Robust Real-Time Needle Tracking in 2-D Ultrasound Images Using Statistical Filtering

Kim Mathiassen, *Student Member, IEEE*, Diego Dall'Alba, Riccardo Muradore, *Member, IEEE*, Paolo Fiorini, *Fellow, IEEE*, and Ole Jakob Elle

***Abstract*— Percutaneous image-guided tumor ablation is a minimally invasive surgical procedure for the treatment of malignant tumors using a needle-shaped ablation probe. Automating the insertion of a needle by using a robot could increase the accuracy and decrease the execution time of the procedure. Extracting the needle tip position from the ultrasound (US) images is of paramount importance for verifying that the needle is not approaching any forbidden regions (e.g., major vessels and ribs), and could also be used as a direct feedback signal to the robot inserting the needle. A method for estimating the needle tip has previously been developed combining a modified Hough transform, image filters, and machine learning. This paper improves that method by introducing a dynamic selection of the region of interest in the US images and filtering the tracking results using either a Kalman filter or a particle filter. Experiments where a biopsy needle has been inserted into a phantom by a robot have been conducted, guided by an infrared tracking system. The proposed method has been accurately evaluated by comparing its estimations with the needle tip's positions manually detected by a physician in the US images. The results show a significant improvement in precision and more than 85% reduction of 95th percentile of the error compared with the previous automatic approaches. The method runs in real time with a frame rate of 35.4 frames/s. The increased robustness and accuracy can make our algorithm usable in autonomous surgical systems for needle insertion.**

***Index Terms*— Biomedical image processing, Kalman filtering, medical robotics, needle insertion, particle filtering, ultrasound (US) imaging.**

I. Introduction

PERCUTANEOUS image-guided tumor ablation is a minimally invasive surgical procedure for the treatment of malignant tumors [1]. A needle-shaped probe is inserted from the outside into the patient in order to destroy one or more pathological areas through the application of energy or chemicals. The advantages compared with surgical resection are the potential to destroy only a minimal amount of healthy tissue [2], lower cost [1], and faster recovery. Imaging techniques used for monitoring the placement of ablation probe include ultrasound (US), computed tomography (CT), and magnetic resonance imaging [3]. However, US guidance is the most commonly used modality for thermal ablation [1] and biopsy.

Automating the insertion of a needle by using a robot could increase the accuracy and decrease the execution time of the procedure [4], [5]. Moreover, in cryoablation treatments, robot-aided insertion will reduce the number of CT scans the patient has to undergo. The needle is the end-effector of the robot and its tip should be positioned inside the human body to hit a target from a specific insertion point. Then, estimating the needle tip position from the US images is of paramount importance for verifying that the needle is not approaching any forbidden regions (e.g., major vessels, ribs, and nerves) and for eventually updating the planned trajectory.

There are several reasons for tracking the needle tip during the insertion. The estimated needle tip may be used directly by the controller of the robot inserting the needle as a feedback variable. In addition, when a robot is inserting the ablation probe, the insertion trajectory is planned using preoperative images and errors in the registration or movement during the intervention may cause deviations from the precalculated plan. In manual insertion, it is hard even for expert radiologists to position the device accurately: an autonomous tracking system for the needle could be very helpful to increase the precision.

Robotic approaches to needle insertion are shown in [6] and [7], but none of these use US image feedback during the insertion. Pollock *et al.* [6] use CT, while Boctor *et al.* [7] use a preoperative 3-D US model and a magnetic tracker. A real-time algorithm for finding straight biopsy needles is presented in [8] based on a modified version of the Hough transform. However, a detailed accuracy analysis of the needle tip detection is not provided. In [9], two real-time algorithms for finding curved needles in 2-D US images are presented, one based of the Hough transform and one approximating the cured needle as an arc of constant radius. The algorithms find points on the needle, but do not provide a reliable estimation of the needle tip position. Those algorithms are compared with a novel method to find biopsy needles in [10] on transrectal US images, where both of the above algorithms are found to give biased results. The algorithm in [10] defines an objective function from three needle tip metrics, and selects the needle tip based on this function. The results are very accurate,

Manuscript received March 8, 2016; accepted June 25, 2016. Date of publication August 25, 2016; date of current version April 11, 2017. Manuscript received in final form June 30, 2016. This work was supported by the European Union Seventh Framework Programme (FP7/2007-2013) under Grant 270396 (I-SUR) and Grant 248960 (SAFROS). Recommended by Associate Editor M. Fujita.

K. Mathiassen and O. J. Elle are with the Department of Informatics, University of Oslo, N-0373 Oslo, Norway, and also with The Intervention Center Oslo University Hospital, N-0424 Oslo, Norway (e-mail: kimmat@ifi.uio.no; oelle@ous-hf.no).

D. Dall'Alba, R. Muradore, and P. Fiorini are with the Altair Robotics Laboratory, Department of Computer Science, University of Verona, 37134 Verona, Italy (e-mail: diego.dallalba@univr.it; riccardo.muradore@univr.it; paolo.fiorini@univr.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2016.2587733

but all the images of the biopsy procedure need to be available before the execution of the algorithm. Therefore, the method is not suitable for real-time applications. A Gabor filter is used in [11] for localizing the needle, and is improved in [12] by introducing an entropy-based parameter tuning scheme. Kalman filtering of the result is introduced in [13] to reduce estimation noise.

Segmentation of 3-D US images is done in [14] for brachytherapy needles and [15] for biopsy needles. Another method for localization and tracking of biopsy needles is presented in [16]. This method uses the random sample consensus algorithm and Kalman filtering, and adapts the region of interest (ROI) to decrease the execution time. These methods have a much higher computational cost than the 2-D segmentation methods and can only be used in real-time applications using dedicated graphics processing unit [17]. Some recent attempts have been proposed by moving the US probe to track the needle. In [18], the image plane is perpendicular to the needle and the US probe is moved by a robot to continually see the tip. The main drawback of this approach is that neither the target itself nor the needle axis is visible in the US image. Chatelain *et al.* [19] present a method to localize and track a needle by combining the random sample consensus algorithm with Kalman filtering on the images from a 3-D US probe. It also presents a visual servoing algorithm for keeping the needle within the field of view.

Rather than segmenting the needle to find the tip from gray-scaled US images, it is possible to use stylet vibration and power Doppler, as shown in [20] and [21]. Even though this method guarantees a very high accuracy, it requires a mechanical device to induce the vibration of the needle increasing the overall complexity and cost.

In this paper, we present a new real-time needle tip estimation method for rigid needles based on 2-D US images. The method presented here improves the one proposed in [22] in the following aspects.

1) Online adaptation of the ROI for estimating the needle axis in a more robust and reliable way.
2) *Implementation of Statistical Filtering:* Kalman filter (KF) and particle filter (PF) have been used to improve the accuracy and precision of the tip tracking, to filter out the noise, and to cope with outliers.
3) The algorithm can also rely on velocity measurements to improve the tracking accuracy when the insertion is performed by a robotic system.

The performance of the algorithm in manual and robotic insertions is deeply investigated and compared, and the differences between using a KF and a PF for this specific application are evaluated. The main objective of this paper is to design an accurate and robust observer for a robotic system inserting a needle, while a secondary objective is to provide a method that may assist physicians inserting a needle manually.

This paper is organized as follows. In Section II, the proposed approach is presented, and in Section III, the experimental setup is described. Experimental results are reported in Section IV, whereas their discussion is in Section V. Finally, the conclusions are drawn in Section VI.

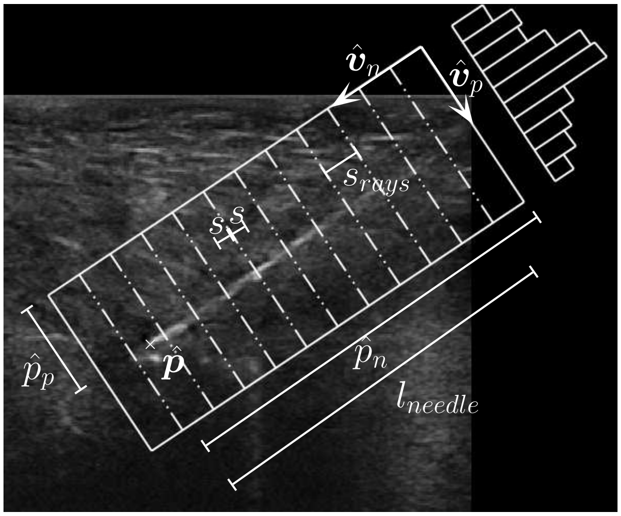


Fig. 1. Sample US image showing the ROI (solid lines) with the rays (dashed-dotted lines). The smoothed histogram of the ray score values is shown in the top-right corner.

II. Estimation Algorithms

The problem of tracking the needle tip in a US image can be decomposed in three sequential phases:

1) needle detection in the image;
2) estimation of the needle axis[1] (i.e., orientation $\hat{\boldsymbol{v}}_n$ and entry point $\hat{p}_p$);
3) localization of the needle tip (i.e., position along the needle axis $\hat{p}_n$).

This paper will focus on the last two phases of the procedure. A robot or a radiologist will insert the needle, and when it is inside the phantom by a certain length, the estimation algorithm is activated. The true position of the needle tip is denoted by $\boldsymbol{p}$ and the true needle axis is denoted by $\boldsymbol{v}_n$. The estimated needle tip position and the needle axis are denoted by $\hat{\boldsymbol{p}} = [\hat{p}_p\ \hat{p}_n]^T$ and $\hat{\boldsymbol{v}}_n$, respectively. These values are shown in Fig. 1.

A. Needle Axis Estimation

The method described in [9] has been improved to find the needle axis in a more reliable way. In this section, the original method is briefly recalled and our changes highlighted. The algorithm is based on a search method, where a range of insertion angles are checked and the one with the highest score is selected. The range of insertion angles is a parameter of the method, and will be discussed in more detail later. We start by looking at the algorithm when it checks one insertion angle, i.e., one iteration of the search.

1) One Search Step: The algorithm presented in this section is summarized in Algorithm 1, and references to the lines in the algorithm will be given throughout the section. The algorithm is defined as a function with inputs listed between parentheses after the function name, and the output listed in brackets before the function name. First, an ROI needs to be defined (line 2). In [9] the authors suggest to "*Choose a rough*

[1] Notation: in this paper, we use the symbol $\hat{x}$ for the estimation of x.

