Robust trocar detection and localization during robot-assisted endoscopic surgery

Lin Dong and Guillaume Morel
Sorbonne Universités UPMC Univ. Paris 06, UMR 7222, ISIR, Paris, France.
Email: {dong, morel}@isir.upmc.fr

***Abstract*—In endoscopic surgery, trocars impose kinematic constraints. Whenever a robot manipulates endoscopic instruments, it needs to know the trocar location with respect to its base frame. In the literature, this knowledge is acquired thanks to an installation / registration procedure, prior to the operation.**

In this paper, we are considering a comanipulation scenario: Both the robot and the surgeon hold the instrument. All along a procedure, the instrument can be inserted into or removed from different trocars by the surgeon. It is needed to detect in real time whether the instrument is inserted in a trocar and to compute the fulcrum coordinates. The proposed algorithm is described in terms of both theory and practical realization. Its effectiveness is verified experimentally.

I. Introduction

Endoscopic surgery is an approach of Minimally Invasive Surgery (MIS). At the beginning of an endoscopic surgery procedure, small incisions are made to install trocars, through which an endoscope and elongated instruments are introduced into patient's body. During the procedure, the surgeon maneuvers the instruments through the trocars to perform different surgical tasks under the supervision of the video obtained through the endoscope.

The existence of trocars creates a kinematic constraint which limits the surgical instrument motion to 4 Degrees of Freedom (DOFs): three independent rotations around the insertion point and one translation along the instrument longitudinal axis. When a robot is used to manipulate instruments, it is crucial to know the trocar position information with respect to the robot base body. The question of identifying this location has been the object of dense research in the past decades.

An option consists in using an external localizer. For example, in [1], a registration procedure consists, for the surgeon, in moving the endoscope around the fulcrum, while an external stereo camera pair watches the scene. The lines corresponding to the endoscope axis are extracted from several images and, thanks to a Hough transform, their intersection is computed to form a 3D trocar position estimation. In [2] and [3], at the initial calibration procedure, a camera is fixed to manually select 3D lines representing instrument's axis, whose intersection corresponds to the insertion point. This insertion point is then upgraded using a Hand-Eye matrix and the robot geometric model.

In order to avoid the use of external equipment, direct registration/installation of the robot is more often used. This is the case when using a robot exhibiting a Remote Center of Motion (RCM). Using such a mechanism requires a precise installation of the robot base body in the workspace prior to the operation, in such a way that its RCM precisely fits with patient's entry point in order to avoid tissue damage. An example is the da Vinci robot, which is made up of four interdependent arms mounted on a single base. Each of its arms has a RCM in order to respect the constraint formed by the trocar, [4]. The robot installation procedure requires a passive arm to position the base body of each active arm. A simpler option consists in directly placing the robot on the patient, as proposed in [5] (endoscope holder) or [6] (instrument holder). Here, there is no need for an extra passive arm to position the robot base as the holder is automatically centered on the trocar. However, for all the RCM-based solutions, in the event of robot relocation during the procedure, the realignment of the robot arms to trocars requires a complete new installation process.

If the robot is to be used at several trocar locations during the same operation, a preferable approach is to use a 6 DOF robot without RCM to avoid re-installation. As a price for versatility, extra work is to be done in order to guarantee that the fulcrum constraint is respected. In [7], a fully actuated 6 DOF robot equipped with a force sensor is proposed. The force sensor is used both to control the movements while minimizing forces at the trocar and to estimate the fulcrum location in real time.

Exploiting a force sensor raises concerns in terms of cost, robustness, and compatibility with OR constraints. To avoid using such an equipment, the 6 DOF robot can be partially actuated and equipped with two passive joints at the wrist. The instrument can thus naturally rotate around the fulcrum point while limiting forces exerted to that point, see e.g. [8]. The AESOP robot, used in [9], makes use of such joints. To compute the online trocar point position, joint position data is collected. An algorithm that computes the best intersection between instrument axes at successive locations is used. This method does not require precise positioning of the robot, thus the setup procedure is facilitated. However, the algorithm proposed in [9] is suboptimal as it uses an average filter of a series of two-point estimates. Moreover, this solution is built on the assumption that i) the instrument is indeed inserted into a trocar; ii) the entry point does not move.

