ($\vec{r}_x$). The base of the robot is fixed on the chamber ceiling and the frame associated to it is also fixed and denoted as $\mathcal{R}_0$. Other frames $\mathcal{R}_i$ with $i \in (1, 2, .., 6)$ correspond to six robot joints (see Fig. 2). The end of the robot which is free to move holds the tool or end-effector which is a sample holder in our case. The frame associated with the center of the object (around which the rotation has to be done) is noted $\mathcal{R}_t$.

The position of the end-effector clearly depends on the state of each joint. The pose may be computed as a series of transformations involving every link from the base frame to the tool frame:

$$ {}^0\mathbf{T}_t = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \cdots {}^{n-1}\mathbf{T}_n {}^n\mathbf{T}_t \tag{1} $$

where ${}^i\mathbf{T}_k$ denotes the 4×4 transformation matrix between $\mathcal{R}_k$ to $\mathcal{R}_i$ with $k, i \in (0, 1, .., n)$ and $n = 6$.

Each transformation matrix in (1) includes geometric parameters of the robot: lengths between links, angles between stages, etc. Once the robot has been designed, the theoretical estimation of forward kinematics is obtained from its CAD model and position of the tool may be calculated. However, due to the process of assembly and manufacturing, these base theoretical parameters are never the true ones which results in low accuracy of object positioning, and the situation is even worse at small scales.

The accuracy of the robot may be improved by following a calibration procedure with the goal of refinement of robot geometric parameters. In the following sections, these parameters are defined and the corresponding calibration procedure is presented.

B. Point-link calibration

Calibration procedure implies the refinement of geometric parameters of forward kinematics and tool transformation. The vector of parameters $\boldsymbol{\xi}$ can be decomposed for each transformation separately. For transformations between the links of the robot, the Khalil and Kleinfinger notation [2] is used. The transformation matrix between two joints j_{i-1} and j_i can be written as follows (3):

$$ {}^{i-1}\mathbf{T}_i = \text{rot}(\vec{y}, \beta_i) \cdot \text{rot}(\vec{x}, \alpha_i) \cdot \text{trans}(\vec{x}, d_i) \cdot \text{rot}(\vec{z}, \theta_i) \cdot \text{trans}(\vec{z}, r_i) \tag{2} $$

And for tool transformation, corresponding to the position of the TCP (Tool Center Point):

$$ {}^n\mathbf{T}_t = \text{trans}(\vec{x}, x_t) \cdot \text{trans}(\vec{y}, y_t) \cdot \text{trans}(\vec{z}, z_t) \tag{3} $$

The theoretical values of robot forward kinematics are presented in Table I. It should be noted that for revolute joints, angles θ_i are replaced with joint variables (the values given by sensors). The same is true for r_i and prismatic joints.

The standard calibration procedure comprises the following steps. First, the tool position $\hat{\mathbf{x}}_c$ is measured by a precise external sensor for many different robot configurations $c \in (1, 2, .., N_c)$. Typically, in the experience, N_c varied from 25 to 50. The joint variables $\mathbf{j}_c$ are also stored. Therefore, two values should match each other: tool position measured by an external sensor $\hat{\mathbf{x}}_c$ and the same computed from forward kinematics and tool transformation $\mathbf{x}_c(\mathbf{j}_c, \boldsymbol{\xi})$. The goal is then

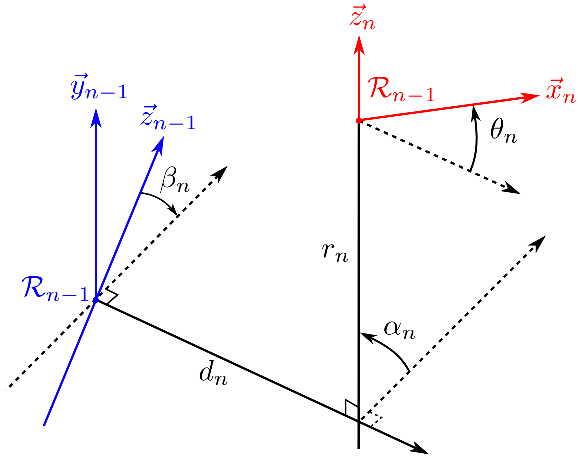

Fig. 3. Transformation between robot links.