Algorithm 1 One Search Step in Estimating the Needle Axis

```
 1: function [$\bar{p}_p$, $H_{max}$]=AXISSEARCHSTEP($\boldsymbol{I}$, $\bar{\theta}$, $\hat{p}_n[k-1]$)
 2:     Define ROI using $\bar{\theta}$ and $\hat{p}_n[k-1]$ from previous step
 3:     Reduce $N_{rays}$ if necessary
 4:     // Find score values
 5:     for $k \in [1, N_{rays}]$ do
 6:         $R(r_k, y) \leftarrow \frac{1}{2s+1}\sum_{i=r_k-s}^{r_k+s} I(i, y) \quad \forall\, y \in ROI$
 7:         $R^{\Delta}(r_k, y) \leftarrow R(r_k, y+1) - R(r_k, y) \forall\, y \in ROI$
 8:         for $y \in ROI$ do
 9:             $S_{lead}(r_k, y) \leftarrow \sum_{w=0}^{w_{max}} R^{\Delta}(r_k, y+w)$
10:             $S_{lag}(r_k, y) \leftarrow \sum_{w=0}^{w_{max}} R^{\Delta}(r_k, y-w-1)$
11:             $S(r_k, y) \leftarrow S_{lag} - S_{lead} - |S_{lag} + S_{lead}|$
12:         end for
13:     end for
14:     // Create histogram and smooth it
15:     $H(b) \leftarrow 0 \quad \forall\, b \in [1, N_{bins}]$
16:     for $k \in [1, N_{rays}]$ do
17:         for $y \in ROI$ do
18:             $b \leftarrow \left\lfloor \frac{N_{bins} y}{l_{ray}} \right\rfloor$
19:             $H(b) \leftarrow H(b) + S(r_k, y)$
20:         end for
21:     end for
22:     for $b \in [1, N_{bins}]$ do
23:         $H_S(b) \leftarrow H(b) + \frac{1}{2}H(b-1) + \frac{1}{2}H(b+1)$
24:     end for
25:     // Find points that lie on the needle
26:     $[H_{max}, b_{max}] \leftarrow \max_{b \in [1, N_{bins}]} H_S(b)$
27:     for $k \in [1, N_{rays}]$ do
28:         $[-, p_k] \leftarrow \max_{y \in [b_{max}-3b, b_{max}+3b]} S(r_k, y)$
29:     end for
30:     // Find the average position of the points
31:     $\bar{p}_p \leftarrow \text{mean}(p_k)$
32: end function
```

approximation of the needle's axis line in the US image based on the expected trajectory and define a rectangular ROI." We have modified the method to dynamically adapt the ROI according to the current estimation of the tip position. As said previously, the axis is estimated first, and then, it is used to estimate the tip position. When estimating the axis for the next image, the current tip estimate is used to update the next ROI. In this way, the tip estimate is fed back to the axis estimation; hence, we call this feature *feedback*. This can be seen in Fig. 2, where the estimated needle position along the needle axis $\hat{p}_n$ is given to the axis estimation step.

The main axis of the rotated rectangular ROI is equal to the estimation of the needle direction,[2] $\bar{\boldsymbol{v}}_n$, whereas $\bar{\boldsymbol{v}}_p$ is the perpendicular axis, as shown in Fig. 1. The ROI is then a rectangular box around the needle.

The algorithm requires to cast a number of rays parallel to $\bar{\boldsymbol{v}}_p$ within the ROI, as shown in Fig. 1 (dashed-dotted lines). Each pixel along each ray should be averaged with the pixels on both sides of the ray. The rays R are defined in the US image plane by

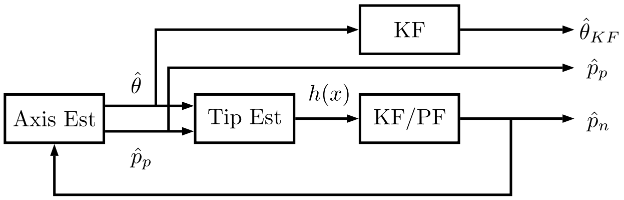


Fig. 2. Diagram describing the different parts in the method. "Axis Est" is the axis estimation described in Sections II-A1 and II-A2. "KF" is the Kalman filter used on the angle $\hat{\theta}$, described in Section II-A3. "Tip Est" is the tip estimation described in Section II-B. "KF/PF" is the Kalman filter or PF used to estimate the tip position, described in Sections II-C and II-D.

$$R(r_k, y) = \frac{1}{2s+1}\sum_{i=r_k-s}^{r_k+s} I(i, y) \tag{1}$$

where r_k is the position of the kth ray in the direction $\bar{\boldsymbol{v}}_n$, y is the position in the $\bar{\boldsymbol{v}}_p$ direction, s is the number of pixels used to calculate the average, and $I(i, y)$ is the pixel grayscale value in position (i, y). $\boldsymbol{I}$ denotes the matrix containing the pixel grayscale values of the entire image. These rays are evenly distributed along $\bar{\boldsymbol{v}}_n$, and the values should be calculated for all y in the ROI (line 6). By averaging the pixel values, the noise and features that are not parallel to $\bar{\boldsymbol{v}}_n$ are filtered out.

Our method adapts at run time the number of rays N_{rays} along the needle axis (line 3). The spacing between the rays (s_{rays}; see Fig. 1) is the distance from the first ray to the estimated tip position (l_{needle}; see Fig. 1) divided by the number of rays (N_{rays}). As a rule of thumb, the distance between two rays should be larger than twice the averaging width s of the rays, to avoid the rays to overlap. If the distance between the rays (s_{rays}) is smaller than s, the number of rays is decreased. If it is larger, the number of rays is increased.

When the rays are computed, the derivative along $\bar{\boldsymbol{v}}_n$ is evaluated as

$$R^{\Delta}(r_k, y) = R(r_k, y+1) - R(r_k, y) \tag{2}$$

for all r_k and all y in the ROI (line 7). This is the edge detection step of the method. Since finding the derivative in this way yields noisy results, the next steps are needed to filter out the noise because of an averaging effect.

Traversing each ray (along the $\bar{\boldsymbol{v}}_p$ direction) looking at a window of the derivative values R^{Δ}, a score value should be calculated (lines 8–12) using the following equation:

$$\begin{aligned} S(r_k, y) = {} & S_{\text{lag}}(r_k, y) - S_{\text{lead}}(r_k, y) \\ & - |S_{\text{lag}}(r_k, y) + S_{\text{lead}}(r_k, y)| \end{aligned} \tag{3}$$

where S_{lead} and S_{lag} are the sum of the derivative values of the leading and lagging half of the window with size w_{max}. They are given as

$$S_{\text{lead}}(r_k, y) = \sum_{w=0}^{w_{\text{max}}} R^{\Delta}(r_k, y+w) \tag{4}$$

$$S_{\text{lag}}(r_k, y) = \sum_{w=0}^{w_{\text{max}}} R^{\Delta}(r_k, y-w-1). \tag{5}$$

[2] Notation: $\bar{x}$ is used for variables in one search step, while $\hat{x}$ is used for final estimates.

Bright line structures in the image, which are parallel to $\bar{\boldsymbol{v}}_n$, will have a high score value. This will include the needle, but also other structures in the image. The needle will have a high score on approximately the same position along $\bar{\boldsymbol{v}}_p$ for all the rays, while other nonneedle structures will be distributed on different positions on the different rays. A histogram (H) is made using the score values of all the rays (lines 15–21). As the needle will have a high score value on approximately the same ray position, these values will be added together in the same histogram bin, while the nonneedle structures will be distributed on several bins. Then, this histogram is smoothed by adding one half of the bin values from the bins on the right and on the left (lines 22–24). The function $[f_{\max}, x_{\max}] = \max_{x \in \mathcal{S}} f(x)$ finds the maximum of $f(x)$ for all x in the set $\mathcal{S}$, and returns the maximum value of $f(x)$ as $f_{\max}$ and the x that corresponds to the maximum value as $x_{\max}$. A "$-$" symbol indicates that the return value is unused. The last step is performed to detect the needle even if it lies on the boundary of two bins. A smoothed histogram is illustrated in the top-right corner of Fig. 1. The points that belong to the needle axis are selected by looking at the region centered on the highest bin of the smoothed histogram and including three bins on both sides (lines 26–29). For each ray, the position within this region that has the highest score value is considered to be on the needle axis. These positions yield a set of positions that all lie on the needle axis.

The needle can be found by the set of points on the needle axis $\bar{v}_p$, since we assume to work with rigid needles (e.g., cryoablation needles). The average position along $\bar{\boldsymbol{v}}_p$ is the estimated needle axis position for this iteration. This position is denoted by $\bar{p}_p$ and is returned to the search algorithm along with the value of the highest bin of the smoothed histogram, denoted by $H_{\max}$. A summary of one iteration of the search is given in Algorithm 1.

2) Searching for the Needle Axis: The estimated insertion angle $\hat{\theta}$ is found by coarse-fine searching over a range of possible insertion angles. In the original method, an incremental step of 1.5° was used, and the angle showing the highest bin score values $H_{\max}$ was selected as the best estimation $\hat{\theta}$ of the true insertion angle θ. We have improved this approach by implementing a two-step procedure in order to get a higher resolution (see Algorithm 2). First, a search is conducted with a step of 1° (lines 2–4). The search range is from θ_{start} to θ_{end} and can be varied, but this affects the computation time of the method. The angle with the highest score value is selected (lines 5 and 6), and a new search is conducted (lines 7–9). This time an increment of 0.1° is used to refine the estimation on the range ±0.5° around the result of the coarse step. This yields a resolution of 0.1° at a low computational cost. The angle $\bar{\theta}$ and position $\bar{p}_p$ pair that has the highest $H_{\max}$ in this search is the best estimation of the insertion angle $\hat{\theta}$ and position $\hat{p}_p$ (lines 10-12). The estimated needle axes $\hat{\boldsymbol{v}}_n$ and $\hat{\boldsymbol{v}}_p$ can be calculated from the estimated insertion angle. In summary, the search is carried out as follows.

