2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Robust Image-based Computation of the 3D Position of RCM Instruments and its Application to Image-guided Manipulation

David Navarro-Alarcon, Zerui Wang, Hiu Man Yip, Yun-hui Liu, Fangxun Zhong, Tianxue Zhang, Jiadong Shi and Hesheng Wang

Abstract— **In this paper, we address the 3D position control of RCM-constrained instruments with monocular cameras. To compute the instrument's position from a single 2D image, we develop an innovative gradient descent algorithm which rotates and translates a line segment (over the plane spanned by the imaged instrument and the optical centre) until it best aligns with the manipulated tool. In contrast with other approaches in the literature, our algorithm only requires to simultaneously observe two feature points; the proposed iterative algorithm is not based on the exact solution, therefore it can still work with noisy image measurements. We derive a kinematic controller that uses the proposed position estimator to guide the 3D motion of a robotic instrument with a monocular camera. We evaluate the performance of our approach with numerical simulations and experiments.**

I. Introduction

In minimally invasive surgery, the surgical instruments are inserted into the patient's body through small entry ports in order to reduce the access trauma. An endoscopic camera must also be introduced into the body to observe these instruments and conduct the operation, e.g. the dissection of tissues/organs. With the now ubiquitous use of robotic technologies in surgery (the best example being the DaVinci Surgical System), researchers are now attempting to automate some surgical operations performed by these (manually) tele-operated systems. A key step to achieve this ambitious goal is to incorporate visual feedback into the robot control algorithms, which can be used e.g. to monitor and control deformations of soft tissues [1], to servo-position a laparoscopic instrument [2], and in general to guide the entire surgical procedure insider the patient's body [3]. Our aim in this paper, is to develop algorithms to compute and control the 3D position of a robotic instrument with real-time image feedback.

Using a stereo endoscope system seems the natural option to visually guide the 3D trajectory of an instrument (see e.g. [4], [5]). Note, however, that monocular endoscope cameras still remain the most commonly used imaging technologies in minimally invasive surgical procedures; algorithms developed for monocular cameras can be easily implemented with conventional laparoscopes. The major challenge to control the instrument's 3D position with 2D images comes from the well-known fact that the projection of a 3D point onto the image plane removes one degree-of-freedom, namely the depth of the point. This lack of depth information makes the observed position of a feature point ambiguous, i.e. it may lay anywhere along the ray spanned by the optical centre and the observed point. The projected images of laparoscopic instruments satisfy certain geometric properties—such as collinearity and remote centre of motion (RCM)—which can be exploited to compute and control the instrument's 3D position from 2D image measurements.

There are some works in the literature that deal with the control and guidance of instruments using visual feedback, see [6] for a comprehensive survey on this topic. The single-camera 3D positioning problem is addressed by Krupa et al. in [7]; this work presents a method to regulate the distance between an instrument that continuously projects a laser pattern and an organ. In [8], Nageotte et al. present a method to compute the pose of a surgical instrument by using a specially designed six-point marker that is rolled up around the tool; this pose estimation technique is applied in [9] to control the trajectory of a surgical robot. A closed-form analytical solution (in the form of a homogeneous linear system) to the 3D pose computation problem can be found in [10]. However, as pointed out by Doignon et al. in [11], this exact solution may be ill-conditioned in the presence of noisy image data; multiple image points (with three as the minimum) must be observed on the instrument in order to robustify this approach. To estimate the tool's pose, Allan et al. proposed in [12] a gradient-descent algorithm that minimises a pixel-constructed energy-like functional.

To contribute to this problem, in this paper we present a new method that computes the 3D position of a robotic instrument from a single captured image. Our on-line algorithm only requires to observe two feature points on the instrument, whereas other methods (e.g. [7]–[9], [11]) require to simultaneously observe at least three points. This algorithm is based on the gradient-descent minimisation of a cost function that quantifies the alignment between an adaptive line segment and the manipulated instrument. For that, we propose a new compact parametrisation of the problem in terms of two variables: a rotation angle and a translation distance of the line segment over the plane spanned by the optical centre and the projected instrument. This iterative algorithm can work even with noisy image measurements since it is designed to minimise a distance error and not to find the exact analytical solution. We propose a motion controller to

DNA, ZW, HMY, YHL, FZ, TZ and JS are with the Department of Mechanical and Automation Engg., The Chinese University of Hong Kong, HKSAR. Corresponding author e-mail: david@cuhk.edu.hk.

H. Wang is with Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China and Key Laboratory of System Control and Information Processing, Ministry of Education of China.

This work is supported in part by the HK RGC (grant numbers 415011, CUHK6/CRF/13G) and the HK ITF (grant number ITS/112/15FP).

978-1-4673-8025-6/16/$31.00 ©2016 IEEE

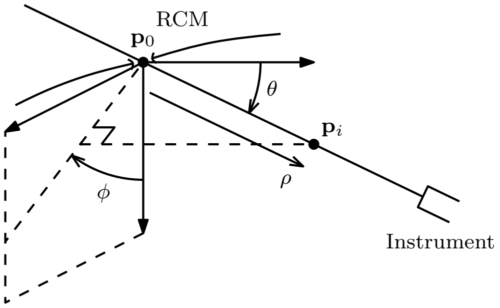

Fig. 1. Conceptual representation of the spherical coordinates used to represent the 3-DOF configuration of the instrument.

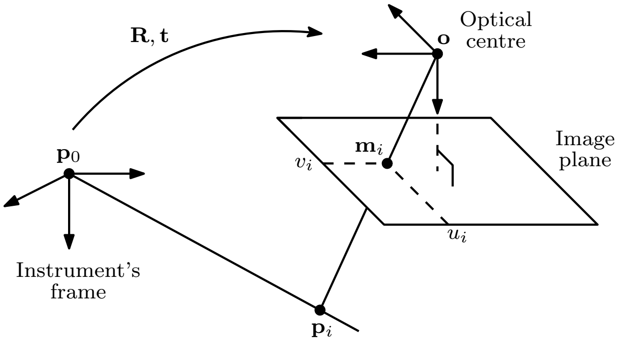

Fig. 2. Perspective projection of a point $\mathbf{p}_i$ onto the camera's image plane.

guide the 3D trajectory of a manipulator with this single-view approach. An experimental study is presented to validate the performance of our method.

