Analysis of CAD Model-based Visual Tracking for Microassembly using a New Block Set for MATLAB/Simulink

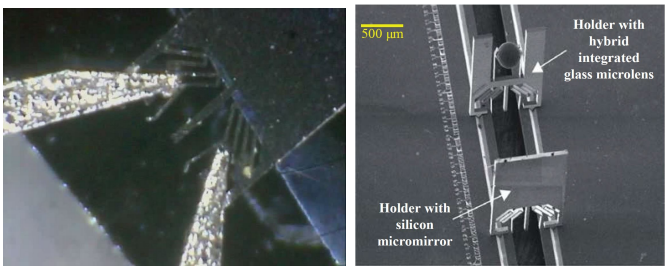
Andrey V. Kudryavtsev, Guillaume J. Laurent, Cédric Clévy, Brahim Tamadazte and Philippe Lutz
FEMTO-ST, AS2M, Univ. Bourgogne Franche-Comté, Univ. de Franche-Comté/CNRS/ENSMM, Besançon, France.
Email: andrey.kudryavtsev@femto-st.fr

***Abstract*—Microassembly is an innovative alternative to the microfabrication process of MOEMS which is quite complex. It usually implies the use of microrobots controlled by an operator. The reliability of this approach has been already confirmed for the micro-optical technologies. However, the characterization of assemblies has shown that the operator is the main source of inaccuracies in the teleoperated microassembly. Therefore, there is a great interest in automating the microassembly process. One of the constraints of automation in microscale is the lack of high precision sensors capable to provide the full information about the object position. Thus, the usage of visual-based feedback represents a very promising approach allowing to automate the microassembly process. The purpose of this paper is to characterize the techniques of object position estimation based on the visual data, i.e. visual tracking techniques from the ViSP library. These algorithms allows to get the 3D object pose using a single view of the scene and the CAD model of the object. The performance of three main types of model-based trackers is analyzed and quantified: edge-based, texture-based and hybrid tracker. The problems of visual tracking in microscale are discussed. The control of the micromanipulation station used in the framework of our project is performed using a new Simulink block set. Experimental results are shown and demonstrate the possibility to obtain the repeatability below 1 μm.**

I. Introduction

The needs for highly integrated, complex and smart MOEMS (Micro-Opto-Electro-Mechanical Systems) keep increasing especially for instrumentation and biomedical tools application fields [1]–[4]. The main lock for these developments relies in microfabrication process that are quite complex. To tackle this problem, several innovative alternatives are emerging. Among them, the use of microrobotic-based cells in order to achieve automatic elementary microassembly tasks.

Several works already established the viability of this approach: they notably show that the key feature relies in the capability of the system to achieve modular and highly accurate assembling, i.e. typically smaller than 5 μm (maximum acceptable error) [5], [6]. Several concepts of microoptical benches to be assembled have also been proposed [7]–[9]. In [10], it is also established that a positioning accuracy smaller than 1 μm can be achieved in teleoperated mode. The operator being the main source of inaccuracies [10], there is a great interest in automating the microassembly process. It implies increasing the throughput yield but also quantifying the main sources of inaccuracies which is of great interest for the design of MOEMS blocks and microrobots, the clean room fabrication and assembly strategies.



(a) (b)

Fig. 1: a) Microcomponent picking using the microgripper; b) example of assembled microoptical bench [10].

The main objective of the described works is to achieve automated assemblies of MOEMS through visual servoing approach and then study and characterize its precision and robustness in the microscale. To achieve this goal, a strategy based on high level closed-loop vision control will be implemented. The studied methods are model-based visual tracking algorithms from the ViSP (Visual Servoing Platform) library, which is able to directly provide the 3D object pose using a single view of the scene [11].

The components to be assembled are displayed in Fig. 1. The assembly consists in the insertion of a holder in the V-groove guiding rails of silicon baseplate using a micromanipulation station that will be described in Section II.

Previous works on the field of automatic microassembly demonstrate the possibility of model-based visual trackers application [12]. However, these strategies can not be employed directly in the present case, because of several constraints. Firstly, all objects of the scene are made of silicon which causes reflections. Secondly, the object contains deformable parts. Finally, the ratio between length and thickness is very high. Thus, the goal of this paper is to implement CAD-based visual tracking techniques and to quantify its performances for further automation of microassembly of MOEMS. For this, first of all, we should discriminate the sources of noises: noises related to the algorithm, influence of the environmental factors and vibrations of the robot structure. Then, the tracker precision will be quantified using a 3D planned path. To simplify the analysis of microassemblies, micromanipulation control and visual tracking, software have been developed in the form of Simulink blocks in C++ language.

TABLE I: Characteristics of the stages comprised in the robot used in a micromanipulation station.