3) Filtering Estimated Angle Using Kalman Filter: A KF will be used to reduce variations in the estimated angle $\hat{\theta}$ between the images. The tip estimation step in Section II-B is sensitive to small deviations around the bright pixels representing the needle in the image. Therefore, a KF is implemented to have a smoother estimation of $\hat{\theta}$ in Cascade to the previous steps. Fig. 2 shows the block diagram describing the whole method.

Algorithm 2 Estimating the Axis by Searching for It Over a Range of Possible Angles

```
 1: function [$\hat{p}_p$, $\hat{\theta}$]=ESTIMATEAXIS($\boldsymbol{I}$, $\hat{p}_n[k-1]$)
 2:     for $\theta_k \in [\theta_{start}, \theta_{end}]$ steps of 1° do
 3:         [$p_k$, $H_k$] $\leftarrow$ AXISSEARCHSTEP($\boldsymbol{I}$, $\theta_k$, $p_n$)
 4:     end for
 5:     $[-, k_{max}] \leftarrow \max_k H_k$
 6:     $\theta_{max} \leftarrow \theta_{k_{max}}$
 7:     for $\theta_k \in [\theta_{max} - 0.5°, \theta_{max} + 0.5°]$ steps of 0.1° do
 8:         [$p_k$, $H_k$] $\leftarrow$ AXISSEARCHSTEP($\boldsymbol{I}$, $\theta_k$, $p_n$)
 9:     end for
10:     $[-, k_{max}] \leftarrow \max_k H_k$
11:     $\hat{\theta} \leftarrow \theta_{k_{max}}$
12:     $\hat{p}_p \leftarrow p_{k_{max}}$
13: end function
```

Algorithm 3 Estimating the Tip

```
function [$\boldsymbol{h}$]=ESTIMATETIP($\boldsymbol{I}$, $\hat{p}_p$, $\hat{\theta}$)
    Define the ROI using $\hat{p}_p$ and $\hat{\theta}$
    Filter $\boldsymbol{I}$ using $\boldsymbol{K}^{SQ}$ and $\boldsymbol{K}^{G''}$
    $h(x) \leftarrow \sum_{i=1}^{5} w_i E_i(x)$        $\forall\, x \in ROI$
end function
```

B. Needle Tip Estimation

To estimate the needle tip position, we analyze five features defined along the needle axis $\hat{\boldsymbol{v}}_n$. These features are combined in a linear function and the needle tip position is located in correspondence of its maximum [10]. With respect to [10], our approach uses different features except for one, and a new method is implemented for choosing the tip position. This method improves the method proposed in [22].

The image filtering and feature calculation is shown as the "Tip Est" block in Fig. 2, and summarized in Algorithm 3. In Cascade of this block, there is the "KF/PF" block: the tip position is filtered by either a KF or a PF to improve accuracy and robustness by reducing noises and removing outliers.

1) Image Filters and Features: Five different features based on two different image filters were taken into account in [22]. Each feature $E_i(x)$ is calculated along the estimated needle axis $\hat{\boldsymbol{v}}_n$, where x is the position along the needle axis. A high value of these features indicates a high likelihood that x is the needle tip. These features are combined linearly in a function h

$$h(x) = \sum_{i=1}^{5} w_i E_i(x) \tag{6}$$

where w_i is the relative weight for the feature $E_i(x)$. Peaks of this function give a set of possible needle tip positions, the most likely one corresponds to the maximum of $h(x)$. $\boldsymbol{h}$ denotes a vector containing the values of $h(x)$ for a set of x values.

Since the needle in the US images is characterized by high gray-level intensities, a sudden drop is expected at the needle tip. A function that has a high value when such drop in intensity occurs is the derivative of the grayscale image I^{GS}

$$E_1(x) = \sum_{u \in \text{TR}} I^{\text{GS}}(u) - \sum_{v \in \text{LR}} I^{\text{GS}}(v) \tag{7}$$

where LR is the leading region of pixels and TR is the trailing region of pixels. This feature is a modified version of the feature used in [10] based only on the current image.

In order to take into account temporal information, the difference between the current and the previous image is calculated as

$$\boldsymbol{I}_k^{\Delta} = \boldsymbol{I}_k^{\text{GS}} - \boldsymbol{I}_{k-\delta}^{\text{GS}} \tag{8}$$

where k is the time step. We use the image δ time steps behind instead of just one to increase the position displacement of the needle in the two images. If the needle moves, there should be a region with high intensity where the needle has moved, and low intensity everywhere else. The needle tip should be located at the border of the high intensity region. Therefore, we also consider the derivative of difference image using (7) to define the second feature E_2. The difference image should ideally have a bright line where the needle has moved. This bright line has two ends giving two possible needle tip locations. Using (7), the real needle tip should be positive and the needle location in the image at $k-\delta$ should be negative. This assumes that the needle axis is in the same place in the two images, but the surrounding tissue may move. However, if the needle axis is shifted between the two images, the assumption is no longer valid and the feature will not necessarily represent the needle tip position.

The next two features (E_3 and E_4) also use (7), but with the images filtered by $\boldsymbol{K}^{\text{SQ}}$ and $\boldsymbol{K}^{G''}$. The first image filter uses a squared kernel $\boldsymbol{K}^{\text{SQ}}$ given by

$$K^{\text{SQ}}(x, y) = \begin{cases} 1 & \text{if } -\frac{1}{2} < y_s < \frac{1}{2} \\ -1 & \text{else} \end{cases} \tag{9}$$

where y_s is a scaled version of y, ranging from -1 to 1. The size s_x in the $\bar{\boldsymbol{v}}_n$ direction should be twice the size of s_y in the $\bar{\boldsymbol{v}}_p$ direction. The size s_y should be twice of the echo created when the waves reflect back from the needle. This depends both on the needle and medium which the waves propagate in. The second filter uses a kernel based on the second derivative of a Gaussian $\boldsymbol{K}^{G''}$ given by

$$K^{G''}(x, y) = -2\left(\frac{s_{\text{scale}}^2 y^2}{\sigma^2} - 1\right) e^{-\frac{s_{\text{scale}}^2 y^2}{\sigma^2}} \tag{10}$$

where $y \in [-((s_y - 1)/2), ((s_y - 1)/2)]$, s_y is the kernel size in the $\bar{\boldsymbol{v}}_p$ direction, σ is the standard deviation, and s_{scale} is a scaling parameter. These filters enhance the visibility of the needle in the image by enhancing the tubular shape of the needle.

In the case of $\boldsymbol{K}^{\text{SQ}}$, an offset is observed between the peak of the feature and the position of the needle tip. This offset is due to the long shape of the kernel and can be removed by shifting the image of $-s_n$ pixels along $\hat{\boldsymbol{v}}_n$.

The last feature E_5 is similar to E_3, but with a shift in the $\hat{\boldsymbol{v}}_p$ direction. On grayscale US images, there is a bright spot produced by the bevel tip of the needle. This distorts the negative boundary below the needle in the $\boldsymbol{K}^{\text{SQ}}$ image. The region goes from negative to zero around the location of the needle tip. By shifting the image n pixels up and then use (7), we can detect this, where n is determined based on the needle and US probe used.

The last improvement with respect to [22] is the implementation of statistical filters: the tip position is estimated by a KF and a PF. The implementation of these two filters will be described in Section II.C-II.D and the corresponding performance improvements compared in Section IV.

C. Filtering the Tip Estimation Using a Kalman Filter

A KF will be used here to estimate the needle tip position. This step is shown as the "KF/PF" block in Fig. 2, as both a KF and a PF could be implemented. To estimate the needle tip, we use a second-order kinematic model. The first component of the state vector $\boldsymbol{x}_k$ is the needle tip position along $\hat{\boldsymbol{v}}_n$ denoted by x_k, and the second component is the velocity along the same axis denoted by v_k. This yields to the state-space model

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{w}_k \tag{11}$$

where T_s is the sampling time and $\boldsymbol{w}_k$ is the driving white Gaussian noise. The measurement equation for the tip position will be shown in Section II.C.1-II.C.2. If a robot is used to insert the needle, also the velocity is known or, at least, can be accurately estimated. Therefore, we have two different measurement updates for the KF, depending on the availability or not of velocity measurements.

1) Finding Needle Tip Position Candidates: To find possible needle tip position candidates, we observe that a high value of $h(x)$ in (6) indicates a high probability for a needle tip [10]. The goal is to find all the peaks of this function and select the right one. This is done by finding all the points that are a local maximum within a M pixel radius[3]

$$x \in \mathcal{J} \quad \text{if } h(x) \geq h(x+i) \quad \forall i \in [-M \ldots M]. \tag{12}$$

This procedure returns a set of plausible points $\mathcal{J}$: these points will undergo a probabilistic selection as described in the following.

2) Probabilistic Selection of Measurement: The points in $\mathcal{J}$ represent the possible needle tip positions and we need to select the most likely one to be used in the KF through the measurement equation. For this selection, we will use the Bayesian theory to find the probability of each point given the predicted needle tip position. This can be achieved by using Bayes' rule

$$P(j_i|\bar{x}_k) = \frac{P(\bar{x}_k|j_i)P(j_i)}{P(\bar{x}_k)} \tag{13}$$

where $j_i \in \mathcal{J}$ and $\bar{x}_k$ is the predicted needle tip position from the KF. In order to find this probability, the values of the right side of the equations need to be found. Let $P(\bar{x}_k|j_i)$ be

[3] We set $M = 10$ in the experimental section.

a Gaussian distribution with mean $\bar{x}_k$ and standard deviation σ. The standard deviation is a design parameter that has to be chosen: the smaller its value, the lower the probability of the points that are far away from the prediction. The probability function can then be written as

$$P(\bar{x}_k|j_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(j_i-\bar{x}_k)^2}{2\sigma^2}} \tag{14}$$

whereas $P(\bar{x}_k)$ is approximated by

$$P(\bar{x}_k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\bar{x}_k^2}{2\sigma^2}}. \tag{15}$$

The prior probabilities are calculated using the $h(x)$ function

$$P(j_i) = \frac{h(j_i)}{\sum_{j_i \in \mathcal{J}} h(j_i)}. \tag{16}$$

Now, the probability of each point j_i given the prediction $\bar{x}_k$ can be computed. The point with the highest probability is selected to be the *new* measurement for the KF

$$[-, z_k] = \max_{j_i \in \mathcal{J}} P(j_i|\bar{x}_k). \tag{17}$$

The time-varying variance R_k for the measurement is updated according to

$$R_k = \frac{R_{\text{MIN}}}{P(j_i|\bar{x}_k)} \tag{18}$$

where R_{MIN} is the minimum variance possible.