The rest of the paper is organised as follows: in Section II we derive the models; Section III presents the image-based algorithm; in Section IV we propose the controller; Section V presents the experiments; Section VI gives final conclusions.

II. Modelling

A. Kinematics of Laparoscopic Instruments

In minimally invasive surgery, the surgical instrument is inserted into the patient's body through a small entry port. This situation imposes certain kinematic constraints to the displacements of the instrument, i.e. it can only performs two rotations around the fixed entry point and one linear translation along the tool's longitudinal axis. Note that in our model, we do not consider the rotation around the insertion axis, since this angle does not contribute to the 3D position.

We parametrise the instrument's DOF with standard spherical coordinates. To this end, let the $\theta, \phi \in \mathbb{R}$ denote the system's orientation angles, and $\rho \in \mathbb{R}$ denote the insertion distance of the tool. We compute the 3D position of a feature point $\mathbf{p}_i \in \mathbb{R}^3$ with respect to the instrument's remote centre of motion, here denoted by $\mathbf{p}_0 \in \mathbb{R}^3$, as follows:

$$\mathbf{p}_i = \rho \begin{bmatrix} \sin(\theta)\cos(\phi) & \sin(\theta)\sin(\phi) & \cos(\theta) \end{bmatrix}^{\mathsf{T}} \quad (1)$$

See Fig. 1 for a conceptual representation of the system.

B. Perspective Projection of a Point

Consider a monocular static camera that observes the manipulated instrument. We model the perspective projection of $\mathbf{p}_i$ onto the image plane (located at unit distance from the optical centre $\mathbf{o}$) with the following expression:

$$z_i \begin{bmatrix} \mathbf{m}_i^{\mathsf{T}} & 1 \end{bmatrix}^{\mathsf{T}} = \mathbf{C} \begin{bmatrix} \mathbf{p}_i^{\mathsf{T}} & 1 \end{bmatrix}^{\mathsf{T}} \quad (2)$$

where $\mathbf{m}_i \in \mathbb{R}^2$ represents the point's image measurements; the scalar $z_i \in \mathbb{R}$ denotes the positive depth (measured along the axis normal to the image plane) of the 3D point; the constant matrix $\mathbf{C} \in \mathbb{R}^{3\times 4}$ represents the calibration matrix of the camera, which has the following form:

$$\mathbf{C} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \quad (3)$$

for $\mathbf{K} \in \mathbb{R}^{3\times 3}$ as the upper-triangular matrix of intrinsic parameters, and $\mathbf{R} \in \mathbb{R}^{3\times 3}$ and $\mathbf{t} \in \mathbb{R}^3$ as the rotation matrix and translation vector, respectively, of the instrument's coordinate frame with respect to the camera's frame. Fig. 2 depicts a conceptual representation of this model.

C. Geometry of the Imaged Instrument

Consider that the camera observes two known feature points on the instrument[1]. We denote the distance from the remote centre of motion to the point $\mathbf{p}_1$ by $d_1 \in \mathbb{R}$, and by $d_2 \in \mathbb{R}$ the distance between the two feature points

$$d_1 = \|\mathbf{p}_1 - \mathbf{p}_0\|, \quad d_2 = \|\mathbf{p}_2 - \mathbf{p}_1\| \quad (4)$$

In our formulation, we assume that both scalar distances d_1 and d_2 are exactly known. To obtain d_1, requires to calibrate the instrument's insertion such that $\rho = d_1$. Note that this positive distance simply represents part of the forward kinematics of the manipulator, and can be calibrated at the beginning of the task. The distance d_2 can be obtained, e.g. by using a specially designed marker on the instrument.

The instrument back projects as an image line that passes over $\mathbf{m}_1$ and $\mathbf{m}_2$. This image line along with the optical centre $\mathbf{o}$ define a plane that contains the feature points $\mathbf{p}_i$. Fig. 3 depicts a conceptual representation of this setup. In our model, the centre of motion is not assumed to be observable by camera (the constant point $\mathbf{m}_0$ can be computed by intersecting image lines, as we show in later sections), however, it must be located in front of the image plane, i.e. with a depth distance greater than one ($z_0 > 1$).

Problem statement. Given the image points $\mathbf{m}_i$ (for $i = 0, 1, 2$), the matrix $\mathbf{K}$, and the scalar distances d_i, develop a robust on-line algorithm to visually measure the 3D position of the instrument from a single captured image.

III. Image-based Computation of 3D Position

A. Rectifying Image Data

In our method, the instrument's centre of rotation does not need to be observed by the camera. We can obtain the projection of the RCM point by computing the intersection

[1] These points may not necessary come from real point features; they may represent e.g. the centroid of a transversal strip on the instrument.

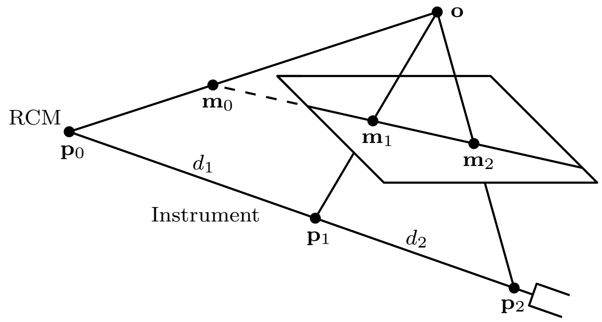

Fig. 3. Conceptual representation of the imaged instrument with two features points $\mathbf{p}_1$ and $\mathbf{p}_2$.