TABLE I
THEORETICAL MODEL OF ROBOT FORWARD KINEMATICS (INITIAL VALUES OF PARAMETERS $\boldsymbol{\xi}_0$).

	β_i degrees	α_i degrees	d_i mm	θ_i degrees	r_i mm
j_1	0	0	0	0	j_1
j_2	0	-90.00	0	90.00	j_2
j_3	0	-90.00	0	180.00	j_3
j_4	0	0	0	j_4	0
j_5	0	-90.00	0	$j_5-90.00$	0
j_6	0	-90.00	0	j_6	-41.6
		x mm		y mm	z mm
TCP		0		-5	50

to find such values of parameters that minimize the difference between them:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{c=1}^{N_c} \|\hat{\mathbf{x}}_c - \mathbf{x}_c(\mathbf{j}_c, \boldsymbol{\xi})\|^2 \tag{4}$$

However, as it was already mentioned, it is difficult to find a sensor allowing to provide accurate position measurements at microscale. Therefore, another calibration method that relies only on joint parameters could be used: the *point-link calibration* [3]. The idea behind this method is to place the robot in many different configurations and ensure that for every configuration the position of the tool is the same (with an arbitrary orientation):

$$\forall(\mathbf{j}_a, \mathbf{j}_b) : \mathbf{x}(\mathbf{j}_a, \boldsymbol{\xi}) = \mathbf{x}(\mathbf{j}_b, \boldsymbol{\xi}) \tag{5}$$

Therefore, the calibration problem transforms into:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{a=1}^{N_{c-1}} \sum_{b=a+1}^{N_c} \|\mathbf{x}(\mathbf{j}_a, \boldsymbol{\xi}) - \mathbf{x}(\mathbf{j}_b, \boldsymbol{\xi})\|^2 \tag{6}$$

which can be solved by many different optimization algorithms, from global optimization to Levenberg-Marquardt algorithm, that is the most common approach for such tasks.

III. MAINTAINING OBJECT LOCATION

A. Visual servoing

The position of the object center $\mathbf{Q}_t$ (tool) is determined by its three coordinates $(q_{tx}, q_{ty}, q_{tz})^\top$ and the goal is to maintain it in the center of the image for any changing robot configuration. This imply the need to measure the relevant displacement and to compensate it by moving the robot accordingly. The depth coordinate is measured using the autofocus technique presented above. For two other coordinates, visual servoing keep the object in the center of the image. Visual servoing is a technique allowing to control a robot by using the feedback from a visual sensor [4]. In order to measure the displacement of one point, the homography-based approach is used.

Assume the starting point of the algorithm is an in-focus image with the object to track located in its center. This image is used to extract the features that are then tracked using KLT algorithm in all following images [5]. Object center is not a feature itself, thus, in order to track it, one needs to find the transformation between the current image and first image and then find the projection the object center in this current image. This transformation is a homography that defines the following relation between two corresponding points:

$$\mathbf{q}'_t = \mathbf{H}\mathbf{q}_t \tag{7}$$

where $\mathbf{H}$ is a 3×3 homography matrix. Therefore, once $\mathbf{H}$ is estimated, the new position of the object center is easily found using (7). The estimation of homography is done from tracked features using linear algorithm inside RANSAC scheme [6].

Once the homography is estimated, the error between current object position and the desired one (image center) is written as:

$$\mathbf{e}(t) = \mathbf{q}_t(t) - \mathbf{c} \tag{8}$$

where $\mathbf{c}$ is the center of the image and $\mathbf{q}_t(t)$ is a time-varying current position of the tool. It is important to note that the error $\mathbf{e}$ is computed in camera frame $\mathcal{R}_c$ and, in order to express it in the robot base frame to find the joint speed, it has to be multiplied by the transformation matrix ${}^0\mathbf{T}_c$. This matrix is known from CAD model of SEM vacuum chamber. Finally, the control law for maintaining the object in the center of the image is written as:

$$\dot{\mathbf{j}} = -\lambda {}^0\mathbf{T}_c \mathbf{e} \tag{9}$$

where $\dot{\mathbf{j}}$ is the vector of joint speeds and λ is a proportional gain.

B. Autofocus

By using 2D visual servoing, it is possible to assure that the object always stays in the center of the image. However, if the object move by an out-of-plane displacement, its image will be unfocused and blur. Even in the case of a visual servoing usable on an unfocused object, its accuracy will be reduced. Moreover, the object could simply disappear of the image, what makes the visual servoing algorithm unusable. As a result, it is important to use an autofocus method.