	Product reference	Specifications
Translation stages : XY	M-111-DG *PI Mercury*	Stroke : 15 mm Backlash : 2 µm Min. Inc. Motion : 0.05 µm Unidir. repeatability : 0.1 µm
Z	M-121-DG *PI Mercury*	Stroke : 25 mm Backlash : 2 µm Min. Inc. Motion : 0.05 µm Unidir. repeatability : 0.1 µm
Rotation stage : Θ	SR3610S *SmarAct*	Stroke : 360° Resolution : $< 10\mu$ °

The remainder of this paper is organized as follows: Section II presents the used equipment, micromanipulation station in particular. Section III introduces the software architecture allowing to interface different equipment such as translation and rotation stages, camera, etc. Section IV of this article presents the steps of trackers analysis and performances quantification. Finally, conclusions and prospects are discussed at the end.

II. Experimental setup

For the precise control of position and the alignment of optical path, a 3D microassembly station is used. It comprises a serial robot of 4 degrees of freedom (XYZΘ) with a 4-DOF microgripper [13] and a vision system (Fig. 2). The whole system is mounted on an antivibration table. The characteristics of the robot are represented in Table I.

The choice of the vision system plays a crucial role in quality of measurements. There are several factors that should be taken into account when working at microscale. First of all, a camera should have an important magnification. Secondly, it is very important to have a sufficient depth of field, because it is indispensable to have a focused image in the whole area of object displacement. Getting sufficient depth of field can be particularly challenging with high magnification. The third factor is the working distance: camera should not be located in the robot workspace. The last factor is resolution, which influence directly the amount of image detail, so, the quality of tracking. Therefore, the choice of vision system is a compromise between all presented parameters, especially between the pair magnification/depth of field. The chosen camera is a high resolution camera which allows to reduce magnification preserving a required depth of field. The parameters of the vision system are presented in Table II. The depth of field of presented vision system is about 2 mm, while the visual field is about 4x4 mm^2. The position of the camera in relation to the robot is represented in Fig. 3.

III. Simulink blocks development

As it was mentioned before, the software has been developed in the form of Simulink blocks. Taking into account the fact that the software efficiency (in terms of execution time) is not the main goal of present work, MATLAB/Simulink usage gives a lot of advantage for fast prototyping and numerical

TABLE II: Characteristics of the vision system.

	Product reference	Specifications
Camera : IDS uEye	UI-3480CP *Aptina*	CMOS Rolling Shutter Pixelpitch : 2.2 µm Pixel Class : 5 Megapixel Resolution (h x v) : 2560x1920
Objectif	CVO GM10HR35028	Class : high resolution Focal distance : 50 mm

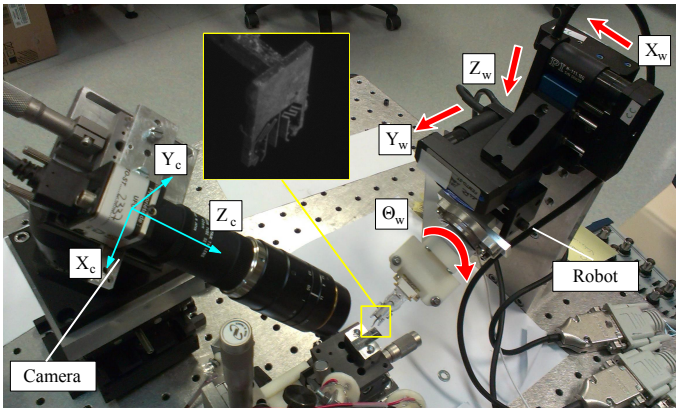
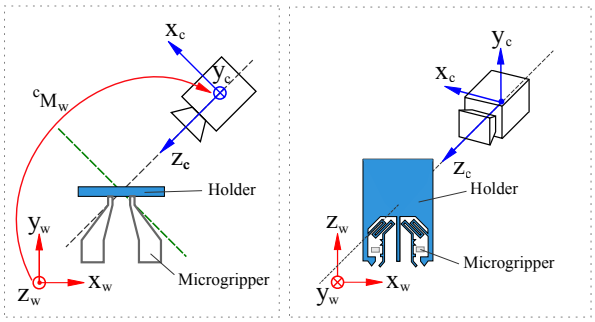


Fig. 2: Configuration of the micromanipulation station with four DOF.

data treatment. In present case the software efficiency is not an issue, because the image acquisition frequency doesn't exceed 15 fps. However, it could be improved by implementing the C++ code in a real-time operating system.