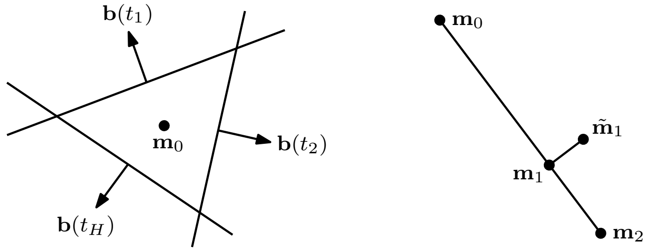

Fig. 4. (left) Computation of $\mathbf{m}_0$ from multiple image lines. (right) Projection of the non-collinear point $\tilde{\mathbf{m}}_1$ onto the line joining $\mathbf{m}_0$, $\mathbf{m}_2$.

between image lines. In principle, all projected lines should intersect at the same image point, however, due to measurement noises and mechanical misalignments, the intersection of these lines may not be not uniquely defined.

To cope with this issue, we compute a point $\mathbf{m}_0$ that minimises (in the least-squares sense) the distance to L image lines. Let $\mathbf{b}(t_k) \in \mathbb{R}^2$ denote a unit vector that is orthogonal to the image line that passes through the points $\mathbf{m}_i(t_k)$, at the time instant t_k. Given $L > 2$ image lines, we compute the point $\mathbf{m}_0$ with the following expression:

$$\mathbf{m}_0 = \left(\sum_{k=1}^{L} \mathbf{b}(t_k)\mathbf{b}^{\intercal}(t_k) \right)^{-1} \sum_{k=1}^{L} \mathbf{b}(t_k)\mathbf{b}^{\intercal}(t_k)\mathbf{m}_2(t_k) \tag{5}$$

Note also that for noisy image measurements, the image points $\mathbf{m}_i$ may not necessary be collinear. In our method, we project the coordinates of the $i = 1$ measurement point, which we denote by $\tilde{\mathbf{m}}_1 \in \mathbb{R}^2$, onto the line segment that joins $\mathbf{m}_0$ and $\mathbf{m}_2$ as follows:

$$\mathbf{m}_1 = \frac{(\tilde{\mathbf{m}}_1 - \mathbf{m}_0) \cdot (\mathbf{m}_2 - \mathbf{m}_0)}{(\mathbf{m}_2 - \mathbf{m}_0) \cdot (\mathbf{m}_2 - \mathbf{m}_0)} (\mathbf{m}_2 - \mathbf{m}_0) + \mathbf{m}_0 \tag{6}$$

See Fig. 4 for conceptual representation of these points.

B. Constructing the 2D Model from Sensor Feedback

By using the matrix $\mathbf{K}$ of intrinsic camera parameters, we can compute the unit 3D directional vector $\mathbf{u}_i \in \mathbb{R}^3$ that spans the ray that passes through the points $\mathbf{m}_i$ and $\mathbf{p}_i$. This vector is computed from sensor feedback as follows [13]:

$$\mathbf{u}_i = \left(\mathbf{K}^{-1} \begin{bmatrix} \mathbf{m}_i^{\intercal} & 1 \end{bmatrix}^{\intercal} \right) / \left\| \mathbf{K}^{-1} \begin{bmatrix} \mathbf{m}_i^{\intercal} & 1 \end{bmatrix}^{\intercal} \right\| \tag{7}$$

The directional vectors $\mathbf{u}_0$, $\mathbf{u}_1$ and $\mathbf{u}_2$ along with the optical centre $\mathbf{o}$ form a plane that contains the instrument's centre of rotation $\mathbf{p}_0$ and the two feature points $\mathbf{p}_1$ and $\mathbf{p}_2$ (see Fig. 5). This property suggests that the problem of computing the 3D position of the instrument can be re-formulated as a 2D problem. To see this, let us first compute the 3-vector $\mathbf{n} = (\mathbf{u}_0 \times \mathbf{u}_2) / \|\mathbf{u}_0 \times \mathbf{u}_2\|$ that is orthogonal to the plane spanned by $\mathbf{u}_0$ and $\mathbf{u}_2$. Next, let us compute the following rotation matrix:

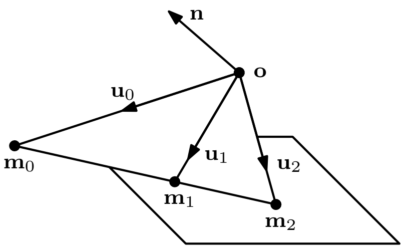

Fig. 5. The plane computed with the vectors $\mathbf{u}_i$ and image points $\mathbf{m}_i$.

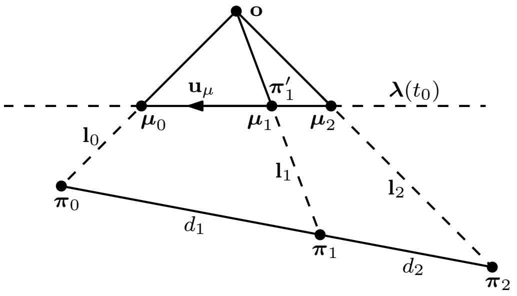

Fig. 6. Computation of the tool's position represented as a 2D problem.

$$\mathbf{H} = \begin{bmatrix} \mathbf{u}_0 & \frac{\mathbf{n} \times \mathbf{u}_0}{\|\mathbf{n} \times \mathbf{u}_0\|} & \mathbf{n} \end{bmatrix}^{\intercal} \in \mathbb{R}^{3 \times 3} \tag{8}$$

which transforms the image points and instrument's points into a more convenient representation with null terms in the "z" coordinate (i.e. the distance along the normal vector $\mathbf{n}$).

Consider the following coordinate transformation:

$$\begin{bmatrix} \boldsymbol{\mu}_i \\ 0 \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{\pi}_i \\ 0 \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \tag{9}$$

where the 2-vectors $\boldsymbol{\mu}_i \in \mathbb{R}^2$ and $\boldsymbol{\pi}_i \in \mathbb{R}^2$ represent the new coordinates of the image points and instrument's points over the plane, respectively. Using these transformed coordinates, we can compute the geometric line $\mathbf{l}_i \in \mathbb{R}^3$ that passes through the points $\boldsymbol{\mu}_i$ and $\boldsymbol{\pi}_i$, and which satisfies (see [13])

$$\begin{bmatrix} \boldsymbol{\mu}_i^{\intercal} & 1 \end{bmatrix} \cdot \mathbf{l}_i = \begin{bmatrix} \boldsymbol{\pi}_i^{\intercal} & 1 \end{bmatrix} \cdot \mathbf{l}_i = 0 \tag{10}$$

To complete our model, we compute the unit directional vector $\mathbf{u}_\mu \in \mathbb{R}^2$ that points from $\boldsymbol{\mu}_1$ to $\boldsymbol{\mu}_0$

$$\mathbf{u}_\mu = (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) / \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\| \tag{11}$$

Fig. 6 shows a representation of the computed 2D model.