Autofocus is a subject already studied in the specific case of electron microscopy [7]–[9]. The algorithm presented below is a new version of the algorithm presented in [10], initially developed for keeping an object in focus, but it also allows to estimate the depth coordinate precisely enough for such applications as automatic manipulation, assembly or robot calibration.

First, a definition of the sharpness has to be choose. This sharpness, used inside a sharpness function, will be significant of the depth variation, or equivalently from focal distance variation or working distance in case of SEM. Some definition could be used, as Pertuz *et al.* explained in [11] for general cases, but also Rudnaya *et al.* in [9] in the specific case of microscopy domain. In the present case, the normalized variance was selected:

$$S(\mathbf{I}) = \frac{1}{MN}\frac{1}{\mu}\sum_{M}\sum_{N}(\mathbf{I}(q_x, q_y) - \mu)^2 \tag{10}$$

where $S(\mathbf{I})$ is the sharpness of image $\mathbf{I}$, μ is the mean of intensity values, M and N are image width and height, respectively.

The implement an autofocus algorithm, this sharpness fonction is used as an objective function for an optimisation algorithm. In this way, the working distance is the interesting input parameter. So the method has to keep the sharpness $f_n(\xi) = S$ to a maximum value by updating the working distance ξ:

$$\xi_n = \arg\max_{\xi\in\mathbb{R}} f_n(\xi) = \arg\min_{\xi\in\mathbb{R}}(-f_n(\xi)) \tag{11}$$

In the present case, the objective function is not stationary, so optimization algorithms based only on function evaluations, such as the Golden-section search [12] based on the reduction of the interval that contains the maximum, are not suitable. It is better to use methods based on first-order derivative, or gradient-based methods. However neither the sharpness function (nor its derivatives) are directly available. But it is possible to use an approximation of gradient, as provided by the method of backward differences :

$$f'(\xi_n) = \frac{f(\xi_n) - f(\xi_n - \Delta\xi)}{\Delta\xi} \tag{12}$$

Finally, a gradient descent/ascent method could be used:

$$\xi_{n+1} = \xi_n - \alpha f'(\xi_n) \tag{13}$$

with α the learning rate and ξ_{n+1} the estimate of the working distance that would give the best value of image sharpness.

The main limit is now that such method could be unusable in presence of noises, that could be important in the case of our objective function (based on the sharpness of a SEM image). Because of high variations in the gradient value, the working distance could change accordingly and make the focus to be lost. Therefore, a lot of authors worked on improvement for gradient methods, such as *Momentum* [13], *AdaGrad* [14], *RMSProp* [15] or *Adam* [16]. This last is the most interesting in our case, because it allows better filtering of the gradient. It is founded on an adaptive moment estimation, based on the observation of the exponentially weighted moving average of the gradient:

$$\begin{cases} m_{n+2} \leftarrow \beta_1 m_n + (1-\beta_1) f'(\xi_n) \\ v_{n+2} \leftarrow \beta_2 v_n + (1-\beta_2) f'(\xi_n)^2 \\ \xi_{n+2} \leftarrow \xi_n - \alpha \frac{m_{n+2}}{\sqrt{v_{n+2}}+\varepsilon} \end{cases} \tag{14}$$

with m the first moment variable, v the second moment variable and ε a constant allowing to avoid division by zero at first iterations.

Algorithm 1 Autofocus in SEM

```
$\alpha \leftarrow 0.00001$                                ▷ optimization parameters
$\beta_1 \leftarrow 0.6$
$\beta_2 \leftarrow 0.6$
$\Delta\xi \leftarrow 10^{-5}$
$\varepsilon \leftarrow 10^{-8}$
$m \leftarrow 0$
$v \leftarrow 0$
$\xi_{n-1} \leftarrow \xi_0$                               ▷ initial working distance
$f(\xi_{n-1}) \leftarrow f(\xi_0)$                         ▷ initial sharpness
set working distance $\xi_n = \xi_{n-1} + \Delta\xi$       ▷ initialize displacement
while autofocus is activated do
    acquire image $\mathbf{I}_n$
    get sharpness score $f(\xi_n) \leftarrow S(\mathbf{I}_n)$ (Eq. 10)
    estimate gradient $\hat{g} = f'(\xi_n) = \frac{f(\xi_n)-f(\xi_{n-1})}{\xi_n-\xi_{n-1}}$ (Eq. 12)
    $m \leftarrow \beta_1 m + (1-\beta_1)\hat{g}$
    $v \leftarrow \beta_2 v + (1-\beta_2)\hat{g}^2$
    set working distance $\xi \leftarrow \xi + \alpha\frac{m}{\sqrt{v+\varepsilon}}$
end while
```