A problem with this approach is how to initialize it. Since the initial position of the needle tip is unknown, it is not possible to calculate the probabilities. The simplest approach is to choose the initial position at the position the needle is expected to appear. Because the estimation is very uncertain in the beginning, the σ parameter is dynamically adjusted to avoid relying too much on the prediction. It begins with σ_{MAX} and is reduced by $\Delta\sigma$ at each step of the KF until it reaches σ_{MIN}.

3) Adding Robot Measurements: The previous KF uses only position information obtained from the US images. This makes the algorithm generic and it can be used during manual insertion of the needle. When a robot is inserting the needle, the velocity measurements can be added to the measurement equation as

$$\boldsymbol{y}_k = \begin{bmatrix} z_k \\ \dot{z}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{r}_k \tag{19}$$

where z_k is from (17), $\dot{z}_k$ is the measured velocity, and the measurement noise has a matrix variance $\boldsymbol{R}_k = \left[\begin{smallmatrix} R_k & 0 \\ 0 & R_v \end{smallmatrix}\right]$. The KF will be evaluated with and without the velocity measurements.

D. Filtering the Tip Estimation Using a Particle Filter

The PF is a probabilistic approach, where the underlying probability density functions (pdfs) are sampled. The PF used here is the generic PF [23] with systematic resampling and using the prior as the importance sampling function. The PF will be compared with the KF introduced in Section II-C. It is worth highlighting that the PF works also when the distributions are not Gaussian, but multimodal. This step is shown as the "KF/PF" block in Fig. 2. When using KF, one point is selected as the outcome of the tip estimation phase, whereas when using the PF, the whole $h(x)$ function is taken into account in the measurement equation.

The system model is the same as for the KF (11), but the measurement model is different. The $h(x)$ function is scaled into a pdf $h_{\text{pdf}}(x)$: $h_{\text{pdf}}(x)$ is positive or equal to zero for all x and its integral is equal to one, that is

$$h_{\text{pdf}}(x, \bar{x}) = \frac{h_{\text{mod}}(x, \bar{x})}{\sum_{\forall x} h_{\text{mod}}(x, \bar{x})} \tag{20}$$

with

$$h_{\text{mod}}(x, \bar{x}) = (h(x) - \min_x h(x)) \frac{1}{\sigma_s\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x-\bar{x})^2}{\sigma_s^2}} \tag{21}$$

where $\bar{x}$ is the current estimate of the tip position, and σ_s is a parameter to be tuned.

Linear interpolation is used to find the values of $h_{\text{pdf}}(x, \bar{x})$

$$P(y_k|x_k) = h_{\text{pdf}}(x_k, y_k) \tag{22}$$

where y_k is the maximum of $h(x)$, and x_k is the position particle.

1) Adding Robot Measurements: As with the KF, the robot velocity is added as a measurement for the PF. The noise in the velocity measurement is modeled as Gaussian white noise. Adding the Gaussian white noise to the prior yields

$$P(\boldsymbol{y}_k|\boldsymbol{x}_k) = h_{\text{pdf}}(x_k, y_k) \frac{1}{\sqrt{2R_v\pi}} e^{-\frac{1}{2}\frac{(v_k-\dot{z}_k)^2}{R_v}} \tag{23}$$

where x_k and v_k are the position and velocity particles, y_k and $\dot{z}_k$ are the measurements, and R_v is the variance of the Gaussian white noise.

E. Training the Linear Function

To tune the weights in (6), we use data from a separate experiment. A genetic algorithm [24] is used to optimize the weights. The genetic algorithm is designed to minimize the following function:

$$\min_{w} \sum_i |\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i| \tag{24}$$

where i denotes the ith frame of the training set, $\boldsymbol{p}_i$ is the ground truth position vector of the needle tip measured by the optical tracking system, and $\hat{\boldsymbol{p}}_i$ is the position vector estimate from $h(x)$.

F. Summary

The two needle estimation methods are summarized in Algorithms 4 and 5 using KF and PF, respectively. Fig. 2 shows the different parts of the method. The "Axis est" block is described in Algorithm 2 and line 2 in Algorithms 4 and 5. The "KF" block filtering $\hat{\theta}$ is lines 16 and 12 in Algorithms 4 and 5, respectively. The "Tip est" block is given in Algorithm 3 and line 4 in Algorithms 4 and 5. The "KF/PF" block is given in lines 4–15 of Algorithm 4 and lines 4–11 of Algorithm 5.

Algorithm 4 Estimating the Tip Using Kalman Filter

```
function $[\hat{\boldsymbol{p}}, \hat{\theta}]$=ESTIMATENEEDLEKF($\boldsymbol{I}$, $\hat{p}_n[k-1]$)
    $[\hat{p}_p, \hat{\theta}] \leftarrow$ ESTIMATEAXIS($\boldsymbol{I}$, $\hat{p}_n[k-1]$)
    $\boldsymbol{h} \leftarrow$ ESTIMATETIP($\boldsymbol{I}$, $\hat{p}_p$, $\hat{\theta}$)
    Find the needle tip position candidates using (12)
    Calculate the Kalman filter prediction
    $\mathcal{J} \leftarrow \{x|h(x) \geq h(x+i)\ \forall\ i \in [-M \dots M]\}$
    $[-, z_k] \leftarrow \max_{j_i \in \mathcal{J}} P(j_i|\bar{x}_k)$
    $R_k \leftarrow \frac{R_{MIN}}{P(j_i|\bar{x}_k)}$
    if Robot measurements available then
        $\dot{z}_k \leftarrow$ robot velocity
        Update the Kalman filter update using (19)
    else
        Update the Kalman filter update using $z_k$ and $R_k$
    end if
    $[\hat{p}_n\ -]^T \leftarrow \boldsymbol{x}_k$
    Filter $\hat{\theta}$ to get $\hat{\theta}_{KF}$
end function
```

Algorithm 5 Estimating the Tip Using Particle Filter

```
function $[\hat{\boldsymbol{p}}, \hat{\theta}]$=ESTIMATENEEDLEPF($\boldsymbol{I}$, $\hat{p}_n[k-1]$)
    $[\hat{p}_p, \hat{\theta}] \leftarrow$ ESTIMATEAXIS($\boldsymbol{I}$, $\hat{p}_n[k-1]$)
    $\boldsymbol{h} \leftarrow$ ESTIMATETIP($\boldsymbol{I}$, $\hat{p}_p$, $\hat{\theta}$)
    if Robot measurements available then
        Update the Particle filter using the prior:
        $P(\boldsymbol{y}_k|\boldsymbol{x}_k) \leftarrow h_{pdf}(x_k, y_k)\frac{1}{\sqrt{2R_v\pi}}e^{-\frac{1}{2}\frac{(v_k-\dot{z}_k)^2}{R_v}}$
    else
        Update the Particle filter using the prior:
        $P(y_k|x_k) \leftarrow h_{pdf}(x_k, y_k)$
    end if
    $\hat{p}_n \leftarrow x_k$
    Filter $\hat{\theta}$ to get $\hat{\theta}_{KF}$
end function
```

III. Experimental Setup and Methods

In order to train and validate the proposed method, two needle insertion experiments were conducted using a robot. The robot inserted the needle into a beef meat phantom, while the insertion was captured by a US video stream. Two different samples of beef meat were used for the training and validation experiments. The robot was a UR5 from Universal Robots, Denmark. The biopsy needle was an Angiotech ProMag 14 GA × 10-cm biopsy needle from Medical Device Technologies Inc., Florida, USA. The needle was sanded using a sanding paper for better visibility. The US Machine was a System Five from GE Vingmed, Norway. The experimental setup is shown in Fig. 3.

The needle was mounted on the robot end-effector, and the US probe was positioned on the meat phantom using a rigid arm. Both the US probe and the needle were tracked using an infrared (IR) tracking system. The IR tracking system was an OptiTrack system from Natural Point Inc., USA, using 12 FLEX three cameras and the Arena software. The tracking system was used to ensure that the needle was in the image plane of the US probe and that the insertion had the desired angle.

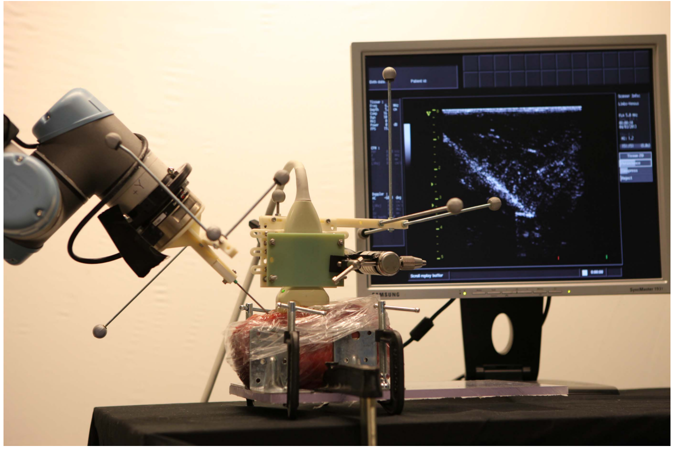


Fig. 3. Experimental setup consisting of a robot, a phantom, and a US machine.

TABLE I
Data Set for Training and Verification

	1	2
Number of needle insertion experiments	6	18
Number of US Frames	513	1390
Mean starting needle length in image [mm]	11.7	11.0
Mean stop needle length in image [mm]	46.3	42.3

The needle is imaged using an FLA 5-MHz probe and using the Limbs-Venous program on the US machine. The depth was set to 5 cm, frequency to 8 MHz, power to 0 dB, compression to 16, dynamic range to 10, and rejection to 0. The frame rate was 15.9 according to the US machine. The US image stream was acquired using a video VGA2Ethernet frame grabber from Epiphan,[4] California, USA. The frame grabber acquired 20 frames/s.