Simulink is a data flow graphical programming language tool for modeling, simulating and analyzing multidomain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It is widely used in automatic control, aerospace engineering, signal processing and computer engineering applications [14]. In many fields, MATLAB/Simulink has already become the



(a) Top view (b) Side view

Fig. 3: Camera position related to the object; x_c, y_c and z_c refer to the coordinates in the camera frame; x_w, y_w and z_w refer to the world frame;

number one simulation language. Furthermore, the creation of custom Simulink blocks is possible using a MATLAB S-function. Custom code may be written in M, in C or in Fortran languages.

The development of software in the form of Simulink blocks has several advantages. Firstly, it allows creating one block which exercises one function, thus, the system becomes modular. Secondly, the fact of working in a Simulink environment allows the use of all already existing blocks, such as mathematical functions, scopes, advanced control laws, filters, etc. Thirdly, it became possible to use the dedicated functions such as "Robotic toolbox" developed by Peter Corke; it is a software package that allows a MATLAB user to readily create and manipulate data types fundamental to robotics such as homogeneous transformations, quaternions and trajectories [15]. Finally, as MATLAB/Simulink is widely used in a field of engineering, the developed blocks can be easily integrated in other projects. For all these reasons, the software for robot control and vision was developed in the form of Simulink blocks.

A. C++ Wrapper for S-function

The development of Simulink blocks using standard C++ language had become possible owing to the EasyLink interface (https://sourcesup.renater.fr/easylink/). It was developed in the FEMTO-ST institute and it allows to use C++ compiler in order to create Simulink blocks. The fact of working with C++ language simplifies the use of external libraries.

B. Blocks for Robot Control

In order to integrate the possibility of robot control in MATLAB/Simulink, a custom block set has been developed. It contains the blocks allowing the entire control of SmarAct(http://www.smaract.de/) and PI Mercury(http://www.physikinstrumente.com/) stages:

- two types of blocks for a stage: one block for position control and one block for speed control. There is a possibility to impose the displacement limits for every movement which allows to secure the station, in particular fragile components such as microgripper;
- one block for microgripper control. It allows to control the two fingers of microgripper either simultaneously or separately;
- one block which provides access to a joystick. It provides a possibility of teleoperation control of the station. The code of this block is based on SDL 2.0 (Simple DirectMedia Layer) library (http://www.libsdl.org/).

All of the blocks of the library are represented in Fig. 5. Fig. 4 represents an example of control of SmarAct stage.

The block set contains also the blocks of 3D model-based tracking that will be described in the following section.

C. Blocks for Tracking

In order to implement the tracking techniques, the C++ ViSP library was used. ViSP standing for Visual Servoing

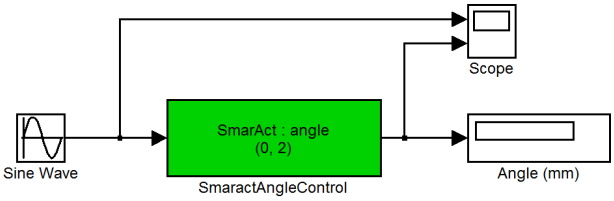


Fig. 4: Usage example of a block of rotation stage angle control

Platform is a modular cross platform library created by INRIA Bretagne Atlantique, Lagadic team. It allows prototyping and developing applications using visual tracking and visual servoing techniques [11].

The tracking techniques studied in this article use a 3D model of the tracked object to provide 3D information of the object position from a monocular image. They can be divided into three main classes described in [16]:

- Edge-based tracking. It relies on the high spatial gradients outlining the contour of the object or some geometrical features;
- Texture-based tracking, which uses texture information for object tracking;
- Hybrid tracking. It is appropriate to track textured objects with visible edges.

For every type of tracker one block was created, thus, three blocks.

IV. Experimental results

In order to be able to estimate the performances of visual trackers at microscale and choose one for further assembly automation, the main influential factors should be qualitatively determined. The present analysis was divided in three steps:

- Trackers analysis in a still frame. It consists in testing visual trackers in a situation, when the video flow from the camera is replaced by a video flow composed of the same repeated image. This experiment allows to quantify the performance of the algorithm separately, as the influence of all other noise sources is removed;
- Trackers analysis with a static object. The pose is estimated in a video flow containing the object while the robot is not moving. It implies that, besides the noise coming from the algorithm, the object is exposed to environmental factors.
- Planned path tracking. It consists in comparing the pose estimation between robot sensors and visual tracker by using a predefined trajectory. The resulting error includes algorithm noise, environmental factors influence and manipulator imperfection.