In the context of comanipulated endoscopic surgery (Sec. II) these hypotheses do not hold: The problem is not only to localize the trocar but also to detect the trocar presence. An adapted mathematical approach, pertaining to

Least Square (LS) optimization, is proposed in Sec. III. Its practical implementation is detailed in Sec. IV. It is based on rules for selecting appropriate data to feed the LS algorithm and criteria to robustly and rapidly detect the trocar presence. Results shown in Sec. V prove the efficiency of the proposed method.

II. Specific aims

A. Context: comanipulation

The present paper is part of a research aimed at studying comanipulation for laparoscopic surgery. The concept of comanipulation indicates that the surgeon and the robot together maneuver the instrument during the surgical procedure. Different from telesurgery, comanipulated surgery requires the surgeon to stand beside the patient's bed. Apart from laparoscopic surgery, comanipulation has been widely employed in different kinds of surgeries such as total knee arthroplasty, hip replacement surgery, neurosurgery, prostate biopsy, etc.

In this work, we use a 6 DOF robot named Achille, manufactured by Haption company, as a comanipulator. It has 6 joints: The first three are motorized and the last three are passive revolute joints intersecting at the wrist point. The sampling rate of Achille is 1000 Hz.

As illustrated in Fig. 1, the surgeon and the robot Achille simultaneously manipulate the instrument. The robot displays force fields aimed at guiding the surgeon's gesture or filtering the surgeon's tremor during fine movements. Some of these functions require, for the implementation, an estimation of the trocar location, see *e.g.* [10]. This estimation is to be run on the fly, while the robot and the surgeon perform a comanipulated task.

In this paper, without loss of generality, we consider that, in the free mode, the surgeon can manipulate the instrument through a trocar to perform a given gesture. He/she may remove it from the trocar, and insert it through another incision. The robot never imposes movements.

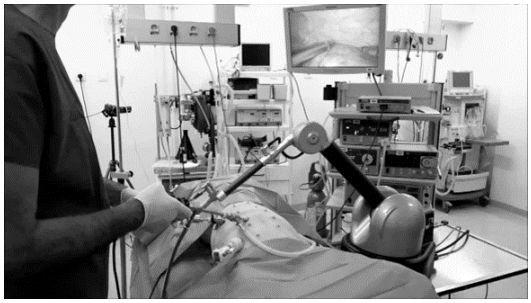

Fig. 1. Comanipulation for assistance to endoscopic surgery

We then look for an algorithm that can robustly detect the trocar presence and further obtain its position in real time without requiring any prior information or registration. The algorithm shall work without being able of imposing a movement. Also, it shall allow to detect position changes of both patient and robot base without readjustment of the equipment during the surgical operation.

B. Principle of trocar position estimation

Figure 2 illustrates the principle used for trocar localization. The reference frame is a frame $\mathcal{F}$ attached to the base of the robot. The coordinates ${}^0\mathbf{p}_F$ of the fulcrum point F in $\mathcal{F}$ are to be determined from geometric measures in q configurations. At the i^{th} configuration, the instrument axis is a straight line passing through a given point P_i, and directing along a given unit vector $\mathbf{z}_i$. It is assumed that, from the joint position sensors of the robot, and thanks to a kinematic model, we can measure the position ${}^0\mathbf{p}_{P_i}$ of P_i and the components ${}^0\mathbf{z}_i$ of $\mathbf{z}_i$ in frame $\mathcal{F}$, $\forall i \in \{1..q\}$.

In theory, whenever the instrument is inserted into a trocar, all the lines intersect at the fulcrum F. In practice, due to measurement noise, backlash between the trocar and the instrument, tissue deformations or calibration errors in the robot kinematic model, successive lines will not perfectly intersect. An optimization algorithm is therefore to be used to estimate the "best fulcrum location", given a set on straight lines. This is done in the next section.