The first experiment was used to train the method and the second to validate the accuracy of the method. In both experiments, the insertion angle was either 35° or 45°, measured by the tracking system. The insertion speeds were 5, 10, or 15 mm/s. This results in six different combinations of parameters. For the training experiment, one insertion for each parameter combination was conducted, while the validation experiment had three insertions for each parameter combination. The characteristics of the two experiments are shown in Table I.

The robot inserting the needle was controlled in Cartesian velocity in the end-effector frame. The proprietary controller of the robot accepts joint velocities, and a Cartesian controller using the inverse Jacobian of the manipulator was implemented to control the robot [25]. The user selected the duration of the insertions.

The needle axis and needle tip position were manually extracted from the US images by a physician to evaluate the accuracy of our method. The physician marked the needle axis and tip on the acquired US images, starting with the image when approximately 10 mm of the needle is visible. To speed up this time-consuming manual task, the physician

[4] https://www.epiphan.com/wp-content/uploads/2014/07/epiphan-vga2ethernet-brochure.pdf

TABLE II
KF PARAMETERS

		KF	KF$_v$
σ_{MAX}	[px]	300	300
σ_{MIN}	[px]	200	200
$\Delta\sigma$	[px]	5	5
$\hat{\boldsymbol{P}}_0$	[px^2]	$\begin{bmatrix} 20^2 & 0 \\ 0 & 5^2 \end{bmatrix}$	$\begin{bmatrix} 20^2 & 0 \\ 0 & 5^2 \end{bmatrix}$
$\boldsymbol{Q}$	[px^2]	$\begin{bmatrix} \frac{\Delta T^2}{2}0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$	$\begin{bmatrix} \frac{\Delta T^2}{2}0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$
R_{MIN}	[px^2]	30^2	30^2
$\boldsymbol{R}$	[px^2]	R_k	$\begin{bmatrix} R_k & 0 \\ 0 & 1^2 \end{bmatrix}$

only evaluates every third image, and linear interpolation is used to find the needle axis and the needle tip position for the unevaluated images.

The algorithms were executed on a computer running Ubuntu 12.04 and Xenomai 2.6.0 real-time framework. The computer had an Intel Core i7 M620 CPU running at 2.67 GHz and 8 GB of RAM. The algorithm ran at the highest priority level and paging was turned OFF.

A. Training the Method

When training the tip estimation method, only a selection of the samples from the training experiment was used. Only samples where the estimated axis was less than 3 mm from the true axis (e.g., $|\hat{p}_p - p_p| < 3$ mm) were used in training the weights in (6). This is to ensure that the training data actually represent the needle tip.

B. Implementation and Parameter Choices

In our implementation of the axis estimation algorithm, we use 12 rays that are 97 pixels long and have an averaging region of 9 pixels on each side of the ray. When using feedback (i.e., dynamic ROI defined from the current tip position estimate), there is a 20 pixels spacing between the rays. Then, the minimum number of rays is 3 and the maximum ray spacing is 45 pixels. The top corner of the ROI (see Fig. 1) is set to 40 px to the left and 20 px downward of the top-right corner of the US image. We use bilinear interpolation to calculate the pixel values in the ROI.

We are working with 14 gauge biopsy needle with a nominal diameter of 2.108 mm. Considering a scale factor of 0.10 mm/pixel for the acquired US images, the needle width s_n is approximately 20 pixels. The kernel $K^{G''}$ should be positive in the range $\pm((s_n)/2)$. This is satisfied with $s_y = 89$. The K^{SQ} kernel size was set to 40×20. The parameter δ used for calculating the image I_k^{Δ} is set to 10.

The variances of the system and measurement noises in the KF used for estimating the insertion angle $\hat{\theta}$ are $Q_\theta = 1$ and $R_\theta = 4^2$, respectively, whereas the initial covariance $P(0) = 10^2$.

The KF parameters for the tip estimation are listed in Table II. The process noise of the KF is simplified by having the position noise magnitude as a function of the velocity noise magnitude. By doing so, we simplify the tuning process by reducing the tuning problem to find the ratio between the process and measurement noise [26].

The PF uses 200 particles. The standard deviation of the Gaussian distribution, which is used to get the initial particles, is $((\Delta T)/(\sqrt{2}))50$ px for position. The standard deviation for the velocity in the state equation is 50 px/s when no velocity measurements are available and 5 px/s when such measurements are available. The standard deviation σ_s in (21) is 150 when no velocity measurements are available and 75 when such measurements are available. The process noise has a standard deviation of $((\Delta T)/(\sqrt{2}))100$ px for the position, 100 px/s for velocity when no velocity measurements are available, and 5 px/s when velocity measurements are available. The variance R_v was set to 1 px^2.

C. Benchmarking Method

The proposed method in this paper will be compared with the method in [8]. They have made a real-time needle estimation method using Hough transform on segmented US images, modifying the transform to achieve real-time performance by utilizing a coarse-fine search. We have implemented the method according to their description with some small exceptions. The methods find a set of possible needle axes. We have simplified the selection of the correct axis. The method originally searches all possible insertion angles. We have limited this to ±5° of the expected insertion angle. This is the same condition we use for our method, and this means that the comparison is fair. We select the axis that has the highest Hough transform value.

We extract the values from the segmented US image along the needle axis. This data are processed using a morphological close operation, with a structuring element of size 21, to close the gaps along the needle axis. Then, the tip is assumed to be at the first gap along the needle axis.

IV. EXPERIMENTAL RESULTS

The goal of the proposed method is twofold: 1) accurately estimate the needle orientation and tip position and 2) perform all the computation in real time. We will start by showing the needle orientation and position errors and later the execution times of the different methods.

A. Experimental Results of Estimation Error

To evaluate the performance of the algorithms, the same error metrics for needle axis and tip error proposed in [10] are used plus an additional one for the angular error, as in [22]. The first error metric is the *axis estimation error* $e_\theta = \theta - \hat{\theta}$, given by the difference between the rotation obtained by the tracking system θ and the estimated rotation $\hat{\theta}$. This error metric is not used in [10]: they resort to the arccos to find the angle difference between the true needle axis vector and the estimated needle axis vector. This is the same as taking the absolute value of the above error measure, i.e., $|e_\theta|$.

The second error measure is the *tip position error*, given by $||\boldsymbol{e}|| = ||\boldsymbol{p} - \hat{\boldsymbol{p}}||$, where $\boldsymbol{p}$ is the position of the needle tip in the image plane, and $\hat{\boldsymbol{p}} = [\hat{p}_n \ \hat{p}_p]^T$ is the estimated

TABLE III

NEEDLE AXIS AND TIP ERROR (MEAN ± STANDARD DEVIATION [95th/2.5th–97.5th PERCENTILE])

	#	Method	e_θ [°]		$\lvert e_\theta \rvert$ [°]		$\lvert\lvert \boldsymbol{e} \rvert\rvert$ [mm]		e_n [mm]		e_p [mm]	
Manual	1	DF	$-2.66^1 \pm 3.58$	[-8.99–2.41]	3.33 ± 2.96	8.67	13.75 ± 10.09	33.68	$9.32^1 \pm 13.43$	[-17.59–37.44]	$-0.98^1 \pm 4.77$	[-15.76–6.95]
	2	OM	0.06 ± 2.20	[-5.11–3.49]	1.67 ± 1.44	4.44	3.55 ± 6.23	17.21	-0.13 ± 7.14	[-21.32–10.33]	-0.04 ± 0.63	[-0.88–1.23]
	3	OM+F	-0.02 ± 2.16	[-6.07–3.36]	1.51 ± 1.54	4.48	3.13 ± 5.99	19.33	-0.42 ± 6.70	[-22.74–8.07]	-0.03 ± 0.79	[-0.76–1.29]
	4	KF	-0.06 ± 1.95	[-4.71–2.79]	1.48 ± 1.26	3.92	1.81 ± 1.61	5.42	$1.16^1 \pm 2.03$	[-1.99–6.20]	-0.04 ± 0.63	[-0.88–1.23]
	5	KF+F	-0.02 ± 1.73	[-5.04–2.38]	1.21 ± 1.23	3.75	1.68 ± 1.72	5.54	$1.16^1 \pm 2.01$	[-1.53–6.22]	0.00 ± 0.63	[-0.63–1.14]
	6	PF †	-0.06 ± 1.95	[-4.71–2.79]	1.48 ± 1.26	3.92	2.68 ± 2.92	8.81	$2.33^1 \pm 3.15$	[-1.34–10.32]	-0.04 ± 0.63	[-0.88–1.23]
	7	PF+F †	$0.10^2 \pm 1.74$	[-4.96–2.66]	1.23 ± 1.23	3.89	2.64 ± 3.03	8.66	$2.43^1 \pm 3.14$	[-0.92–10.55]	$-0.03^3 \pm 0.65$	[-0.69–1.19]
Robot	8	KF_v	-0.06 ± 1.95	[-4.71–2.79]	1.48 ± 1.26	3.92	0.94 ± 0.74	2.12	0.04 ± 1.01	[-1.62–2.11]	-0.04 ± 0.63	[-0.88–1.23]
	9	KF_v+F	-0.23 ± 1.63	[-4.48–2.16]	1.17 ± 1.16	3.57	0.82 ± 0.76	2.17	0.02 ± 0.92	[-1.39–2.22]	0.05 ± 0.63	[-0.56–1.01]
	10	PF_v †	-0.06 ± 1.95	[-4.71–2.79]	1.48 ± 1.26	3.92	1.69 ± 2.26	7.66	$0.94^1 \pm 2.58$	[-3.17–8.80]	-0.04 ± 0.63	[-0.88–1.23]
	11	PF_v+F†	$-0.12^4 \pm 1.70$	[-5.09–2.35]	1.20 ± 1.22	3.51	1.81 ± 2.32	7.21	$0.75^5 \pm 2.76$	[-4.84–8.32]	0.01 ± 0.66	[-0.69–1.14]
		OptiTrack	$-2.89^1 \pm 7.14$	[-18.21–14.94]	6.26 ± 4.49	15.20	3.89 ± 2.89	9.33	$-1.24^1 \pm 1.61$	[-4.61–1.12]	$1.46^1 \pm 4.15$	[-8.80–8.63]

[1] Significantly different from zero at p=0.001

[2] Significantly different from zero at p=0.001 for 4 of 50 runs

[3] Significantly different from zero at p=0.001 for 3 of 50 runs

[4] Significantly different from zero at p=0.001 for 12 of 50 runs

[5] Significantly different from zero at p=0.001 for 47 of 50 runs

needle tip position. The third error measure is the *error on the needle axis* $\hat{\boldsymbol{v}}_n$ defined as $e_n = p_n - \hat{p}_n$, where p_n is the position of the needle tip measured by the tracking system, projected onto $\hat{\boldsymbol{v}}_n$, and $\hat{p}_n$ is the estimated needle tip position along $\hat{\boldsymbol{v}}_n$. The last error measure compares the *perpendicular positions* p_p and $\hat{p}_p$, $e_p = p_p - \hat{p}_p$.