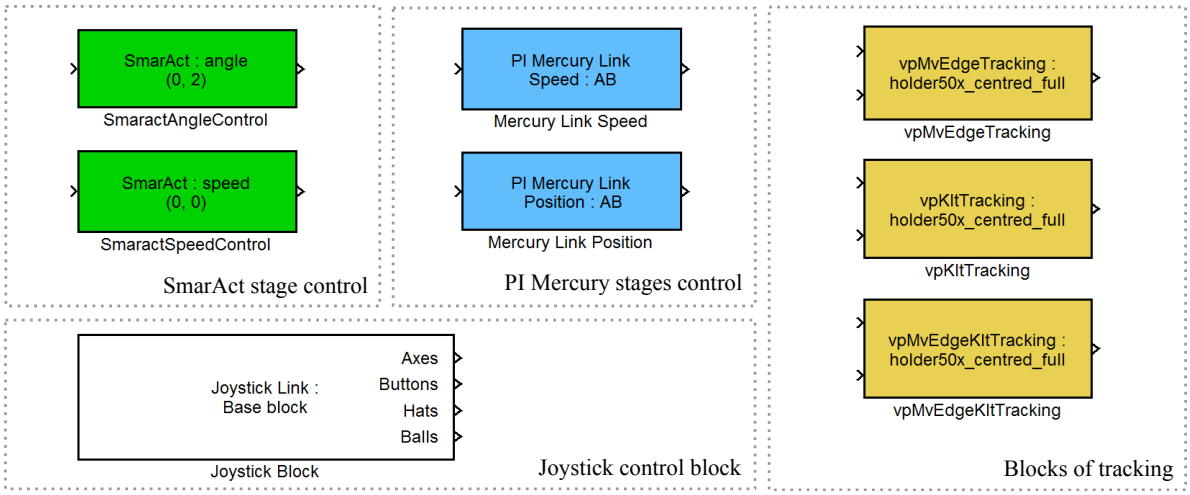


Fig. 5: Overview of the blocks of MATLAB/Simulink library developed to control the micromanipulation station in teleoperated or automated mode

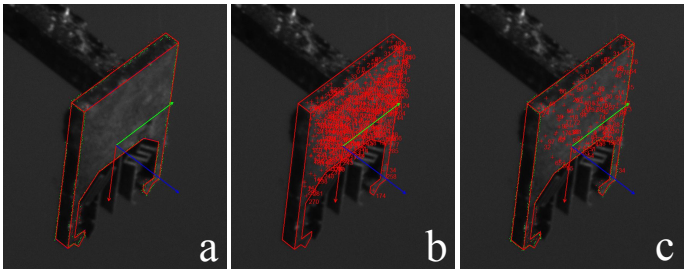


Fig. 6: Different types of tracking : a) edge-based; b) texture-based; c) hybrid.

TABLE III: Standard deviation of position measurement for different trackers in a still frame for 1000 iterations in the camera frame.

Coordinates	Edge-based	Texture-based	Hybrid
X	0.031 µm	0 µm	0.117 µm
Y	0.059 µm	0 µm	0.010 µm
Z	3.204 µm	0 µm	0.760 µm
roll	0.0071°	0°	0.0194°
pitch	0.0068°	0°	0.0152°
yaw	0.0014°	0°	0.0252°

A. *Trackers analysis in a still frame*

The first stage of pose estimation analysis consisted in tracking of the object in a still frame (Fig. 6). The tracking results obtained using the blocks displayed in previous section are represented in the form of standard deviation in Table III. These results demonstrate the performances of the visual trackers in 6-DOF in case all external factors are eliminated, because the video flow is replaced by a single image. It should be noted that even with a still frame, the edge-based tracking is not perfectly stable, neither is hybrid tracking. The noise of X and Y translation coordinates does not exceed 0.117 µm (Table III) which represents an error of less than 30% of the image pixel size. This noise level represents a maximal limit of tracker performance for our application. The predominant noise on the Z coordinate can be explained by the fact that the focal distance is considerably higher than the size of the sensor, and the pinhole projection model becomes close to a parallel one, thus, it is more difficult to estimate the depth coordinate. The problem of depth coordinate estimation with long focal distance has also been established for SEM environments where assumption of parallel beams is close to reality [17] [18].

B. *Trackers analysis with a static object*

In this section, the object pose is recorded while the robot is not moving. This experiment allows to quantify the global influence of the environmental factors: e.g., luminosity and temperature change during the experiment, vibrations of axes of robot and of camera support, human presence in the room, etc. The resulting standard deviation errors (Table IV) were obtained using the video stream with the object fixed in the microgripper and the pose is recorded over 1000 images with 15 frames per second.

TABLE IV: Standard deviation of position measurement for different trackers with a static object in the camera frame.