C. Parametrising the Line

Let us define the coordinates of a geometric line $\boldsymbol{\lambda} \in \mathbb{R}^3$ (note that this definition also applies to $\mathbf{l}_i$) in terms of arbitrary 2D points $\mathbf{s}_j = [s_{jx}, s_{jy}]^{\intercal}$ as follows:

$$\boldsymbol{\lambda}(\mathbf{s}_1, \mathbf{s}_2) = \begin{bmatrix} s_{2y} - s_{1y} & s_{1x} - s_{2x} & s_{2x}s_{1y} - s_{1x}s_{2y} \end{bmatrix}^{\intercal} \tag{12}$$

Now, consider a point $\boldsymbol{\pi}'_1 \in \mathbb{R}^2$ which lays along $\boldsymbol{\lambda}$, and which represents an estimation of the instrument's feature point $\boldsymbol{\pi}_1$ over the plane; we compute $\boldsymbol{\pi}'_1$ with the expression

$$\boldsymbol{\pi}'_1 = q_1 \boldsymbol{\mu}_1 / \|\boldsymbol{\mu}_1\| \tag{13}$$

where $q_1 \in \mathbb{R}$ represents a scalar displacement that is measured along $\mathbf{l}_1$. To define the orientation of $\boldsymbol{\lambda}$, we compute a second point $\mathbf{r} \in \mathbb{R}^2$ with the expression

$$\mathbf{r} = \boldsymbol{\pi}'_1(q_1) + \begin{bmatrix} \cos(q_2) & -\sin(q_2) \\ \sin(q_2) & \cos(q_2) \end{bmatrix} \mathbf{u}_\mu \tag{14}$$

where $q_2 \in \mathbb{R}$ represents a rotation angle that is measured around $\boldsymbol{\pi}'_1$. With these two points, the configuration of the line can be completely determined as $\boldsymbol{\lambda}(\boldsymbol{\pi}'_1, \mathbf{r})$.

In our method, we initialise at the time instant $t = t_0$ the displacement and rotation variables with $q_1(t_0) = \|\boldsymbol{\mu}_1\|$ and $q_2(t_0) = 0$, which renders $\boldsymbol{\lambda}$ as the line that passes through the points $\boldsymbol{\mu}_i$, as depicted in Fig. 6. To simplify notation, the variables q_1 and q_2 are grouped into the following 2-vector:

$$\mathbf{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^\intercal \tag{15}$$

D. The On-line Iterative Algorithm

The core idea behind our algorithm is to translate and rotate $\boldsymbol{\lambda}$ (namely to vary q_1 and q_2) until the distance from $\boldsymbol{\pi}'_1$ to the intersection with the lines $\mathbf{l}_0$ and $\mathbf{l}_2$ has length d_1 and d_2, respectively. We compute the intersection point $\widehat{\mathbf{w}}_i = [\widehat{w}_{ix}, \widehat{w}_{iy}, \widehat{w}_{iz}]^\intercal$ (denoted in homogeneous coordinates) between the adaptive line $\boldsymbol{\lambda}$ and the constant line $\mathbf{l}_i$ by using the following expression [13]:

$$\widehat{\mathbf{w}}_i = \boldsymbol{\lambda} \times \mathbf{l}_i \tag{16}$$

The inhomogeneous representation $\mathbf{w}_i \in \mathbb{R}^2$ is obtained as

$$\mathbf{w}_i = \begin{bmatrix} \widehat{w}_{ix}/\widehat{w}_{iz} & \widehat{w}_{iy}/\widehat{w}_{iz} \end{bmatrix}^\intercal \tag{17}$$

To design the algorithm, we iteratively compute the scalar energy-like function

$$V(\mathbf{q}) = \frac{1}{2}\left(\|\mathbf{w}_0 - \boldsymbol{\pi}'_1\| - d_1\right)^2 + \frac{1}{2}\left(\|\mathbf{w}_2 - \boldsymbol{\pi}'_1\| - d_2\right)^2 \tag{18}$$

which quantifies the alignment between the line segment joining $\mathbf{w}_0$ with $\mathbf{w}_2$, and the manipulated instrument.

Proposition 1. *The gradient descent update rule*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q} = -\boldsymbol{\Gamma}\frac{\partial V}{\partial \mathbf{q}}^\intercal \tag{19}$$

for $\boldsymbol{\Gamma} \in \mathbb{R}^{2\times 2}$ as a positive diagonal gain matrix, asymptotically minimises the cost function V.

Proof. Consider $V(\mathbf{q})$ as a Lyapunov-like function for (19); by computing its time-derivative we obtain

$$\dot{V} = -\frac{\partial V}{\partial \mathbf{q}}\boldsymbol{\Gamma}\frac{\partial V}{\partial \mathbf{q}}^\intercal \tag{20}$$

which shows that V monotonically decreases along (19) [14]. ■

E. Computing the Instrument's Pose

Once $V(\mathbf{q})$ has been minimised (up-to an arbitrary value $\epsilon > 0$), we use the vector of variables $\mathbf{q}$ to compute the other two instrument's points over the plane

$$\boldsymbol{\pi}'_0 = \boldsymbol{\pi}'_1 + d_1\left(\mathbf{r} - \boldsymbol{\pi}'_1\right), \quad \boldsymbol{\pi}'_2 = \boldsymbol{\pi}'_1 - d_2\left(\mathbf{r} - \boldsymbol{\pi}'_1\right) \tag{21}$$