1) Estimation Error: Table III shows the error results. The 11 methods are divided in two groups. The first is named "Manual," as these methods do not require a robot and can, therefore, also be used for manual insertion. The second group is named "Robot," as these methods require a robot or some other means of measuring the insertion velocity in order to work. The mean of the error measures e_θ, e_n, and e_p has been tested for significant difference from a zero mean distribution using the Student's t-test at a significance level of 0.001.

The first method in Table III was proposed in [8], and is used as a reference to evaluate the performance of the new methods, and is denoted "DF" for the authors Ding and Fenster. The second method was proposed in [22], and forms the basis of the methods proposed in this paper. It is denoted "OM" for original method. The next method (OM+F) uses an adaptive ROI, where the needle tip position is given as feedback to the needle axis estimation method, as explained in Section II-A: this feature is denoted by "+F" in Tables III–VI for all methods where it is implemented. The next two methods (KF and KF+F) use the KF introduced in Section II-C. Then, the next two methods use the PF introduced in Section II-D, denoted by PF and PF + F.

Since the PF samples the probability distribution randomly, it has been run 50 times and the result in Table III is the average of these runs. This is indicated with the † mark. However, it is worth highlighting that there were only minor differences among the runs.

The last four methods (from 8 to 11) use the velocity measurements from the robot in addition to the US image to estimate the tip position, as explained in Sections II-C3 and II-D1. This is denoted by the subscript v. The last entry in Table III is "OptiTrack," which is the error of the tracking system.

The $\hat{\boldsymbol{v}}_n$ axis points from the needle entry point toward the needle tip; thus, a negative/positive value of e_n means that the needle is inserted shorter/longer than estimated.

TABLE IV

COMPARING METHODS FOR SIGNIFICANT DIFFERENCE IN THE STANDARD DEVIATION (σ) OF THE ERROR MEASURE e_θ. THE p-VALUE IS SET TO 0.001, "-" DENOTES NO SIGNIFICANT DIFFERENCE, "↓" DENOTES SIGNIFICANTLY LOWER DIFFERENCE, AND "↑" DENOTES SIGNIFICANTLY LARGER DIFFERENCE

			Methods											
#	Method	σ [°]	1	2	3	4	5	6	7	8	9	10	11	OT
1	DF	3.58		↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↓
2	OM	2.20	↓		-	↑	↑	↑	↑	↑	↑	↑	↑	↓
3	OM+F	2.16	↓	-		↑	↑	↑	↑	↑	↑	↑	↑	↓
4	KF	1.95	↓	↓	↓		↑	-	↑	-	↑	-	↑	↓
5	KF+F	1.73	↓	↓	↓	↓		↓	-	↓	-	↓	-	↓
6	PF†	1.95	↓	↓	↓	-	↑		↑	-	↑	-	↑	↓
7	PF+F†	1.74	↓	↓	↓	↓	-	↓		↓	-	↓	-	↓
8	KF_v	1.95	↓	↓	↓	-	↑	-	↑		↑	-	↑	↓
9	KF_v+F	1.63	↓	↓	↓	↓	-	↓	-	↓		↓	-	↓
10	PF_v †	1.95	↓	↓	↓	-	↑	-	↑	-	↑		↑	↓
11	PF_v+F†	1.70	↓	↓	↓	↓	-	↓	-	↓	-	↓		↓
OT	OptiTrack	7.14	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	

TABLE V

COMPARING METHODS FOR SIGNIFICANT DIFFERENCE IN THE STANDARD DEVIATION (σ) OF THE ERROR MEASURE e_n. THE p-VALUE IS SET TO 0.001, "-" DENOTES NO SIGNIFICANT DIFFERENCE, "↓" DENOTES SIGNIFICANTLY LOWER DIFFERENCE, AND "↑" DENOTES SIGNIFICANTLY LARGER DIFFERENCE

			Methods											
#	Method	σ [mm]	1	2	3	4	5	6	7	8	9	10	11	OT
1	DF	13.43		↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
2	OM	7.14	↓		-	↑	↑	↑	↑	↑	↑	↑	↑	↑
3	OM+F	6.70	↓	-		↑	↑	↑	↑	↑	↑	↑	↑	↑
4	KF	2.03	↓	↓	↓		-	↓	↓	↑	↑	↓	↓	↑
5	KF+F	2.01	↓	↓	↓	-		↓	↓	↑	↑	↓	↓	↑
6	PF†	3.15	↓	↓	↓	↑	↑		-	↑	↑	↑	↑	↑
7	PF+F†	3.14	↓	↓	↓	↑	↑	-		↑	↑	↑	↑	↑
8	KF_v	1.01	↓	↓	↓	↓	↓	↓	↓		↑	↓	↓	↓
9	KF_v+F	0.92	↓	↓	↓	↓	↓	↓	↓	↓		↓	↓	↓
10	PF_v †	2.58	↓	↓	↓	↑	↑	↓	↓	↑	↑		-	↑
11	PF_v+F†	2.76	↓	↓	↓	↑	↑	↓	↓	↑	↑	-		↑
OT	OptiTrack	1.61	↓	↓	↓	↓	↓	↓	↓	↑	↑	↓	↓	

The 95th percentile of the error is also presented in Table III for the errors that are always positive ($\lvert e_\theta \rvert$ and $\lvert\lvert \boldsymbol{e} \rvert\rvert$). For the values e_θ, e_n, and e_p, a range is given, where the lower value is the 2.5th percentile and the upper value is the 97.5th percentile.

TABLE VI
EXECUTION TIME OF THE ALGORITHMS

Method	Mean [ms]	Max [ms]
DF	3.55	4.61
OM	26.13	27.35
OM+F	25.73	26.20
KF	26.13	27.19
KF+F	25.71	27.69
PF	26.09	27.16
PF+F	25.68	26.19
KF$_v$	26.33	27.25
KF$_v$+F	25.91	28.23
PF$_v$	26.31	27.42
PF$_v$+F	25.89	26.30

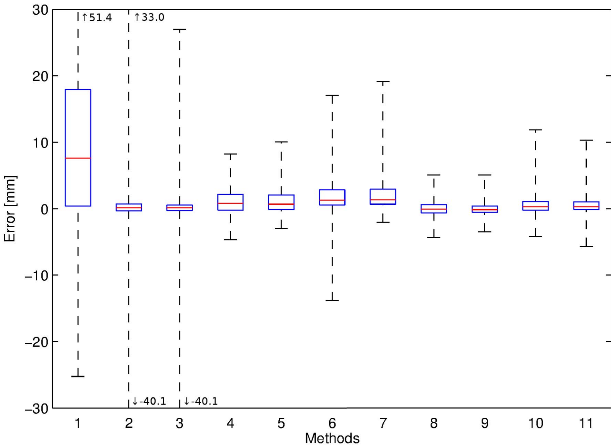


Fig. 4. Box-and-whisker plot of the tip position error (e_n) using both data sets. The sample size is 1390 for all methods, and the whiskers are the minimum and maximum samples. For the PF, all of the 50 runs are displayed.

Fig. 4 shows the box-and-whisker plot of the error e_n for the 11 methods under evaluation. The sample size is 1390 for all methods, and the whiskers are the minimum and maximum samples. For the PF, all of the 50 runs are displayed.

To compare the different methods, we focus on the variance of the estimation errors, because the variance is strictly related to the accuracy. The standard deviation of e_θ and e_n has been compared for significant differences at a 0.001 level in Tables IV and V. These two error measures were chosen, as they represent the main performance measures for the two steps of the method, and they also have a distribution close to the normal one. The F-test [27] has been used to compare all the methods against each other: "-" denotes no significant difference in the standard deviation of the error, whereas "↓" and "↑" denote significantly lower and larger differences, respectively.

The probability distribution of the error e_n is shown in Fig. 5 for the methods OM, KF+F, and KF$_v$+F. The distribution is estimated using a Parzen-Window with a bandwidth of 0.1 px. Fig. 5 does not show the whole distribution. The error is in the range from -5 to 5 mm for 75.2%, 95%, and 99.9% of the samples for each method, respectively.

Fig. 6 shows the error e_θ as a function of the needle visible in the US image for the KF+F method. The other methods have similar characteristics. Gray dots represent the error for one individual image, black solid line is the mean, and black dashed lines are the mean ± the standard deviation. The mean and the standard deviation of the error e_θ are computed on the bins of size 5 mm of the visible needle length, where the first bin contains all the measurements below 15 mm, while the last one contains values above 45 mm.

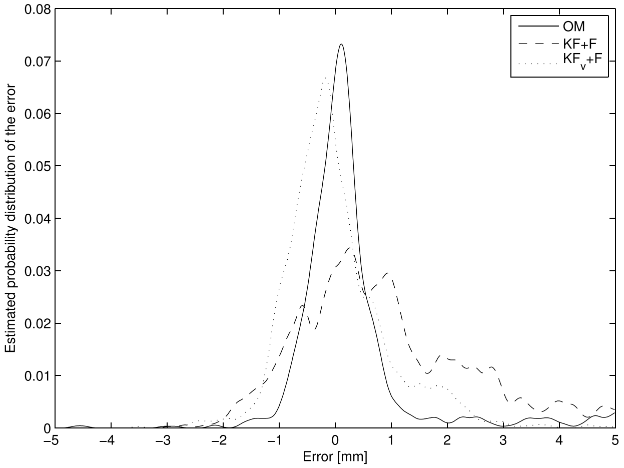


Fig. 5. Error distribution of the error e_n for the three methods OM, KF+F, and KF$_v$+F. The error is in the range from -5 to 5 mm for 75.2%, 95%, and 99.9% of the samples for each method, respectively.