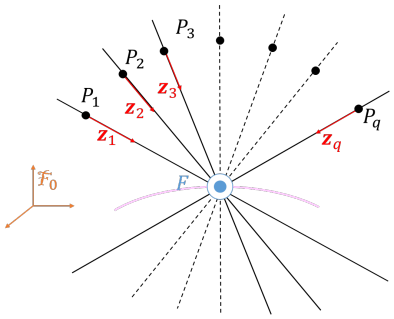

Fig. 2. Principle of trocar detection: Ideally, the fulcrum point F is the intersection of the instrument axes $(P_i, \mathbf{z}_i)$, $i \in \{1 \cdots q\}$ measured from q different instantaneous robot configurations.

III. Least Square Algorithm

A. Estimation of trocar position

To estimate the trocar location from a set of q straight lines $(P_i, \mathbf{z}_i)$, a simple solution is to find the point F whose average distance from the given lines is minimal.

As illustrated in Fig. 3, given a point F, and the i^{th} straight line $(P_i, \mathbf{z}_i)$, the vector $\mathbf{e}_i$ from F to its projection F_i' onto $(P_i, \mathbf{z}_i)$ is given by:

$$\mathbf{e}_i = \mathbf{d}_{F_i'F} = \mathbf{d}_{FP_i} - \mathbf{d}_{F_i'P_i} \ , \tag{1}$$

where $\mathbf{d}_{AB}$ denotes the vector from a point A to a point B. Furthermore:

$$\mathbf{d}_{F_i'P_i} = \left(\mathbf{z}_i^T \mathbf{d}_{FP_i}\right) \mathbf{z}_i \ . \tag{2}$$

Substituting Eq. (2) to Eq. (1) yields:

$$\mathbf{e}_i = \left(\mathbf{I} - \mathbf{z}_i \mathbf{z}_i^T\right) \mathbf{d}_{FP_i} \ . \tag{3}$$

In the base frame $\mathcal{F}_0$, this last equation writes:

$${}^0\mathbf{e}_i = \underbrace{\left(\mathbf{I} - {}^0\mathbf{z}_i \, {}^0\mathbf{z}_i^T\right) {}^0\mathbf{p}_{P_i}}_{=:\ {}^0\mathbf{b}_i} - \underbrace{\left(\mathbf{I} - {}^0\mathbf{z}_i \, {}^0\mathbf{z}_i^T\right)}_{=:\ {}^0\mathbf{A}_i} {}^0\mathbf{p}_F \tag{4}$$

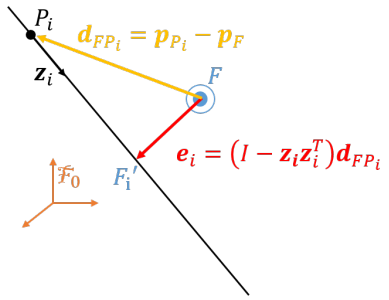

Fig. 3. Projection of F into the instrument axis.

Grouping all the equations for q measurements yields:

$$ {}^0_q\mathbf{e} = {}^0_q\mathbf{b} - {}^0_q\mathbf{A} \cdot {}^0\mathbf{p}_F \ , \tag{5} $$

where:

$$ {}^0_q\mathbf{A} = \begin{pmatrix} {}^0\mathbf{A}_1 \\ {}^0\mathbf{A}_2 \\ \cdots \\ {}^0\mathbf{A}_i \\ \cdots \\ {}^0\mathbf{A}_q \end{pmatrix}, \ {}^0_q\mathbf{b} = \begin{pmatrix} {}^0\mathbf{b}_1 \\ {}^0\mathbf{b}_2 \\ \cdots \\ {}^0\mathbf{b}_i \\ \cdots \\ {}^0\mathbf{b}_q \end{pmatrix}, \ {}^0_q\mathbf{e} = \begin{pmatrix} {}^0\mathbf{e}_1 \\ {}^0\mathbf{e}_2 \\ \cdots \\ {}^0\mathbf{e}_i \\ \cdots \\ {}^0\mathbf{e}_q \end{pmatrix} . \tag{6} $$