and then transform them as follows

$$^{c}\mathbf{p}_i = \mathbf{H}^\intercal \begin{bmatrix} \boldsymbol{\pi}'^{\intercal}_i & 0 \end{bmatrix}^\intercal \tag{22}$$

where $^{c}\mathbf{p}_i \in \mathbb{R}^3$ denotes the coordinates of $\mathbf{p}_i$ represented in the camera's frame. To compute $\mathbf{R}$, we first define at $t = t_j$, the directional vectors $\vec{\boldsymbol{u}}_j, {}^{c}\vec{\boldsymbol{u}}_j \in \mathbb{R}^3$ that point along the instrument's axis

$$\vec{\boldsymbol{u}}_j = \frac{\mathbf{p}_2(t_j)}{\|\mathbf{p}_2(t_j)\|}, \quad {}^{c}\vec{\boldsymbol{u}}_j = \frac{^{c}\mathbf{p}_2(t_j) - {}^{c}\mathbf{p}_0(t_j)}{\|^{c}\mathbf{p}_2(t_j) - {}^{c}\mathbf{p}_0(t_j)\|} \tag{23}$$

By collecting two linearly independent sets of $\vec{\boldsymbol{u}}_j$, $^{c}\vec{\boldsymbol{u}}_j$, we can compute the unit vectors $\vec{\boldsymbol{v}}$, $^{c}\vec{\boldsymbol{v}} \in \mathbb{R}^3$ that are perpendicular to these temporal measurements

$$\vec{\boldsymbol{v}} = \frac{\vec{\boldsymbol{u}}_1 \times \vec{\boldsymbol{u}}_2}{\|\vec{\boldsymbol{u}}_1 \times \vec{\boldsymbol{u}}_2\|}, \quad {}^{c}\vec{\boldsymbol{v}} = \frac{^{c}\vec{\boldsymbol{u}}_1 \times {}^{c}\vec{\boldsymbol{u}}_2}{\|^{c}\vec{\boldsymbol{u}}_1 \times {}^{c}\vec{\boldsymbol{u}}_2\|} \tag{24}$$

The rotation matrix $\mathbf{R}$ is computed from two position measurements by using the Triad Method as follows [15]:

$$\mathbf{R} = \begin{bmatrix} ^{c}\vec{\boldsymbol{u}}_1 & ^{c}\vec{\boldsymbol{v}} & ^{c}\vec{\boldsymbol{u}}_1 \times {}^{c}\vec{\boldsymbol{v}} \end{bmatrix} \begin{bmatrix} \vec{\boldsymbol{u}}_1 & \vec{\boldsymbol{v}} & \vec{\boldsymbol{u}}_1 \times \vec{\boldsymbol{v}} \end{bmatrix}^\intercal \tag{25}$$

The translation vector $\mathbf{t}$ is simply computed as

$$\mathbf{t} = \mathbf{R}^\intercal\, {}^{c}\mathbf{p}_0 \tag{26}$$

The overall implementation algorithm is given as follows:

```
for Every iteration do
    if Algorithm is uninitialised then
        Initialise variables $q_1 = \|\boldsymbol{\mu}_1\|$ and $q_2 = 0$
    end if
    Measure image line and points
    Rectify points $\mathbf{m}_0 \leftarrow$ (5) and $\mathbf{m}_1 \leftarrow$ (6)
    Compute rotation matrix $\mathbf{H} \leftarrow$ (8)
    Compute 2D points $\boldsymbol{\mu}_i \leftarrow$ (9)
    while Cost function $V(\mathbf{q}) \geq \epsilon$ do
        Compute parametric line $\boldsymbol{\lambda}(\boldsymbol{\pi}'_1, \mathbf{r}) \leftarrow$ (12)
        Compute intersection point $\mathbf{w}_i \leftarrow$ (17)
        Update vector of variables $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q} \leftarrow$ (19)
    end while
    Compute instrument's 3D position $^{c}\mathbf{p}_i \leftarrow$ (22)
    if System is not calibrated then
        Compute matrix $\mathbf{R} \leftarrow$ (25) and vector $\mathbf{t} \leftarrow$ (26)
    end if
end for
```

Remark 1. The condition of having a pre-calibrated insertion distance $\rho = d_1$ can be relaxed by observing an additional feature point, here denoted by $\overline{\mathbf{p}}_0$ and its image measurement by $\overline{\mathbf{m}}_0$, located between the RCM point $\mathbf{p}_0$ and $\mathbf{p}_1$. With this three-point approach, the known distance d_2 along with $\bar{d}_1 = \|\mathbf{p}_1 - \overline{\mathbf{p}}_0\|$ impose the necessary constraints to solve the 2D geometry problem that is represented in Fig. 6. By substituting $d_1 = \bar{d}_1$ and $\mathbf{m}_0 = \overline{\mathbf{m}}_0$, the implementation of the three-point approach is analogous to the one described in this section.

Remark 2. There exists two numerical solutions to the geometric problem depicted in Fig. 6. The first one represents the true position vector ${}^c\mathbf{p}_i$, the second one represents the mirrored position of the instrument with respect to $\mathbf{o}$. Identifying the mirrored solution can be done by evaluating the depth component z_i (which should be positive) of ${}^c\mathbf{p}_i$. This vector is corrected by simply multiplying it by -1.