The final autofocus algorithm is represented in Algorithm 1. It includes a lot of advantages, notably the absence of calibration step, of preliminary model, of training data or focus sweeping of the scene. Moreover the robustness to noise allows to work with high frame rate in SEM (tested between 1 and 5 Hz).

IV. Results

To illustrate the control of the robot in the operational space, the objective is rotating around three axis (with $\vec{r_x}$, $\vec{r_y}$ and $\vec{r_z}$) while also maintaining the position of the TCP in translations. The object observed, and used to defined the TCP, is a little sphere of 10 µm of diameter, putted on a little beam (see Fig. 4). The two are coated by gold.

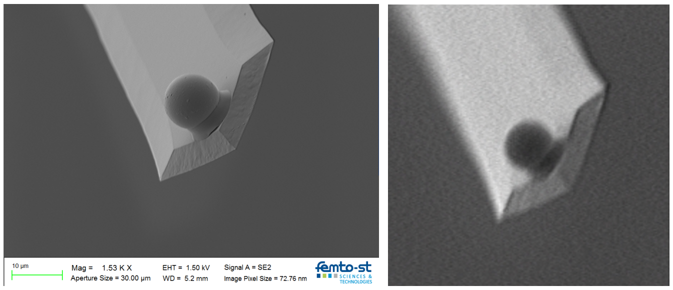

Fig. 4. Object maintened in position during the displacements of the robot: a 10 µm sphere putted on a microbeam. (left) an high-resolution view of the object; (right) a typical image captured during the acquisition for calibration.