In the ideal configuration, when F is the fulcrum location, ${}^0_q\mathbf{e}$ equals zero. With real data, this will not happen. We compute an estimate $\hat{F}_q$ of F by minimizing the norm of ${}^0_q\mathbf{e}$. It is well known that, according to Least Square optimization, the coordinates of this estimate can be computed by:

$$ \begin{aligned} {}^0\mathbf{p}_{\hat{F}_q} &= \underset{{}^0\mathbf{p}_F}{\operatorname{argmin}} \left(\left\| {}^0_q\mathbf{b} - {}^0_q\mathbf{A} \ {}^0\mathbf{p}_F \right\|^2 \right) \\ &= \left({}^0_q\mathbf{A}^T \cdot {}^0_q\mathbf{A} \right)^{-1} \ {}^0_q\mathbf{A}^T \ {}^0_q\mathbf{b} \ . \end{aligned} \tag{7} $$

Practically implementing this algorithm requires answering two main questions. The first one concerns the selection of measures to be included for optimization. The second one concerns the interpretation of the result. Indeed, even if the instrument is not inserted in the trocar, Eq. (7) will provide an estimation. Therefore, after the computation of ${}^0\mathbf{p}_{\hat{F}_q}$, it is necessary to evaluate the effectiveness of this value to be used as an estimation of the trocar position. To do that, we further calculate the average of the norm of the estimated error, denoted as e_q, which is:

$$ e_q = \frac{1}{q} \sum_{i=1}^{q} \left\| {}^0\mathbf{A}_i \cdot {}^0\mathbf{p}_{\hat{F}_q} - {}^0\mathbf{b}_i \right\| . \tag{8} $$

The smaller is the value of e_q, the higher is the probability that the instrument passes through a fixed point.

IV. Trocar Detection and Localization

In order to practically implement the algorithm described above, the general principle is as follows:

Step 1. Built a data list L_q containing q measured data pairs $c_i = \left({}^0\mathbf{p}_{P_i}, {}^0\mathbf{z}_i \right), i = 1, 2, ..., q$.

Step 2. Apply the LS method to L_q and calculate the trocar position ${}^0\mathbf{p}_{\hat{F}_q}$ from Eq. (7).

Step 3. Compute the average of the norm of the estimated error e_q according to Eq. (8).

Step 4. Use e_q to determine whether a trocar is detected or not. If a trocar is detected, then ${}^0\mathbf{p}_{\hat{F}_q}$ is its position, otherwise, ${}^0\mathbf{p}_{\hat{F}_q}$ is abandoned.

In the following, we discuss the realization of the algorithm in detail.

A. Building a data list for trocar estimation

We know that to perform the least square method, more than one measurement is required. Therefore, a data list, denoted as L_q, is built to store q data pairs as the input of the estimation algorithm. The list is implemented with a circular buffer: Once q values have already been stored and a new value is to be considered, the oldest value is discarded. This allows for permanent updating of the data while forgetting the oldest measurement.

To select data pairs to be included list L_q, we should make sure each data pair c_i represents an instrument configuration different from others in the list, so as to ensure the effectiveness and the precision of the least square method. To this aim, two functions are defined in order to evaluate the displacement between two configurations.

1) **Linear displacement of the robot wrist center.** The linear displacement between two robot configurations characterized by $\left({}^0\mathbf{p}_{P_j}, {}^0\mathbf{z}_j \right)$ and $\left({}^0\mathbf{p}_{P_k}, {}^0\mathbf{z}_k \right)$, respectively, is defined by:

$$ d_{j,k} = \left\| \left(\mathbf{I} - {}^0\mathbf{z}_j \, {}^0\mathbf{z}_j^T \right) \left({}^0\mathbf{p}_{P_j} - {}^0\mathbf{p}_{P_k} \right) \right\| \ . \tag{9} $$

Notice that the distance $d_{j,k}$ is not affected by displacements along the instrument axis, as they do not contribute to change Eq. (3).

2) **Orientation of the instrument axis.** The orientation displacement between two robot configurations characterized by $\left({}^0\mathbf{p}_{P_j}, {}^0\mathbf{z}_j \right)$ and $\left({}^0\mathbf{p}_{P_k}, {}^0\mathbf{z}_k \right)$, respectively, is defined by:

$$ \theta_{j,k} = \operatorname{acos}({}^0\mathbf{z}_j, {}^0\mathbf{z}_k). \tag{10} $$

Notice that any rotation of the instrument around $\mathbf{z}$ does not affect $\theta_{j,k}$.