IV. Image-guided Manipulation

We can use the previous image-based algorithm to guide the trajectory of a robotic instrument. For that, we introduce the vector ${}^c\mathbf{p}_d \in \mathbb{R}^3$ which represents the 3D target position of the instrument with respect to the camera frame, and define the configuration vector of the system as $\mathbf{x} = [\theta, \phi, \rho]^\intercal$. By computing the time-derivative of ${}^c\mathbf{p}_2$, we obtain the following first-order differential expression:

$$ {}^c\dot{\mathbf{p}}_2 = \mathbf{J}_p \dot{\mathbf{x}} \tag{27} $$

where $\mathbf{J}_p(\mathbf{x}) \in \mathbb{R}^{3\times 3}$ denotes the position Jacobian matrix, which we compute as

$$ \mathbf{J}_p = \mathbf{R}\frac{\partial \mathbf{p}_2}{\partial \mathbf{x}} \tag{28} $$

To regulate the position of the instrument, we use the following kinematic control law [16]:

$$ \dot{\mathbf{x}} = -k\mathbf{J}_p^{-1}\,\mathrm{sat}({}^c\Delta\mathbf{p}) \tag{29} $$

where ${}^c\Delta\mathbf{p} = {}^c\mathbf{p}_2 - {}^c\mathbf{p}_d \in \mathbb{R}^3$ represents the 3D position error measured with the monocular camera, $\mathrm{sat}() : \mathbb{R}^3 \mapsto \mathbb{R}^3$ denotes a standard vectorial saturation function (see e.g. [17]), and k represents a positive feedback gain. In the literature, this kind of control scheme is typically referred to as position-based visual servoing [18].

V. Experiments

In this section, we evaluate the proposed methods with numerical simulations and experiments.

A. Numerical Simulation

We first test the performance of the algorithm with a 2D simulation experiment. For this numerical study, a tool with scalar distances $d_1 = 2$ and $d_2 = 3$, and with points ${}^c\mathbf{p}_0 = [4, 3, 4]^\intercal$ and ${}^c\mathbf{p}_2 = {}^c\mathbf{p}_0 + (d_1 + d_2)\frac{1}{3}[-1, -2, 2]^\intercal$ is considered. The algorithm is implemented in C++ language and graphically visualised using the OpenCV libraries. To make the numerical study more realistic, noise is simulated in the virtual image points $\mathbf{m}_i$, and also uncertainty is considered in the tool distances d_i (for both cases, noises of $\pm 1/50$ of the assumed measurement are used). Fig. 7 illustrates the evolution of the adaptive line $\boldsymbol{\lambda}$ for different iterations t_j. From this figure we can identify the different steps of the algorithm: first, it aligns $\boldsymbol{\lambda}$ with the image line (step t_0); next, it varies the vector $\mathbf{q}$ in the direction that minimises the scalar function V (steps t_1–t_4); finally, at $V < \epsilon$, it computes the position of the instrument's points over the 2D plane (step t_5). The accompanying video demonstrates the performance of the proposed adaptive algorithm.

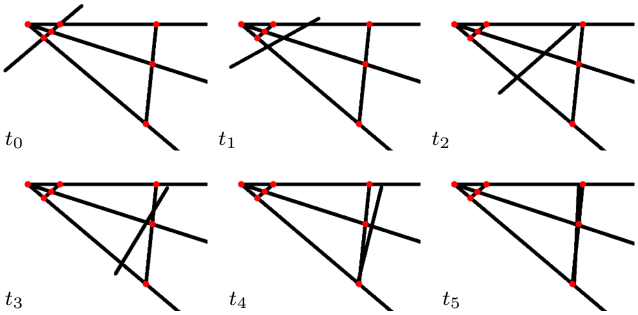

Fig. 7. Numerical simulation of the image-based algorithm.

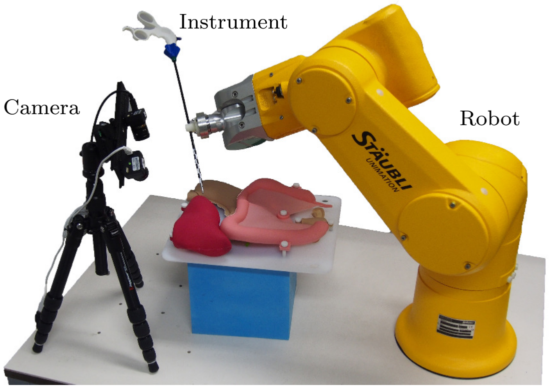

Fig. 8. The experimental robotic set-up.

We emphasise that in the presence of large measurement noise and mechanical misalignments or uncertainties, the energy-like functional may not be minimised to zero, but to a small value ϵ. In this situation, the proposed algorithm computes an approximation of the 3D position.

B. Image-based Computation of 3D Position

A 6-DOF Staubli robot with a surgical instrument attached to the end-effector is used as the experimental platform. To observe collinear features points along the instrument (but not necessary centred on the axis), a simple chessboard-like pattern is rolled-up around the tool. The image coordinates are measured with a Point Grey camera, at a rate of 30fps. The motion of the robot is programmed such that the manipulated instrument satisfies the RCM constraints. Fig. 8 depicts the experimental system.

We evaluate the accuracy of the proposed method by comparing the 3D position obtained with the image-based algorithm, here denoted by ${}^c\mathbf{p}_{a2}$, with the position measured with stereo vision ${}^c\mathbf{p}_{s2}$. In these comparison experiments, the instrument is moved along an arbitrary 3D trajectory. The algorithm is tested by using the feedback from two image points (as depicted in Fig. 3), and by using the feedback from three points (as described in Remark 1); Fig. 9 shows the captured images during these experiments. The arbitrary

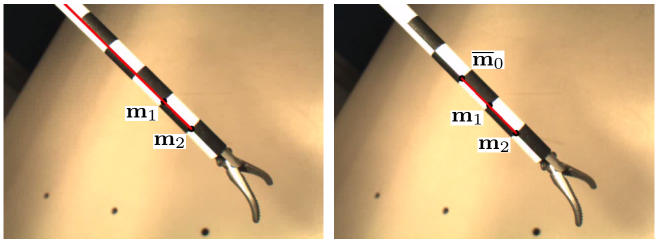

Fig. 9. Captured images of experiments with (left) the two-point approach, and (right) the three-point approach.

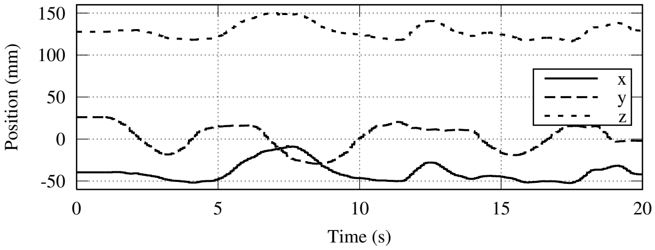

Fig. 10. Trajectory of the point ${}^c\mathbf{p}_2 = [\mathrm{x}, \mathrm{y}, \mathrm{z}]^\mathsf{T}$ measured with the stereo vision system, for the comparison experiment with the two-point algorithm.