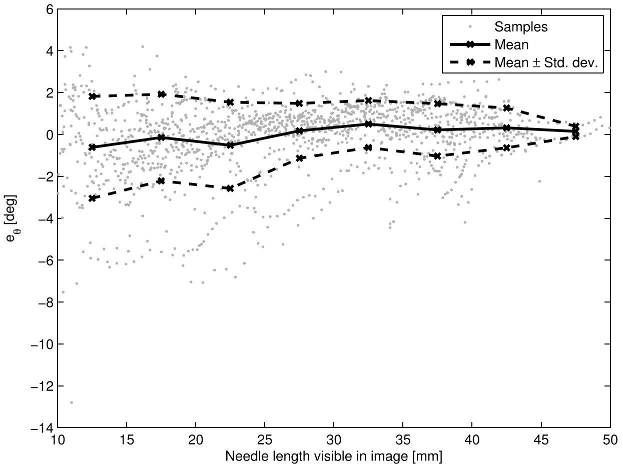


Fig. 6. Error e_θ as a function of how much of the needle is visible in the image for the KF+F method. Gray dots: error for one individual image. Black solid line: mean. Black dashed line: mean ± standard deviation.

B. Experimental Results of Execution Time

The time used for each US frame is reported in Table VI. The standard deviation of the execution time is <0.13 ms for all the methods.

V. Discussion

The proposed methods, estimation error and real-time properties, are discussed in Section V-A, starting with the estimation error.

A. Estimation Error

As described in Section II, the method consists of two steps: the first step estimates the axis of the needle and the second step estimates the tip position. The first step is evaluated using the error in the insertion angle (e_θ and $|e_\theta|$) and the position error perpendicular to the needle (e_p).

The second step is evaluated by the position error along the needle axis (e_n). The last error measure ($\|\boldsymbol{e}\|$) is the overall accuracy of the two steps.

The benchmarking method from [8] has large errors compared with the other methods. The main reason is that it relies on the segmentation of the needle in the US images using thresholding. Although it used an adaptive scheme for thresholding, the needle is often too faint compared with the structures in the meat to be fully segmented. This makes the method not particularly reliable.

Comparing the errors of OM and the results in [22], we note that the lower errors are reported there. In the previous experiments, the method was started when the needle was quite far into the phantom, making the estimation easier, since there are more data to be used during the axis estimation: in this way, the axis estimation gave stable results. In the experiments reported in this paper, it could fail for some images because the algorithm is switched on when 10 mm of the needle is visible in the image, which affects the overall accuracy. In addition, a guide was used in [22] when inserting the needle, which makes it easier to get the needle in the US plane. In this experiment, the robot holding the needle was positioned in correspondence to the US plane based on the measurements of the tracking system, which is less robust than using a guide. If the method is used as a visual aid for a physician, the physician him/herself will ensure that the needle is in the plane. Either by using a guide or manually adjusting the probe, if a robot inserts the needle, a registration between the needle and US probe is needed. This could be achieved by using an IR tracking system, as in this paper, or by other means.

In this paper, all the insertions are done by a robot. If the insertions are done manually, the insertion speed will vary more. This might cause larger estimation errors, as the KF and PF are tuned to the insertions made by a robot.

The errors of the OptiTrack system are given in the results, for comparison with the methods. In general, the insertion angle error and the position error perpendicular to the needle are much larger for the tracking system than all the methods. However, the position error along the needle axis is small, and the standard deviation of the error is lower than 9 out of the 11 methods. When it comes to absolute position error of the needle tip, the OptiTrack system has errors comparable with the PF-based methods, while the KF-based methods have lower errors regardless of using insertion velocity or not. This shows that the methods presented in this paper actually increase the accuracy of both the orientation and position of the needle compared with using an optical tracking system. In addition, the issue of camera occlusion, which lowers the optical tracking system's accuracy, is avoided.

As previously stated, we assume that the needle is rigid. This choice was made because it enables easier validation and method comparison. The axis estimation method in this paper is a modification of the method in [9], where bending needles are detected.

Our method could be extended to work with bending needles, since in real conditions, this is a very common event. The main modification would be the estimation of curved needle trajectory and the computation of features along this trajectory, instead of a straight line. This is left as future work.

1) Axis Estimation Error: Starting with the first step of the method, Table III shows the error of the insertion angle e_θ. All methods except DF, PF+F, and PF$_v$+F have a mean error that is not significantly different from zero. When looking at the standard deviation in Table IV, it is worth remarking that using a KF on $\hat{\theta}$ (as used in methods 4–11) yields a significantly lower standard deviation. Among these methods, using feedback (+F) decreases significantly the standard deviation. It is also worth mentioning that the precision in the angle estimation depends on how much of the needle is visible in the image. This is shown in Fig. 6 where the standard deviation of the error decreases when more of the needle is visible. Pearson's rank correlation coefficient between the standard deviation of the error and the length of the needle visible in the image is -0.94 with a p-value of less than 0.0005, when using the eight standard deviations from Fig. 6.

In terms of the perpendicular position error e_p, the DF method has high errors and a mean that is significantly different from zero. Only 62.2% of the samples have an error smaller than 1 mm. For all the other methods, more than 95% of the samples have an error of less than 1 mm. When comparing the methods using feedback (+F) with the methods not using feedback, the percentiles of the error (e_p) is decreased for the methods using feedback.

2) Tip Estimation Error Without Velocity Measurements: On the second step of the method, we will distinguish between methods with and without velocity measurements from the robot. We start with the methods not using velocity measurements, i.e., the methods 1–7 marked "Manual" in Table III. The error along the needle axis (e_n) shows the performance of the second step: the percentiles of this measure is shown in Fig. 4. The error $||\boldsymbol{e}||$ shows the total position error of the method. As this error measure is far from the Normal distribution, no significance testing is done on it.

The feedback in the OMdecreases the mean and standard deviation of the error $||\boldsymbol{e}||$, but there is no significant difference in precision between the OM method with and without feedback, as shown in Table V. The OM method is very accurate and precise when it is successful. This is shown in Fig. 4 where the 25th and 75th percentiles of the error are among the lowest of all the evaluated methods, and in Fig. 5 where the error distribution is very narrow. The main drawback of the method is that there are many samples with a large error. Almost 25% of the samples have an error e_n of more than 5 mm. The methods proposed in this paper are made to deal with this problem.

The use of the KF improves the tracking performance compared with the original method. The 95th percentile of the total position error is decreased by 68.5% and 67.8% using the KF and KF+F method, respectively. The spread in the error is also reduced. This is shown by the reduction in the standard deviation of e_n (71.9% and 71.6%) and of $||\boldsymbol{e}||$ (74.2% and 72.6%). This is a major reduction in error compared with the original method. The KF and KF+F methods have significantly smaller standard deviation of e_n compared with all methods not using velocity measurements (Methods 1–7). The reduction in error makes the result of the method more reliable to be used in a clinical setting. The total position error with the method using feedback has smaller mean but higher standard deviation. From the other point of view, the position error of the two methods is almost equal: the KF+F method is better, because it estimates the angle more accurately.

The KF methods have biased estimate of the tip position along the needle, as the mean of e_n is significantly different from zero. The error distribution of e_n for the KF+F method is shown in Fig. 5. The distribution has its maximum at 0.26 mm and has a positive skew. It is the skewness of the distribution that creates the bias in the mean of e_n.

The error e_n has a larger reduction in the 2.5th percentile than the 97.5th percentile. This means that the needle is more often estimated to be inserted shorter than it really is. This could come from the tracking model in the KF, which weights more the predicted value of the tip. This causes the KF estimate to lag behind the true needle position in some instances.

The improvement when using PF is not as evident as using the KF. All error measures of the PF methods have lower values than the original method, except the mean of the error e_n. However, the error measures are not as low as when using the KF. It is worth noting that the standard deviation of both the total position error $||\boldsymbol{e}||$ and the tip position error e_n has a major reduction compared with original method; also, the standard deviation of e_n is significantly smaller than the original method.

3) Tip Estimation Error With Velocity Measurements: In this section, we analyze the methods using velocity measurements from the robot (Methods 8–11). Compared with KF and KF+F, the 95th percentile for the total position error is reduced by 87.7% and 87.4% for the methods KF_v and KF_v+F, respectively. The standard deviation of the total position error is reduced by 88.1% and 87.8% and the standard deviation of e_n is reduced by 85.6% and 87.1%. This is a remarkable improvement. The KF_v+F method has significantly lower standard deviation of e_n than all the other methods, making it the most precise method when looking at tip position along the needle axis. One should note that by introducing velocity measurements, the bias in e_n is removed.

The PF using velocity measurements does not perform as well as the KF. This indicates that the measurement method based on the KF is better than the one based on the PF, probably because the unimodal approximation of the signals' distribution is correct.

B. Execution Time

The second performance metric of the methods is the execution time. To use the methods in real-time applications, the computation needs to be finished before the next frame is received from the US machine. Table VI shows that the mean execution time is almost equal for all the methods, except DF which has much lower execution time. The methods using feedback are faster than the methods without feedback. Feedback allows to reduce the number of rays during the axis estimation, because the ROI is dynamically adapted according to how much of the needle is visible in the image.

All methods presented in this paper have a finite predefined maximum number of internal iterations; thus, there is no jitter that could significantly affect the computational time. This is clearly shown by the very low standard deviation of the execution time, which is lower than 0.13 ms for all the methods. The variation in the maximum execution times in Table VI is, therefore, most likely caused by hardware or other processes through priority inheritance. Using the absolute maximum execution time in Table VI yields a maximum frame rate of 35.4 frames/s, which is larger than the frame rate that most commercial US machines can guarantee.