From these distances, the pseudo code of building a data list L_q containing q data pairs is detailed in Algorithm 1 where:

- measure_from_robot() reads the robot joint position and returns the pair $\left({}^0\mathbf{p}_P, {}^0\mathbf{z} \right)$
- add_to_circular_buffer(L_q,c,q) adds c to the list L_q by concatenation, and, if the list length equals q, removes the oldest value of the list.
- δ_d and δ_θ are the linear and rotational displacements thresholds beyond which any new value shall be included.

Choosing q is also of primary importance. In general, to filter out the noise and to obtain a precise estimation, q shall be large. However, a large list takes longer time to be filled in, which slows down the estimation procedure and causes delays.

To deal with this dilemma, we build 2 lists, a small one L_m, with m data pairs, whose main purpose is to rapidly

Algorithm 1 Algorithm building a data list L_q with q data pairs

Initialization:
$c_1 \leftarrow$ measure_from_robot()
$L_q \leftarrow \{c_1\}$
$i \leftarrow 2$
Periodic function (at sampling rate): :
$c_i \leftarrow$ measure_from_robot()
if $[(d_{i-1,i} > \delta_d)$ or $(\theta_{i-1,i} > \delta_\theta)]$ **then**
$L_q \leftarrow$ add_to_circular_buffer(L_q,c_i,n)
$i \leftarrow i + 1$
end if

detect the existence of the trocar, and a larger one L_n, containing n components, which is mainly used to provide a stable and precise trocar position when the trocar has been detected. We denote the estimated trocar positions from these two lists as ${}^0\mathbf{p}_{\hat{F}_m}$ and ${}^0\mathbf{p}_{\hat{F}_n}$, respectively, and the associated trocar position estimation errors as e_m and e_n, respectively.

B. Trocar detection

In this section, we detail a second algorithm used to detect whether the trocar is present.

At the initialization, it is supposed that the instrument is not inserted into a trocar. Then building lists L_m and L_n starts. Whenever the shortest list, L_m, is full, a least square estimation is performed according to Eq. (7) and the associated error e_m is computed thanks to Eq. (8).

To detect presence of a trocar from this configuration, a test is made on the error e_m. When it is small enough (smaller than a threshold δ_{e_m}), this indicates that the instrument probably passes through a fixed point, which most likely is a trocar. However, from the large set of experiments that were performed to assess the robustness of the approach, a special configuration was identified leading to false positive detection: The surgeon may move the instrument outside the patient (not inserted in a trocar) in such a way that the instrument orientation changes while P does not move. This is due to the fact that, with our comanipulation robot, P is the center of the robot passive wrist. In such a configuration, the LS algorithm detects P as the fulcrum point. To eliminate this false positive detection, a second criterion is used for trocar detection: The distance between P and $\hat{F}$ shall be larger than a threshold Δ_{depth}, ensuring that the detected point is not P.

When a trocar has been detected, the algorithm shall detect when the surgeon extracts the instrument from the trocar. Two configurations occur.

1) When the long list L_n is not yet filled, we still have to rely on L_m to make a decision. Since the signal e_m is noisy due to the smallness of m, we detect an extraction of the instrument from the trocar when e_m is larger than a given threshold $\Delta_{e_m} > \delta_{e_m}$.
2) When L_n is full, we rely on e_n, which is less noisy than e_m, and compare it to a threshold Δ_{e_n}.

This corresponds to the pseudo-code of Algorithm 2, where Length(L) returns the number of elements already included in a list L and Least_square_estimation(L) performs a LS estimation from L and returns the estimated position and the associated error. Algorithm 2 is run at each sampling period, in parallel to Algorithm 1 that builds the lists.