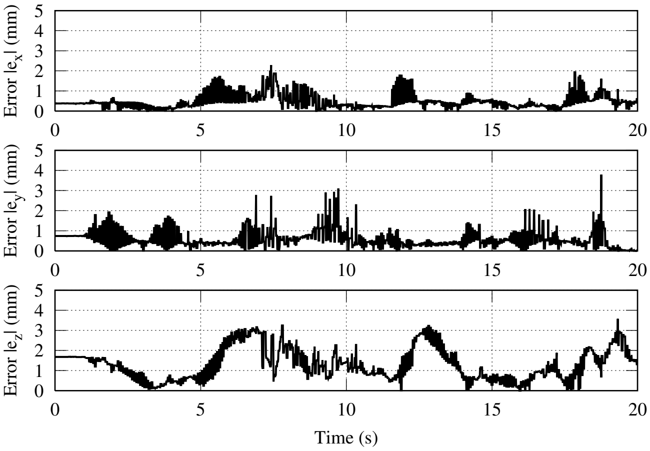

Fig. 11. Magnitude of the position error coordinates $\mathbf{e} = [\mathrm{e_x}, \mathrm{e_y}, \mathrm{e_z}]^\mathsf{T}$ obtained with the two-point algorithm, for the trajectory shown in Fig. 10.

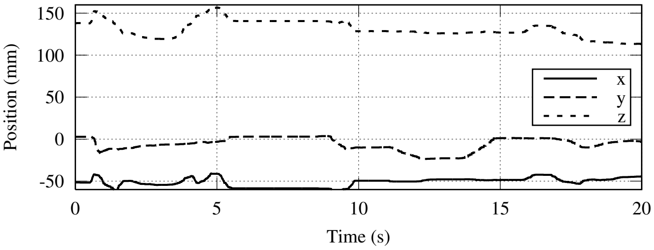

Fig. 12. Trajectory of the point ${}^c\mathbf{p}_2 = [\mathrm{x}, \mathrm{y}, \mathrm{z}]^\mathsf{T}$ measured with the stereo vision system, for the comparison experiment with the three-point algorithm.

position trajectory used for the two-point experiment is shown in Fig. 10, and the coordinates of the computed comparison error $\mathbf{e} = {}^c\mathbf{p}_{s2} - {}^c\mathbf{p}_{a2}$ are shown in Fig. 11. For the three-point approach (i.e. the experiment depicted on the right-hand side of Fig. 9), Fig. 12 shows the performed trajectory, and Fig. 13 shows the comparison error $\mathbf{e}$. From these figures, we can see that the proposed algorithm can compute the 3D position from a single image with an error of usually less than 5mm for each coordinate (note that the three-point approach shows larger errors in the z coordinate).

C. Automatic 3D Positioning

The image-guided positioning controller (described in Section IV) is tested in a simulated surgical scenario. By using the algorithm proposed in Section III, the 3D position of the instrument can be guided with a single endoscopic camera. The motion controller is implemented with a feedback gain $k = 0.005$, and with saturation bounds of 5 mm for all error coordinates. Fig. 14 shows the initial and final configurations for a positioning experiment with an starting error of ${}^c\Delta\mathbf{p}(0) = [-20, 10, 15]^\mathsf{T}$; this experiment only considers the feedback of two feature points. The 3D error trajectory of the instrument can be seen in Fig. 15; the position coordinates of the instrument are computed at every iteration with a single captured image. The asymptotic minimisation of the positioning error is shown in Fig. 16. From this error profile, we can see the 'straightening' effect (around the [5,10] sec period) of the saturation function.

VI. Conclusions

In this paper, we presented an image-based method to compute and control the 3D position of a robotic instrument that satisfies RCM kinematic constraints. We developed an adaptive gradient descent algorithm that iteratively computes the instrument's 3D position from a single captured image. Next, we derived a simple controller to visually guide the 3D trajectory of the instrument with a monocular camera. Finally, we presented an experimental study to validate the proposed methods.

The main idea behind the presented position estimation algorithm, is to rotate and translate a line segment over the plane spanned by the observed instrument and the optical centre. Note that the proposed algorithm does not represent a closed-form exact solution to the problem. Instead, this method iteratively minimises a cost function that quantifies the alignment between the instrument and the adaptive line segment. Hence, the method can be used to estimate (or more specifically, to approximate) the 3D position even when the image measurements present noises and when mechanical misalignments exists. We emphasise the proposed 2-point algorithm may present deviations in the computed position when large errors exist in the estimated RCM point. In general, this monocular method should be used in combination with other approaches, or as the input to other refining algorithms which can compute a more precise position vector.

As future work, we would like to use different types of markers (perhaps based colour detection and not corners) to compute the ideally centred feature points on the instrument. Our group is currently working on the development of a special "rotation-distinguishing marker" (with observable 3

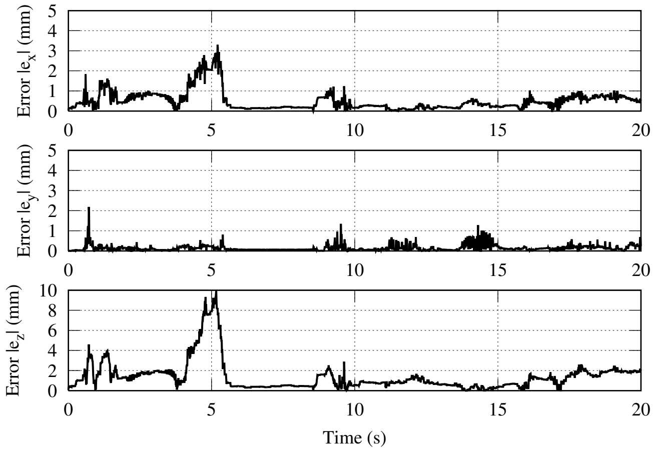

Fig. 13. Magnitude of the position error coordinates $\mathbf{e} = [e_x, e_y, e_z]^\intercal$ obtained with the three-point algorithm, for the trajectory shown in Fig. 12.