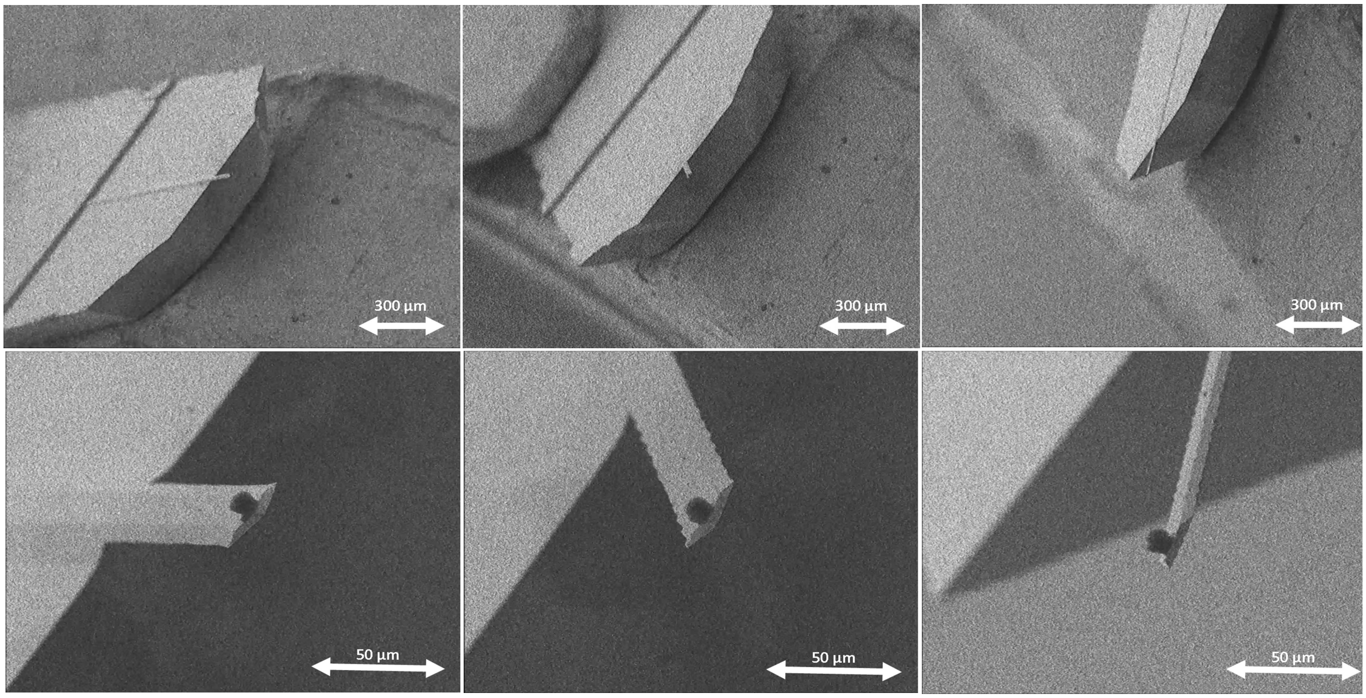

Fig. 5. Image taken during the motion of calibrated robot in open loop, for two magnifications and three positions. The objective is to maintain the sphere at the same point. See the video.

A. Calibration step

To calibrate the robot, about 50 robot configurations have been acquired. To maintain this object, the visual servoing and autofocus algorithms presented above are combined with Matlab/Simulink R2017b, with a frame rate of 1 Hz. The parameters of the SEM are: magnification 373×; aperture size 60 µm; working distance 5.2 mm; image pixel size 299.2 nm; resolution 200×200 pixels (see Fig. 4). The revolute axis are commanded independently, and the translations of the three others axis maintain the object at center.

Next the expression (6) is used to refine the set of parameters $\boldsymbol{\xi}$. The table of calibrated parameters is shown below (see Table II). From the resulting values, one can see that the most important correction was done on the parameter α reflecting the perpendicularity between robot axis. The displacements d and r allowed to correct the center of rotation.

B. Open-loop control

With the calibrated robot it is now possible to perform a rotation around the object center (see Fig. 5 and the joined video). The same object is used (a 10 µm sphere), and the robot receive a sinusoidal consign on the three axis of rotation, with ranges $\Delta\theta_x = 75°$, $\Delta\theta_y = 6°$, $\Delta\theta_z = 7.5°$ (near to the maximal achievable displacements).

Thank to the calibration, the biggest displacement between the TCP and the image center during rotations is near the size of the sphere, so 10 µm (see Fig. 6). This result illustrates that the presented method allows to calibrate a robot inside SEM with a good accuracy, adapted to micro-robotic tasks and notably to keep an object in the field of view during displacements for image acquisitions.

TABLE II
CALIBRATED MODEL OF ROBOT FORWARD KINEMATICS $\boldsymbol{\xi}^*$.

	β_i degrees	α_i degrees	d_i mm	θ_i degrees	r_i mm
j_1	0	0	0	0	j_1
j_2	0	−89.8154	0	89.6759	j_2
j_3	0	−91.2235	0	179.9239	j_3
j_4	−0.3958	1.7134	−0.0002	j_4−0.0768	−0.0002
j_5	0	−90.4253	−0.0496	j_5−89.6437	0.0229
j_6	0	−89.8625	−0.1434	j_6−0.0034	−41.600
		x mm		y mm	z mm
TCP		−0.75364		−6.29653	55.50676

Finally, it may be noted that the small error seems mostly due to the depth estimation: actually, as the object has a spherical form, the autofocus is not performed exactly on the TCP, but on the sphere around it.

V. CONCLUSION

In this paper, a method dedicated for automatic positioning by the use of an affine camera, and its applications in SEM, was presented. The main lock consisted in the problem of

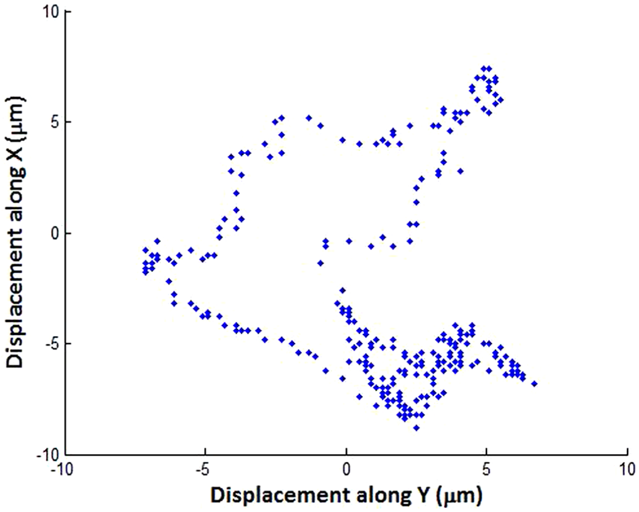

Fig. 6. Displacement of the sphere during the open loop displacement in the plane perpendicular the the optical axis (direct measurement on the video). The error on position maintening stay inferior to 10 µm for rotations equal to $\Delta\theta_x = 75°$, $\Delta\theta_y = 6°$, $\Delta\theta_z = 7.5°$.

keeping the object inside the field of view while performing rotations and the presented algorithm was dedicated to solving it.