Coordinates	Edge-based	Texture-based	Hybrid
X	0.2245 µm	0.7861 µm	0.2551 µm
Y	0.7026 µm	0.8650 µm	0.6207 µm
Z	24.3304 µm	44.1736 µm	14.8640 µm
roll	0.0859°	0.1607°	0.0610°
pitch	0.0577°	0.1254°	0.0539°
yaw	0.0608°	0.0984°	0.0461°

The results confirm the observations made before concerning the imprecision of the measurements along the Z_c axis of camera frame. The standard deviation of measured coordinates X and Y has become bigger comparing to the results with a still frame. The maximal error is 865 nm for texture-based tracker and is about 700 nm for both edge-based and hybrid trackers. However, this level of noise is not suprising. The order of magnitude is similar to the results of noise characterization in millimeter sized micromanipulation systems presented by

Boudaoud *et al.* in [19]. Their experiments allowed to conclude that the vibration at the free end of the cantilever of 30 mm subject to the environmental noise on an anti-vibration table and with human activity reaches 123.7 nm. This value was measured using laser interferometer. Thus, since the distance between the microgripper tip and the point of robot mount is about 20 cm in our system, it may explain that the noise level can reach 700 nm.

C. Planned path tracking

Next step of tracking analysis consists in comparing the measurements between proprioceptive robot sensors and the visual sensor. In this section we will use the hybrid tracker to estimate the position of the object, which gave the best results during the previous experiments. The robot control is performed by applying a position set-point to the robotic axis and in order to simplify the analysis, the rotations are not taken into account. The comparison is performed by using the planned path represented in Fig. 7, where the curves are represented in the world frame.

Thereafter, we use the following notations :

- i - image number;
- $\mathcal{R}_c$ - camera frame;
- $\mathcal{R}_w$ - world frame;
- ${}^c\mathbf{P}_o$ - the pose of the object in the camera frame obtained with the tracker:
$$ {}^c\mathbf{P}_o = \begin{pmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ 0 & 1 \end{pmatrix} $$
where ${}^c\mathbf{t}_o = \{{}^c x_o \; {}^c y_o \; {}^c z_o\}^\top$;
- ${}^w\mathbf{P}_o$ - the pose of the object in the world frame (proprioceptive robot sensors), rotations are not taken into account:
$$ {}^w\mathbf{P}_o = \begin{pmatrix} \mathbf{I}_{3\times3} & {}^w\mathbf{t}_o \\ 0 & 1 \end{pmatrix} $$
where ${}^w\mathbf{t}_o = \{{}^w x_o \; {}^w y_o \; {}^w z_o\}^\top$;
- ${}^c_w\mathbf{P}_o$ with ${}^c_w\mathbf{t}_o = \{{}^c_w x_o \; {}^c_w y_o \; {}^c_w z_o\}^\top$ - object pose in the world frame transformed to the camera frame;

In order to be able to compare the results, it is necessary to transform object coordinates in the world frame ${}^w\mathbf{t}_o$ to the camera frame ${}^c_w\mathbf{t}_o$ for each image. This transformation can be represented by a matrix ${}^c\mathbf{M}_w$ which contains the information about frame rotations and translations:

$$ {}^c_w\mathbf{P}_o = {}^c\mathbf{M}_w \, {}^w\mathbf{P}_o \tag{1} $$

where ${}^c\mathbf{M}_w$ is the homogeneous matrix, which represents the extrinsic parameters of the camera, is not known. In order to estimate it, an optimization algorithm was used. The goal of optimization consists in minimization of the distance $\mathbf{\Delta}$ between the pose obtained from the tracker ${}^c\mathbf{P}_o$ and the sensor values of robot axis ${}^c_w\mathbf{P}_o$ in camera frame. So, for each image:

$$ \mathbf{\Delta} = \begin{bmatrix} {}^c\mathbf{t}_o \\ 1 \end{bmatrix} - {}^c\mathbf{M}_w \begin{bmatrix} {}^w\mathbf{t}_o \\ 1 \end{bmatrix} \tag{2} $$

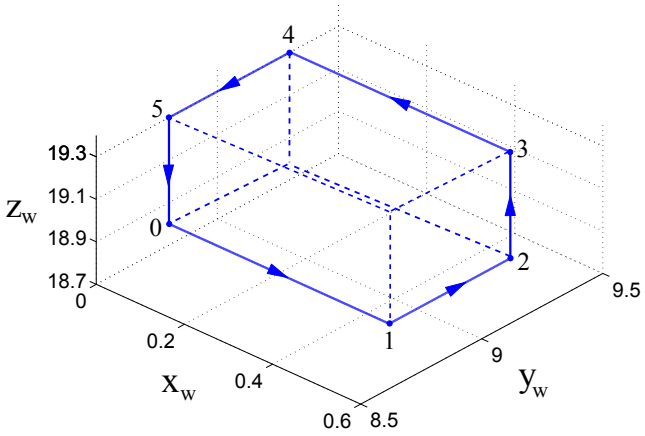


Fig. 7: 3D path in the world frame to be tracked (in mm).