Algorithm 2 Algorithm for trocar detection

Initialization:
trocar_is_present $\leftarrow$ false
Periodic function (at sampling rate):
if trocar_is_present = false **then**
if Length(L_m) $= m$ **then**
$\left({}^0\mathbf{p}_{\hat{F}_m}, e_m\right) \leftarrow$ Least_square_estimation(L_m)
if $e_m < \delta_{e_m}$ and $\left\|\mathbf{d}_{\hat{F_m}P}\right\| > \delta_{\text{depth}}$ **then**
trocar_is_present $\leftarrow$ true
$\hat{F} \leftarrow \hat{F}_m$
end if
end if
else
if Length(L_n) $< n$ **then**
$\left({}^0\mathbf{p}_{\hat{F}_m}, e_m\right) \leftarrow$ Least_square_estimation(L_m)
if $e_m > \Delta_{e_m}$ **then**
trocar_is_present $\leftarrow$ false
else
$\hat{F} \leftarrow \hat{F}_m$
end if
else
$\left({}^0\mathbf{p}_{\hat{F}_n}, e_n\right) \leftarrow$ Least_square_estimation(L_n)
if $e_n > \Delta_{e_n}$ **then**
trocar_is_present $\leftarrow$ false
else
$\hat{F} \leftarrow \hat{F}_n$
end if
end if
end if

V. Experimental results

Experiments for trocar detection and localization are performed with Achille to identify the effectiveness of the proposed solution.

Thresholds for trocar detection are experimentally tuned as follows:

- Length of the large data list L_n: $n = 100$.
- Length of the small sublist L_m: $m = 20$.
- Thresholds for inclusion into a data list: $\delta_d = 0.005$ m and $\delta_\theta = 5°$.
- Thresholds for trocar insertion detection: $\delta_{e_m} = 0.009$ m and $\delta_{\text{depth}} = 0.05$ m.
- Thresholds for trocar extraction detection: $\Delta_{e_m} = \Delta_{e_n} = 0.025$ m.

We use a laparoscopic trainer box to simulate a patient's abdomen where four trocars are attached. The user inserts the instrument into four trocars one by one and manipulates the instrument simultaneously with the Achille robot. For each

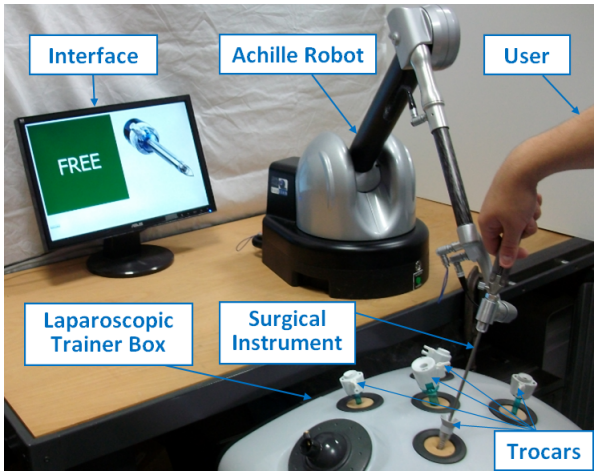

Fig. 4. Trocar detection and localization experiment setup

trocar, the algorithm records the calculated trocar positions as well as the time when trocar status changes, i.e, from not detected to detected, and vice-versa.

Firstly, robustness and rapidness of the algorithm are evaluated. The instrument is inserted in and extracted out of the four trocars in a random order. The time duration of keeping instrument inserted into one trocar is about 10 seconds while the time outside of a trocar lasts about 2 seconds. The total experiment lasts 23 minutes, and involves 108 insertion-extraction cycles. A camera is used to record the whole process.

	Insertion	Extraction	Total
Video	108	108	216
Robot	108	108	216
Success Rate	100%	100%	100%

TABLE I
EXPERIMENTAL RESULTS FOR ROBUSTNESS VERIFICATION

We can visually observe on the video the time when the instrument is inserted in or extracted out of trocars and manually time stamp these events (with a precision of one frame, *i.e.* 25 ms). Using these time stamps as a ground truth, we can measure, from the robot data recorded and synchronized with the video, the delay between the real insertion (resp. extraction) and the detected trocar insertion (resp. extraction). In Table I, we see that all the insertions and extractions are correctly detected, which assesses the robustness of the proposed solution. In Table II, the statistics for the measured delays are reported. The average time delay for insertion and extraction are 1.9 s and 0.79 s respectively, with corresponding standard deviations 0.54 s and 0.29 s. At the scale of a surgical procedure, this rapidness can be viewed as acceptable.