A first step refereed to an autofocus method used to solve the variations of sharpness due to sample displacement. The presented method relied on gradient-based optimization and allowed keeping the object with unknown structure in focus at high frame rate, i.e. using very noisy images. The experiments on the SEM validated the algorithm and proved its robustness to the variation of magnification and displacement speed. The working distance was adjusted automatically that served also for depth estimation, for measuring the translation perpendicular to the image plane. A visual servoing algorithm added to the method allowed a positioning along the two others axis in translation.

Next we used a procedure to calibrate a serial robot inside SEM, performing the rotation around the point of our choice. It started with the acquisition of different robot configurations keeping the same position of the tool. The vectors of joint variables corresponding to every configuration were then used in the optimization process that allowed refining of initial theoretical geometric parameters of the robot and the tool transformation. The calibration detected and measured the error in the kinematic model of the robot, with lengths of joints and angles between joints.

Finally, the method was tested inside SEM with a microsphere of 10 µm. After calibration, the robot was controlled in open loop: applying a rotational speed along the three axis, the object stays in the field of view and the shift from the desired position did not exceed 10µm, so approximately an error of the same order as the size of the object.

The paper showed it is possible to control a serial robot in open loop inside SEM with micrometric accuracy, beforehand using a calibration method based on vision. Next this approach will be used for object manipulation, positioning and image acquisition for 3D reconstruction.

References

[1] L. Cui and E. Marchand. Calibration of scanning electron microscope using a multi-image non-linear minimization process. In *IEEE Int. Conf. Robot. Autom. (ICRA'14)*, pages 5191–5196, 2014.

[2] W. Khalil and J. Kleinfinger. A new geometric notation for open and closed-loop robots. In *IEEE Int. Conf. Robot. Autom. (ICRA'86)*, volume 3, pages 1174–1179, 1986.

[3] W. Khalil, G. Garcia, and J.-F. Delagarde. Calibration of the geometric parameters of robots without external sensors. In *IEEE Int. Conf. Robot. Autom. (ICRA'95)*, volume 3, pages 3039–3044, 1995.

[4] F. Chaumette and S. Hutchinson. Visual servo control, part I: Basic approaches. *IEEE Robot. Automat. Mag.*, 13(4):82–90, 2006.

[5] C. Tomasi and T. Kanade. *Detection and tracking of point features*. 1991.

[6] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[7] R. Nishi, Y. Moriyama, K. Yoshida, N. Kajimura, H. Mogaki, M. Ozawa, and S. Isakozawa. An autofocus method using quasi-gaussian fitting of image sharpness in ultra-high-voltage electron microscopy. *J. Microsc.*, 62:515–519, 2013.

[8] Z.M. Wu, D.H. Wang, and F. Zhou. Bilateral prediction and intersection calculation autofocus method for automated microscopy. *J. Microsc.*, 2048:271–280, 2012.

[9] M.E. Rudnaya, R.M.M. Mattheij, and J.M.L. Maubach. Evaluating sharpness functions for automated scanning electron microscopy. *J. Microsc.*, 240:38–49, 2010.

[10] Andrey V. Kudryavtsev, Sounkalo Dembélé, and Nadine Piat. Autofocus on moving object in scanning electron microscope. In *Ultramicroscopy*, pages 42–47, 2017.

[11] S. Pertuz, D. Puig, and M. A. Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46:1415–1432, 2013.

[12] J. Kiefer. Sequential minimax search for a maximum. *Proc. of American Math. Soc.*, 4:502–506, 1953.

[13] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12:145–151, 1999.

[14] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[15] T. Tieleman and G. Hinton. RMSProp, Coursera: Neural networks for machine learning. *Technical report*, 2012.

[16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.