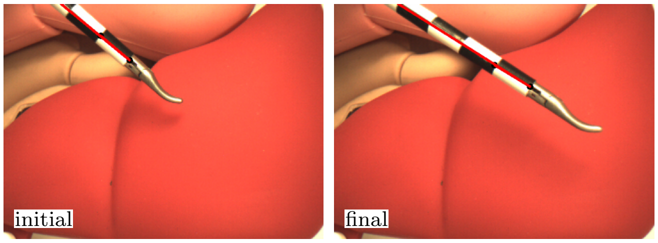

Fig. 14. Captured initial and final images of the positioning experiment.

points) to additionally compute the roll angle of the instrument. Finally, we would like to conduct a detailed experimental ex-vivo study comparing the numerical performance of the different methods to compute 3D position from a single image.

References

[1] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *Int. J. Robot. Res.*, vol. 33, no. 11, pp. 1462–1480, Sep. 2014.

[2] D.-L. Chow, R. Jackson, M. Cavusoglu, and W. Newman, "A novel vision guided knot-tying method for autonomous robotic surgery," in *Proc. Int. Conf. Automation Science and Engg.*, 2014, pp. 504–508.

[3] P. Kazanzides, G. Fichtinger, G. Hager, A. Okamura, L. Whitcomb, and R. Taylor, "Surgical and interventional robotics - core concepts, technology, and design [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 15, no. 2, pp. 122–130, June 2008.

[4] P. Hynes, G. Dodds, and A. Wilkinson, "Uncalibrated visual-servoing of a dual-arm robot for mis suturing," in *Proc. IEEE Int. Conf. Biomedical Robotics and Biomechatronics*, 2006, pp. 420–425.

[5] T. Osa, C. Staub, and A. Knoll, "Framework of automatic robot surgery system using visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 1837–1842.

[6] M. Azizian, M. Khoshnam, N. Najmaei, and R. V. Patel, "Visual servoing in medical robotics: a survey. Part I: endoscopic and direct vision imaging — techniques and applications," *Int. J. Med. Robot. Comp.*, vol. 10, no. 3, pp. 263–274, 2014.

[7] A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux, "Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," *IEEE Trans. Robot. and Automat.*, vol. 19, no. 5, pp. 842–853, Oct. 2003.

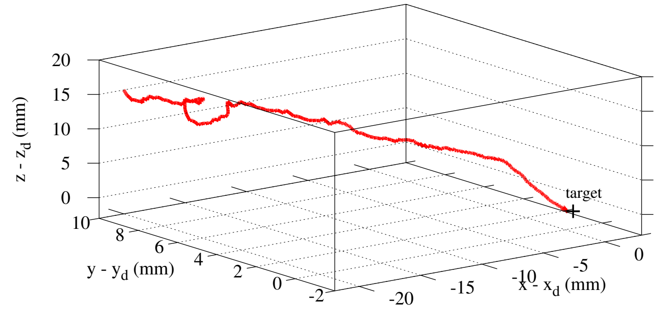

Fig. 15. The 3D trajectory of the image-guided positioning experiment.

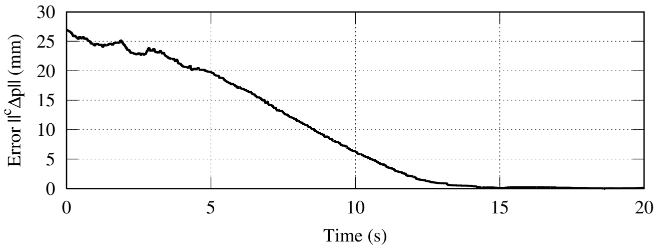

Fig. 16. Magnitude of the error $\|{}^c\Delta\mathbf{p}\|$ obtained during the image-guided positioning experiment.

[8] F. Nageotte, C. Doignon, M. de Mathelin, P. Zanne, and L. Soler, "Circular needle and needle-holder localization for computer-aided suturing in laparoscopic surgery," in *Proc. SPIE Medical Imaging 2005: Visualization, Image-Guided Procedures, and Display*, vol. 5744, 2005, pp. 87–98.

[9] F. Nageotte, P. Zanne, C. Doignon, and M. de Mathelin, "Visual servoing-based endoscopic path following for robot-assisted laparoscopic surgery," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2006, pp. 2364–2369.

[10] R. Haralick, D. Lee, K. Ottenburg, and M. Nolle, "Analysis and solutions of the three point perspective pose estimation problem," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun 1991, pp. 592–598.

[11] C. Doignon, F. Nageotte, B. Maurin, and A. Krupa, "Model-based 3-d pose estimation and feature tracking for robot assisted surgery with medical imaging," in *From Features to Actions - Unifying Perspectives in Computational and Robot Vision. Workshop of IEEE Int. Conf. Robotics and Automation*, 2007, pp. 1–10.

[12] M. Allan, S. Ourselin, S. Thompson, D. Hawkes, J. Kelly, and D. Stoyanov, "Toward detection and localization of instruments in minimally invasive surgery," *IEEE Trans. Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 1050–1058, April 2013.

[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, 2004.

[14] J.-J. Slotine and W. Li, *Applied Nonlinear Control*, 1st ed. Upper Saddle River, NJ: Prentice Hall, 1991.

[15] F. L. Markley and D. Mortari, "How to estimate attitude from vector observations," in *Proc. AAS/AIAA Astrodynamics Specialist Conference*, vol. 103, no. 3, 1999, pp. 1979–1996.

[16] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, vol. 10, no. 2, pp. 47–53, Jun. 1969.

[17] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, W. Lin, and P. Li, "Adaptive image-based positioning of RCM mechanisms using angle and distance features," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2015, pp. 5403–5409.

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