The optimization criteria is then defined as a sum of squared distance between two curves:

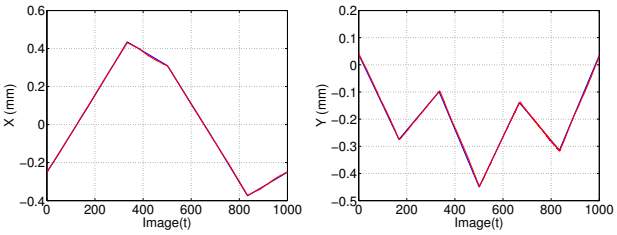
$$ J = \sum_{i=1}^{n} \mathbf{\Delta}_i^T \, \mathbf{A} \, \mathbf{\Delta}_i \tag{3} $$

The previous analysis of obtained coordinates deviation between two curves allows to notice that ${}^c z_o$ coordinate estimated by the tracker is not only affected by noise but contains no real information about the object position, which means that any filter would be useless. So, in order to minimize its influence, the coefficient of 0.001 was applied for the ${}^c z_o$ coordinate.

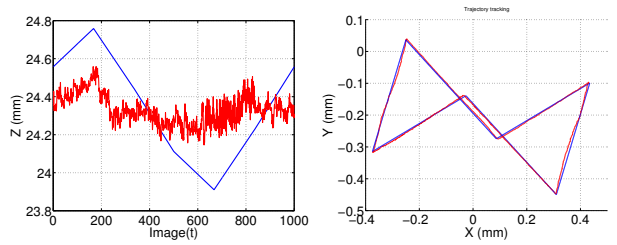
$$ \mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.001 \end{pmatrix} $$

The used optimization algorithm is a Levenberg-Marquardt algorithm [20]. It is implemented in the Mathworks Optimization Toolbox [21] which can be used directly due to the fact that we are working on MATLAB/Simulink environment. It was programmed to take the best fit on 20 optimizations from random initial transforms. The results of comparison between the pose estimation from the visual sensor ${}^c\mathbf{t}_o$ and from robot sensors ${}^c_w\mathbf{t}_o$ transformed to the camera frame are represented in Fig. 8.

The standard deviations of errors between the tracker and the robot joint coordinates presented in Table V attain now 2.8 µm along X and 4 µm along Y axis. These values include noises coming from both algorithm and environmental factors. Their influence was measured in previous experiments: about 120 nm for algorithm noise (Table III) as hybrid tracker was used, and 700 nm for environmental noise (Table IV). Additionally, these deviations include intrinsic robot positioning errors which can typically reach several micrometers of amplitude. For example, M-111-DG translation stages have 150 µrad (Physik Instrumente company datasheet) yaw deviation which in our case (20 cm long robot end-effector) may induce an error of 3 µm. Tan *et al.* have also recently shown that this kind of robot have typical positioning accuracy of a few



(a) ${}^{c}x_o(i)$, ${}^{c}_{w}x_o(i)$ in mm (b) ${}^{c}y_o(i)$, ${}^{c}_{w}y_o(i)$ in mm



(c) ${}^{c}z_o(i)$, ${}^{c}_{w}z_o(i)$ in mm (d) Planned path in a XY plane

Fig. 8: Experimental comparison between the results from the visual sensor ${}^{c}\mathbf{t}_o$ (red) and from robot sensors ${}^{c}_{w}\mathbf{t}_o$ (blue) in $\mathcal{R}_c$ using hybrid tracker;

TABLE V: Standard deviation error obtained between visual hybrid tracker and robot sensors in $\mathcal{R}_c$

Coordinates	Standard deviation error
Δx	2.8165 μm
Δy	4.0918 μm
Δ**z**	**215.8632 μm**

micrometers and that the main sources of positioning errors are backlash, perpendicularity errors and yaw deviations [22].

V. Discussion

The visual tracking methods for object position estimation presented in this paper represent a great interest for multi-DOF position measurements at the microscale. They can be directly applied to many kinds of applications such as in the Micro–Opto–Electro–Mechanical Systems field. For example, they could be used to measure the position of the micromirror presented in [23] where sensor integration is strongly limited by available space constraints. The fact of 3D object model use gives an opportunity to use almost every point of view and not necessarily top or side views. However, even if the trackers performance meets the specifications declared for assembly accuracy, it can not be used directly to achieve a task of 6 DOF microassembly, because of the lack of data accuracy on Z_c.