VI. Conclusion

This paper proposed and compared different algorithms that allow to improve the needle tip tracking precision in the US images with respect to other solutions in the literature. The precision in the estimation of the needle insertion angle has been increased significantly. The improvement comes from: 1) the introduction of a dynamical adaptation of the ROI to find the needle axis and 2) filtering the insertion angle using statistical estimators. The method with overall best performance was the algorithm based on a KF with feedback and velocity measurements (i.e., robotic-aided insertion), where the 95th percentile of the position error was reduced by 87.4% and the standard deviation by more than 87.8% compared with the previous methods. In addition, the standard deviation of the tip position error along the needle was significantly smaller than all other evaluated methods.

When no velocity measurements were available (i.e., manual insertion), the method using KF with feedback had the best performance as well. The method using KF without feedback had very similar tip position error, but the standard deviation of the insertion angle error was significantly lower when using feedback. We, therefore, conclude that using feedback yields better performance. The 95th percentile of the position error was reduced by more than 67.8%, the standard deviation by more than 72.6%. In addition, the standard deviation of the tip position error along the needle was significantly lower than for all other methods not using velocity measurements.

All the methods using KF had better accuracy than the OptiTrack optical tracking system. This indicates that using image information yields more reliable tracking accuracy than using an external tracking system. It also removes the need for having an expensive tracking system when tracking the needle.

As a future work, we plan to improve the algorithms in order to track steering needles and to design a control architecture to optimal positioning the US probe in robotic needle insertion.

References

[1] S. N. Goldberg, G. S. Gazelle, and P. R. Mueller, "Thermal ablation therapy for focal malignancy: A unified approach to underlying principles, techniques, and diagnostic imaging guidance," *Amer. J. Roentgenol.*, vol. 174, no. 2, pp. 323–331, Feb. 2000.

[2] M. Ahmed, C. L. Brace, F. T. Lee, Jr., and S. N. Goldberg, "Principles of and advances in percutaneous ablation," *Radiology*, vol. 258, no. 2, pp. 351–369, 2011.

[3] S. N. Goldberg *et al.*, "Image-guided tumor ablation: Standardization of terminology and reporting criteria," *Radiology*, vol. 235, no. 3, pp. 728–739, Jun. 2005.

[4] M. Bonfè *et al.*, "Towards automated surgical robotics: A requirements engineering approach," in *Proc. 4th IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Jun. 2012, pp. 56–61.

[5] R. Muradore *et al.*, "Development of a cognitive robotic system for simple surgical tasks," *Int. J. Adv. Robot. Syst.*, vol. 12, pp. 1–20, Apr. 2015.

[6] R. Pollock *et al.*, "Prospects in percutaneous ablative targeting: Comparison of a computer-assisted navigation system and the acubot robotic system," *J. Endourol.*, vol. 24, no. 8, pp. 1269–1272, 2010.

[7] E. M. Boctor, M. A. Choti, E. C. Burdette, and R. J. Webster, III, "Three-dimensional ultrasound-guided robotic needle placement: An experimental evaluation," *Int. J. Med. Robot. Comput. Assist. Surgery*, vol. 4, no. 2, pp. 180–191, 2008.

[8] M. Ding and A. Fenster, "A real-time biopsy needle segmentation technique using Hough transform," *Med. Phys.*, vol. 30, no. 8, p. 2222, 2003.

[9] S. H. Okazawa, R. Ebrahimi, J. Chuang, R. N. Rohling, and S. E. Salcudean, "Methods for segmenting curved needles in ultrasound images," *Med. Image Anal.*, vol. 10, no. 3, pp. 330–342, 2006.

[10] D. W. Cool, L. Gardi, C. Romagnoli, M. Saikaly, J. I. Izawa, and A. Fenster, "Temporal-based needle segmentation algorithm for transrectal ultrasound prostate biopsy procedures," *Med. Phys.*, vol. 37, no. 4, p. 1660, 2010.

[11] M. Kaya and O. Bebek, "Needle localization using Gabor filtering in 2D ultrasound images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 4881–4886.

[12] M. Kaya and O. Bebek, "Gabor filter based localization of needles in ultrasound guided robotic interventions," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2014, pp. 112–117.

[13] M. Kaya, E. Senel, A. Ahmad, O. Orhan, and O. Bebek, "Real-time needle tip localization in 2D ultrasound images for robotic biopsies," in *Proc. IEEE Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 47–52.

[14] Z. Wei, L. Gardi, D. B. Downey, and A. Fenster, "Oblique needle segmentation and tracking for 3D trus guided prostate brachytherapy," *Med. Phys.*, vol. 32, no. 9, p. 2928, 2005.

[15] A. Ayvaci, P. Yan, S. Xu, S. Soatto, and J. Kruecker, "Biopsy needle detection in transrectal ultrasound," *Comput. Med. Imag. Graph.*, vol. 35, nos. 7–8, pp. 653–659, 2011.

[16] Y. Zhao, A. Bernard, C. Cachard, and H. Liebgott, "Biopsy needle localization and tracking using ROI-RK method," *Abstract Appl. Anal.*, vol. 2014, Aug. 2014, Art. no. 973147.

[17] H. R. S. Neshat and R. V. Patel, "Real-time parametric curved needle segmentation in 3D ultrasound images," in *Proc. 2nd IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Oct. 2008, pp. 670–675.

[18] G. J. Vrooijink, M. Abayazid, and S. Misra, "Real-time three-dimensional flexible needle tracking using two-dimensional ultrasound," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1688–1693.

[19] P. Chatelain, A. Krupa, and M. Marchal, "Real-time needle detection and tracking using a visually servoed 3D ultrasound probe," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1676–1681.

[20] A. Harmat, R. N. Rohling, and S. E. Salcudean, "Needle tip localization using stylet vibration," *Ultrasound Med. Biol.*, vol. 32, no. 9, pp. 1339–1348, 2006.

[21] M. P. Fronheiser, S. F. Idriss, P. D. Wolf, and S. W. Smith, "Vibrating interventional device detection using real-time 3-D color Doppler," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 55, no. 6, pp. 1355–1362, Jun. 2008.

[22] K. Mathiassen, D. Dall'Alba, R. Muradore, P. Fiorini, and O. J. Elle, "Real-time biopsy needle tip estimation in 2D ultrasound images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 4363–4369.

[23] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[24] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Heidelberg, Germany: Springer-Verlag, 2003.

[25] K. Mathiassen, J. E. Fjellin, K. Glette, P. K. Hol, and O. J. Elle, "An ultrasound robotic system using the commercial robot UR5," *Frontiers Robot. AI*, vol. 3, no. 1, pp. 1–16, 2016.

[26] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1979.

[27] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*, 8th ed. Upper Saddle River, NJ, USA: Person Education, Inc., 2007.

Kim Mathiassen (S'13) received the master's degree in engineering cybernetics from the Norwegian University of Science and Technology, Trondheim, Norway. He is currently pursuing the Ph.D. degree with the Department of Informatics, University of Oslo, Oslo, Norway, and The Intervention Center, Oslo University Hospital, Oslo. The topic of the Ph.D. thesis is robotic ultrasound systems.

He was a Visiting Researcher with the Altair Robotics Laboratory, University of Verona, Verona, Italy, in 2012. His current research interests include robotic ultrasound systems and needle estimation from ultrasound images.

Diego Dall'Alba received the master's degree (Hons.) in intelligent and multimedia systems and the Ph.D. degree in computer science from the University of Verona, Verona, Italy, in 2010 and 2014, respectively. His Ph.D. thesis was on advanced technologies for ultrasound guided percutaneous procedures.

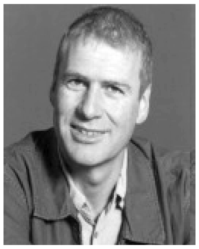
He has been a member of the Altair Robotics Laboratory, Verona, since 2008. He was a Research Associate with several European projects related to surgical robotics. In 2012, he was a Visiting Researcher with the University of British Columbia, Vancouver, BC, Canada. His current research interests include surgical robotics, computer assisted surgery, and ultrasound guided system.

Riccardo Muradore (S'99–M'04) received the Laurea degree in information engineering and the Ph.D. degree in electronic and information engineering from the University of Padua, Padua, Italy, in 1999 and 2003, respectively.

He was a Post-Doctoral Fellow with the Department of Chemical Engineering, University of Padua, from 2003 to 2005. Then, he spent three years with the European Southern Observatory, Munich, Germany, as a Control Engineer, where he was involved in adaptive optics systems. In 2008, he joined the Altair Robotics Laboratory, University of Verona, Verona, Italy. Since 2013, he has been an Assistant Professor. His current research interests include robust control, teleoperation, robotics, networked control systems, and adaptive optics.

Paolo Fiorini (F'09) received the Laurea degree in electronic engineering from the University of Padua, Padua, Italy, the M.S.E.E. degree from the University of California at Irvine, Irvine, CA, USA, and the Ph.D. degree in mechanical engineering from the University of California at Los Angeles, Los Angeles, CA, USA.

He was with the NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA, from 1985 to 2000, where he was involved in telerobotic and teleoperated systems for space exploration. From 2000 to 2009, he was an Associate Professor of Control Systems with the School of Science, University of Verona, Verona, Italy, where he founded the Altair Robotics Laboratory with his students. He is currently a Full Professor of Computer Science with the University of Verona. His current research interests include teleoperation for surgery, service, and exploration robotics funded by several European Projects.



Ole Jakob Elle received the master's degree from the Department of Production and Quality Engineering, Norwegian Institute of Technology, Trondheim, Norway, in 1990, and the Ph.D. degree in robotic surgery from the Norwegian University of Science and Technology, Trondheim, in 2004.

He was a Product Developer with Luxo ASA, Oslo, Norway, from 1996 to 1998, where he was involved in medical light systems. He joined The Intervention Center, Oslo University Hospital, Oslo, in 1998, to be a bridge between the Product Development Department, ComputerMotion Inc., Santa Bargbara, CA, USA, and the Cardiac Surgeons at the hospital within robotic surgery. He is currently the Head of the Section for Technology Research with The Intervention Center, Oslo University Hospital, and also an Adjunct Associate Professor with the Department of Informatics, University of Oslo, Oslo.