The next step is to assess the precision of the method. Unfortunately, there is no known ground truth for a real trocar position. First, backlash appears between the trocar and the instrument. Second, the deformations of the simulated tissue as well as those of the instrument introduce trocar

Delay (s)	Insertion	Extraction
Max	4.3	1.4
Min	1.1	0.17
Mean	1.9	0.79
Standard Deviation	0.54	0.29

TABLE II
EXPERIMENTAL RESULTS FOR RAPIDNESS VERIFICATION

position deviations. For these reasons, rather than verifying the algorithm precision, which is formally impossible, we verify its numerical consistency. A second experiment is designed, where the relative position of the trainer box with respect to the robot base changes. A given robot-trainer box relative position and orientation is called a configuration. In this experiment, we used three different configurations, represented by C_1, C_2 and C_3, respectively. Three sets of data recording estimated trocar positions are collected.

Using k-means clustering method, for each configuration we can identify four groups of points and their corresponding centroids, depicted in Fig. 5. Centroids of trocars obtained from C_1 are illustrated in red circles, C_2 in blue crosses and C_3 in green stars. The centroids of clustered point clouds are used as the mean detected trocar positions.

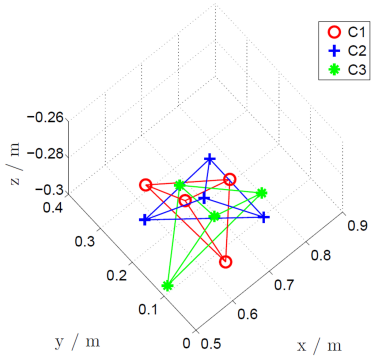

Fig. 5. Central positions of all trocars in 3D view

Since the relative positional relationship of the four trocars in space is independent of their relative positions to the robot, for the above three configurations, by some computation, we should obtain fixed trocar internal distribution information. More specifically, the calculated distance between a given trocar T_m and another trocar T_n, denoted as $d_{T_mT_n}$, is supposed to be theoretically the same for C_1, C_2 and C_3. Table III gives all the distances between any two trocars for the three configurations. The average distance of the three configurations and standard deviations are also shown.

We observe that distances between any two trocar centroids for the three configurations exhibit low standard deviations (typically 1 mm), indirectly assessing the numerical consistency of the proposed algorithm and its "precision".

This is visually illustrated in Fig. 6 and Fig. 7. To obtain these plots, we performed a rigid transformation for data sets of C_1 and C_3 to align them with data set of C_2. After the rigid transformation, it is clear to see that the centroids of the four trocars under three configurations closely coincide,

	C_1	C_2	C_3	Mean	std
$\|\mathbf{d}_{T_1T_2}\|$(m)	0.1719	0.1710	0.1701	0.1710	9.0×10^{-4}
$\|\mathbf{d}_{T_1T_3}\|$(m)	0.2430	0.2453	0.2460	0.2448	1.6×10^{-3}
$\|\mathbf{d}_{T_1T_4}\|$(m)	0.1221	0.1229	0.1212	0.1221	8.5×10^{-4}
$\|\mathbf{d}_{T_2T_3}\|$(m)	0.1733	0.1742	0.1734	0.1736	4.9×10^{-4}
$\|\mathbf{d}_{T_2T_4}\|$(m)	0.1244	0.1234	0.1245	0.1241	6.1×10^{-4}
$\|\mathbf{d}_{T_3T_4}\|$(m)	0.1222	0.1235	0.1229	0.1229	6.5×10^{-4}

TABLE III
DISTANCES BETWEEN TROCAR CENTROIDS

indicating that the relative positional structure obtained from different configurations is consistent.