As it was mentioned before, the non-validity of data obtained from the tracker concerning the Z_c axis of camera can be explained by the fact that we deal with a microobjet, thus the camera magnification is relatively high. The necessity of this high magnification rests upon two factors : firstly, as the form of our object is complex (one dimension is much smaller than two others) the camera must be close enough to be able to detect all of the edges; secondly, the size of visual field should be sufficient to cover all space of object movement necessary

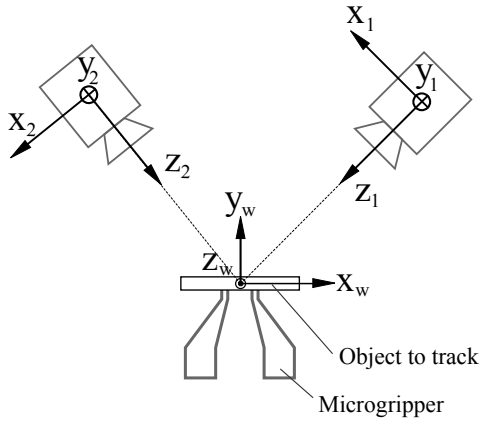


Fig. 9: Cameras position in stereoscopic setup.

to microassembly tasks. So, we encounter the hypothesis also presented in works with Scanning Electron Microscope [24], that at high magnification the projection rays are parallel to each other and are perpendicular to the image plane, which implies the use of parallel projection model.

Nevertheless, several alternative solutions may be used to measure the displacement following Z_c axis. First of all, it is possible to use robot sensors and reconstruct the Z_c coordinate, however, one would have to make a calibration of robotic system to compensate different mechanical defects (hysteresis, creep, dry friction, etc). In this case, the estimation might be limited by stages accuracy which is typically in the micrometer range for such kind of robotic stages [22]. Secondly, the system may be completed by an unidirectional sensor capable to estimate Z_c coordinate such as laser sensors (trinagulation, interferometers, confocal,etc). In the present case it can be placed following Y_w axis (Fig. 3) and Z_c can be reconstructed using a transformation matrix. Finally, a second camera may be used. In order to be able to compensate Z_{c_1} of the first camera, the second one should be placed as it is shown in Fig. 9. However, the stereoscopic vision setup must be calibrated (we have to find the transformation matrix between $\mathcal{R}_1$ and $\mathcal{R}_w$ and the fundamental matrix witch allows to estimate the depth at scale factor. With the association to the CAD model based estimation of Z_c, it is possible to affine the estimation of Z_c) and for three-dimensional reconstruction to be possible, the location and orientation of the cameras at different capture instants must be accurately known [25].

VI. Conclusion

The present work treats the problem of obtaining a high quality position measurement of the micro-object at the microscale for robotic tasks. The chosen approach is a visual tracking based on a CAD-model of microcomponent. The performances of this approach were studied through this paper.

The software for robot control and vision was developed in the form of MATLAB/Simulink blocks. The creation of Simulink blocks in a standard programming language C++ allows to simplify the work with the equipment: program becomes graphic and modular and all existing tools of MATLAB/Simulink can be used.

The analysis of trackers in different situations allowed to estimate the instability of tracking algorithm, environmental factors influence, precision of robot axis. During the experiments presented in Section IV, it was estimated that in present setup, the performance of the algorithm can attain the value of 0.12 µm (Table III) and the noise level due the environment is about 0.7 µm (Table IV). Finally, the robot axis imperfections add about 2 µm of uncertainty. Thus, the sum gives a value of 2.82 µm that corresponds to the results in Table V. The obtained results prove that it is possible to have a precision better than 1 µm for X and Y coordinates using the visual tracking techniques presented in this paper. However, the trackers performance may be improved by using a texture or a periodic structure on the object sides.

Through experimental results, the non-validity of obtained data concerning the Z_c axis of camera was identified. This is due to the fact that the focal distance is considerably higher than the size of a sensor. A discussion section highlights several solutions to enrich the information about object position in 6 DOF.

ACKNOWLEDGMENT

These works have been funded by the Franche-Comté region, partially supported by the Labex ACTION project (contract "ANR-11-LABX-01-01") and by the French RENATECH network through its FEMTO-ST technological facility.

REFERENCES

[1] D. Tolfree and M. J. Jackson, *Commercializing micro-nanotechnology products*. CRC Press, 2010.

[2] W. Noell, W. Sun, N. de Rooij, H. P. Herzig, O. Manzardo, and R. Dandliker, "Optical mems based on silicon-on-insulator (soi) for monolithic microoptics," vol. 2, pp. 580–581, 2002.

[3] F. S. Chau, Y. Du, and G. Zhou, "A micromachined stationary lamellar grating interferometer for fourier transform spectroscopy," *Journal of Micromechanics and Microengineering*, vol. 18, no. 2, p. 025023, 2008.

