2014 5th IEEE RAS & EMBS International Conference on
Biomedical Robotics and Biomechatronics (BioRob)
August 12-15, 2014. São Paulo, Brazil

A Neural Network Approach for Flexible Needle Tracking in Ultrasound Images Using Kalman Filter

André A. Geraldes[1] and Thiago S. Rocha[2]

Abstract— **Percutaneous surgical procedures depend on the precise positioning of the needle tip for effectiveness. Although, robustly locating the tip of the needle still represents a challenge, specially when flexible needles are used. An imaging technique widely used for this task is 2D ultrasound, however the low signal/noise ratio makes it difficult to apply conventional image processing techniques. In this work, we propose an alternative method for detecting the needle in an ultrasound image and tracking it during a complete insertion. The proposed method combines a Multilayer Perceptron network with a Kalman filter estimator for improving robustness. In preliminary experimental characterization, we acquired ultrasound images for creating the data set and evaluated the performance of the tracker with a complete insertion video. However the tracking performance is still far from optimal, the obtained results suggests the neural network approach to be a feasible alternative to this problem.**

I. Introduction

A kind of surgery comprising a substantial fraction of minimally invasive procedures, being used in several medical situations such as anesthesia, biopsy, brachytherapy and neurosurgery [1], are the percutaneous surgical procedures, in which internal organs are accessed by inserting a needle through the patient's skin, avoiding the need of making large incisions. In all of these procedures, the accurate positioning of the needle tip is vital as a position error might spoil the treatment and even cause serious complications to the patient [2]. This situation becomes critical as the target position is located at hard access areas, where there are obstacles in the path of the needle, like blood vessels or nerves.

To solve this problem, flexible needles with tip shaped asymmetrically [3] have been proposed. This kind of needles are able to curve inside the patient's body, allowing the reaching of targets at locations only accessible by a curved-line trajectory. Thence, many works [4], [5], [6], [7] have employed flexible needles in the last decade. Though, since the needle doesn't behave as a rigid body, the use of this technique represents serious problems when it comes to accurately locating the needle tip, during the insertion. This makes the use of a medical imaging system unavoidable.

We can see that other works have made use of 2D [8], [9] and 3D [10] ultrasound images for tracking the needle tip position in these procedures. Among the main advantages of ultrasound imaging are its safety, low cost, non-invasiveness and its high time resolution, which allows its use for real-time applications. On the other hand, finding small objects in ultrasound images is typically hard using this kind of system, because the images contain a lot of speckle, resulting in low signal-to-noise ratio. For that reason, it is not always effective to apply traditional image processing techniques, such as edge detection or texture based algorithms.

An alternative method for identifying the flexible needle in the ultrasound images is to make use of pattern recognition tools such as neural networks. This kind of method is known for being capable of classifying complex structures, after training is performed over a large enough data set. Some examples of the application of these techniques comprise segmentation of an equine leg [11] and breast tumor detection [12] in ultrasound images. It is important to notice, however, that neural network based methods tend to lack in robustness, since the occurrence of false positives is common.

In order to improve the overall performance of the needle tracking task is to incorporate an stochastic filter, such as the well known Kalman filter, in order to decrease the impact of bad measurements. This technique uses a linear model of the system for predicting the needle movement and incorporate the measurements as corrections, taking into account the relative variance between prediction and sensing. The output of the filter is the best estimation of the needle position with an associated uncertainty factor.

A. Contributions and organization of the paper

In this work we propose an alternative method for tracking flexible needles in ultrasound images using a MLP as the main sensor element and a Kalman filter for improving robustness. The main goal of this work is to evaluate if the MLP-based approach is feasible for this kind of application and to compare it with the current state of art.

The rest of the paper is organized as follows. In section II we described the proposed method in details. In section III the experimental setup is described and the preliminary results are presented. A series of discussions are made in section IV and finally section V presents our conclusions and expectations of future work.

II. Tracking Algorithm

The problem of finding a flexible needle in ultrasound images is still considered to be a challenging one, as the needle is usually very thin (typical diameter in the order of hundreds of microns) and the ultrasound images are often

[1]André A. Geraldes is with the Laboratory of Robotics and Automation (LARA), University of Brasília, Brasília, Brasil. andregeraldes@lara.unb.br
[2]Thiago S. Rocha is with the Laboratory of Robotics and Automation (LARA), University of Brasília, Brasília, Brasil. rochasilvathiago@lara.unb.br

978-1-4799-3128-6/6/14/$31.00 ©2014 IEEE

noisy and contain a lot of speckle. However, tracking the needle tip, given its initial position, is much simpler, as it allows restricting the search region to a much smaller area. This difference becomes decisive, when a neural network approach is used, since this type of methods are usually powerful in solving pattern recognition tasks, but requires too much computational power for processing high dimensional inputs.

In the case of percutaneous procedures, the needle entry point is usually known beforehand, which permits adopting a local tracking approach, ignoring the problem of localizing the needle at an arbitrary location. This strategy, however, requires an iterative process of detecting the needle inside a region of interest (ROI) and moving the ROI accordingly, in order to keep the needle in range.

A. Method overview

The Figure 1 shows a top level representation of the tracking algorithm. As it can be seen, the main sensor element is the Multilayer Perceptron (MLP) network, which classifies the pixels of the ultrasound image in needle or background. The input to the MLP is provided by a ROI selector module, that extracts the pixels in a region close to the estimated needle tip location.