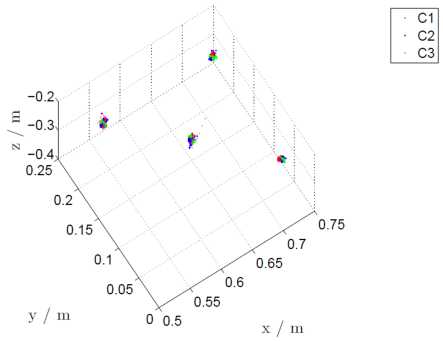

Fig. 6. Experimentally obtained trocar positions in 3D view after transformation

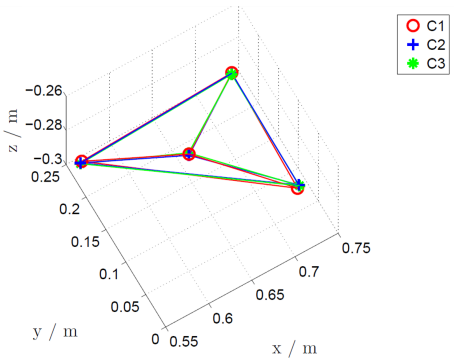

Fig. 7. Central positions of all trocars in 3D view after transformation

VI. CONCLUSION

This paper concerns the real-time trocar detection and localization, which is essential to solve the kinematic constraint and fulcrum effect problems. A solution based on the least square principle is proposed and the practical implementation is described in detail and realized. Experimental results show its practical efficiency. Finally, the proposed approach exhibits the following properties:

- There is no need for registration prior to the operation, which saves the whole surgery time.
- There is no need of external sensors.
- The movement of the patient or the robot during the operation is allowed thanks to the algorithm robustness and its forgetting capabilities.
- Change of trocars during the operation is allowed without new registration.
- The algorithm exhibit numerical stability, precision, robustness and rapidness.

Future validation on animal experiment (where breathing or other movements can be involved) is to be programmed for further experimental validation.

ACKNOWLEDGMENT

This work was partly supported by the project FLUOROMIS funded by French FUI and by french state funds managed by the ANR within the Investissements d'Avenir programme (Labex CAMI) under reference ANR-11-LABX-0004.

REFERENCES

[1] B. Rosa, C. Gruijthuijsen, B. Van Cleynenbreugel, J. Vander Sloten, D. Reynaerts, and E. Vander Poorten, "Estimation of optimal pivot point for remote center of motion alignment in surgery," *International journal of computer assisted radiology and surgery*, vol. 10, no. 2, pp. 205–215, 2015.

[2] A. Agustinos, R. Wolf, J. Long, P. Cinquin, and S. Voros, "Visual servoing of a robotic endoscope holder based on surgical instrument tracking," in *2014 5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob) , Aguust 12-15, 2014, Sao Paulo, Brazil*. IEEE, 2014, pp. 13–18.

[3] R. Wolf, J. Duchateau, P. Cinquin, and S. Voros, "3d tracking of laparoscopic instruments using statistical and geometric modeling," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI*. Springer, 2011, pp. 203–210.

[4] J. Troccaz, *Medical robotics*. John Wiley & Sons, 2013.

[5] P. Berkelman, P. Cinquin, E. Boidard, J. Troccaz, C. Letoublon, and J.-M. Ayoubi, "Design, control, and testing of a novel compact laparascopic endoscope manipulator," *Journal of Systems and Control Engineering*, vol. 217, no. 14, pp. 329–341, 2003.

[6] N. Zemiti, G. Morel, T. Ortmaier, and N. Bonnet, "Mechatronic design of a new robot for force control in minimally invasive surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 2, pp. 143–153, 2007.

[7] A. Krupa, C. Doignon, J. Gangloff, M. de Mathelin, L. Soler, and G. Morel, "Towards semi-autonomy in laparoscopic surgery through vision and force feedback control," in *Experimental Robotics VII*. Springer, 2001, pp. 189–198.

[8] B. Herman, B. Dehez, K. Tran Duy, B. Raucent, E. Dombre, and S. Krut, "Design and preliminary in vivo validation of a robotic laparoscope holder for minimally invasive surgery," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 5, no. 3, pp. 319–326, 2009.

[9] T. Ortmaier and G. Hirzinger, "Cartesian control issues for minimally invasive robot surgery," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), Takamatsu, Japan, October 31 - November 5*, vol. 1. IEEE, 2000, pp. 565–571.

[10] C. Poquet Torterotot, M.-A. Vitrani, and G. Morel, "Proximal comanipulation of a minimally invasive surgical instrument to emulate distal forces," in *Proceedings of the 4th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery (CRAS2014), Genoa, Italy, October 14-16*, 2014, pp. 48–51.