[4] R. Syms, H. Zou, and J. Stagg, "Micro-opto-electro-mechanical systems alignment stages with vernier latch mechanisms," *Journal of Optics A: Pure and Applied Optics*, vol. 8, no. 7, p. S305, 2006.

[5] N. Dechev, W. L. Cleghorn, and J. K. Mills, "Microassembly of 3-d microstructures using a compliant, passive microgripper," *IEEE Journal of Microelectromechanical Systems*, vol. 13, no. 2, pp. 176–189, 2004.

[6] J. Agnus, N. Chaillet, C. Clévy, S. Dembélé, M. Gauthier, Y. Haddab, G. Laurent, P. Lutz, N. Piat, K. Rabenorosoa *et al.*, "Robotic microassembly and micromanipulation at femto-st," *Journal of Micro-Bio Robotics*, vol. 8, no. 2, pp. 91–106, 2013.

[7] S. Bargiel, K. Rabenorosoa, C. Clevy, C. Gorecki, and P. Lutz, "Towards micro-assembly of hybrid moems components on a reconfigurable silicon free-space micro-optical bench," *Journal of Micromechanics and Microengineering*, vol. 20, no. 4, p. 045012, 2010.

[8] A. N. Das, J. Sin, D. O. Popa, and H. E. Stephanou, "On the precision alignment and hybrid assembly aspects in manufacturing of a microspectrometer," pp. 959–966, 2008.

[9] K. Aljasem, L. Froehly, A. Seifert, and H. Zappe, "Scanning and tunable micro-optics for endoscopic optical coherence tomography," *IEEE Journal of Microelectromechanical Systems*, vol. 20, no. 6, pp. 1462–1472, 2011.

[10] C. Clévy, I. Lungu, K. Rabenorosoa, and P. Lutz, "Positioning accuracy characterization of assembled microscale components for micro-optical benches," *Assembly Automation*, vol. 34, no. 1, pp. 69–77, 2014.

[11] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

[12] B. Tamadazte, E. Marchand, S. Dembélé, and N. Le Fort-Piat, "Cad model-based tracking and 3d visual-based control for mems microassembly," *The International Journal of Robotics Research*, vol. 29, no. 11, pp. 1416–1437, 2010.

[13] R. Pérez, J. Agnus, C. Clévy, A. Hubert, and N. Chaillet, "Modeling, fabrication, and validation of a high-performance 2-dof piezoactuator for micromanipulation," *Mechatronics, IEEE/ASME Transactions on*, vol. 10, no. 2, pp. 161–171, 2005.

[14] D. Xue and Y. Chen, *System Simulation Techniques with MATLAB and Simulink*. John Wiley & Sons, 2013.

[15] P. Corke, "A robotics toolbox for matlab," *IEEE Robotics & Automation Magazine*, vol. 3, no. 1, pp. 24–32, 1996.

[16] M. Pressigout and E. Marchand, "Real-time hybrid tracking using edge and texture information," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 689–713, 2007.

[17] L. Cui and E. Marchand, "Calibration of scanning electron microscope using a multi-images non-linear minimization process," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014.

[18] L. Cui, E. Marchand, S. Haliyo, and S. Régnier, "6-dof automatic micropositioning using photometric information," in *IEEE International Conference on Advanced Intelligent Mechatronics*, Besançon, France, 2014.

[19] M. Boudaoud, Y. Haddab, Y. Le Gorrec, and P. Lutz, "Noise characterization in millimeter sized micromanipulation systems," *Mechatronics*, vol. 21, no. 6, pp. 1087–1097, 2011.

[20] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.

[21] T. Coleman, M. A. Branch, and A. Grace, *Optimization Toolbox for Use with MATLAB: User's Guide, Version 2*. Math Works, Incorporated, 1999.

[22] N. Tan, C. Clévy, G. J. Laurent, P. Sandoz, and N. Chaillet, "Characterization and compensation of xy micropositioning robots using vision and pseudo-periodic encoded patterns," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 2819–2824.

[23] A. Espinosa, X. Zhang, K. Rabenorosoa, C. Clévy, S. R. Samuelson, B. Komati, H. Xie, and P. Lutz, "Piston motion performance analysis of a 3dof electrothermal mems scanner for medical applications," *International Journal of Optomechatronics*, vol. 8, no. 3, 2014.

[24] L. Cui, E. Marchand *et al.*, "Calibration of scanning electron microscope using a multi-images non-linear minimization process," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 5191–5196.

[25] G. Roth, R. Laganiere, and S. Gilbert, "Robust object pose estimation from feature-based stereo," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 4, 2006.