After the region classification finishes, the output of the MLP is a matrix indicating the likelihood of each pixel of the ROI belonging to the needle. These likelihood map is processed by a Tip Finder block, which measures the location of the needle tip and also evaluates the associated performance for this measurement, based in geometrical characteristics of the MLP output. Finally, the measured tip position is used to update the estimated tip position through a Kalman filter. This updated estimation is used to select the ROI in the next ultrasound frame.

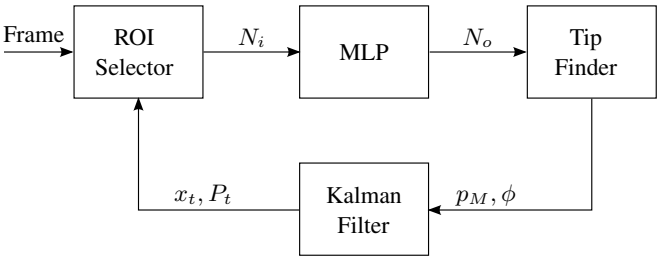


Fig. 1: Top level representation of the needle tracking system

B. ROI selection

The first step in a loop of the tracking algorithm is to select the ROI, from which the MLP input data will be extracted. This is done by receiving one frame from the ultrasound sequence and the current state estimation x_t, provided by the Kalman filter. In order to consider both static and dynamic information regarding the needle position, the last four frames are stored in a circular buffer. At every iteration a square window of width W is cropped from each of the four frames. All crop windows are centered at the current needle estimated position $p = (p_x, p_y)$. Next, the crop windows are resized to ZxZ matrices and concatenated in a single vector, producing an input block N_i of $4Z^2$ elements, which is provided to the MLP. By adjusting the parameters W and Z it is possible to adapt the width of the ROI, the computational effort required and the information loss due to image compression, given by the factor $1 - {}^{Z}/_{W}$.

Besides the estimated position, the ROI selector also analyzes the associated uncertainty P_t. When the position uncertainty is higher than a threshold τ_p, multiple instances of N_i are generated, cropping the stored frames not only at p, but also at the positions $p_{ij} = (p_x + ik, p_y + jk)$, where $i, j \epsilon \{-2, -1, 0, 1, 2\}$ and k is a scale factor, given in pixels. All generated N_i vectors are provided to the MLP, which increases the computational effort, making the tracking algorithm slower. However, this allows the tracker to expand its search region, until the position uncertainty decreases.

C. Needle detection

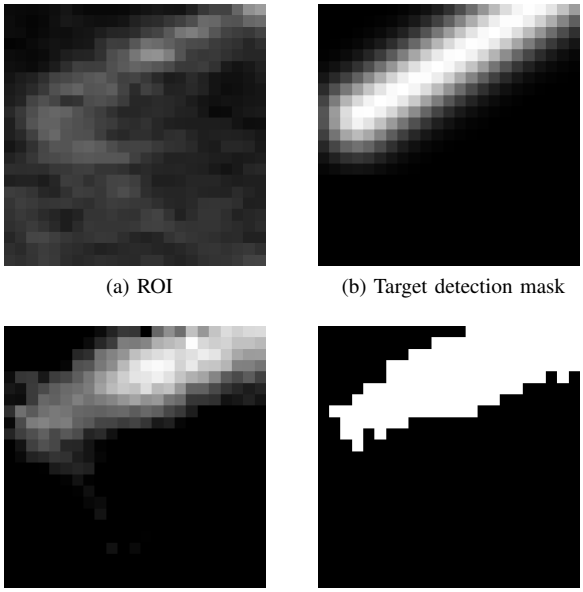
When the neural network receives an input N_i, it generates a ZxZ output matrix N_o, containing the likelihood of each pixel of the ROI belonging to the needle. For training this behavior it was necessary to manually produce these likelihood maps for each image in the training data set, generating target output images. In the produced targets, pixels corresponding to the needle were set to 1, background pixels were set to 0 and the pixels in the needle edges were set to intermediate values, so that the trained network would produce smoother output images.

The Figure 2 shows an example of the behavior of a trained network. As it can be seen, the output of the MLP contains a smooth transition between the needle and the background. This is desirable, because it reduces the occurrence of spurious false positive pixels, but requires thresholding the network output, before searching for the tip position. The threshold value is set as default, but it is adjusted based on the histogram of the network output, in order to ensure that the thresholded image will always contain at least one white region.

In a previous work [13], we have implemented a similar neural network, whose inner structure matches the one used here. That work contains a much more detailed description of the training process and the influence of the selection of the training parameters in the network performance.

D. Needle tip finder and performance evaluator

After the network output N_o is thresholded, the resulting binary image is analyzed in order to find the best estimation for the needle tip location. To increase the spatial resolution, the binary image is resized to an image of width W, even if that distorts the borders of the detected needle. The tip location process consists of segmenting the binary image and finding the largest region, which is considered to be the needle. Then, the center of mass and the region major axis are obtained. The needle tip location p_M is set as a



(a) ROI (b) Target detection mask

(c) Network output (d) Thresholded network output

Fig. 2: Needle detection performed by the MLP for a given ROI.

point over the needle major axis close to the extremity of the needle.

In addition to finding the needle tip location, the segmented region is used to evaluate the performance of the classification performed by the MLP. This is done by calculating a series of geometrical coefficients that contain information about the shape of the detected region. The selected coefficients were:

$$\eta_A = \frac{2A_{needle}}{A_{needle} + A_{other}} - 1 \tag{1}$$

$$\eta_C = \frac{A_{needle}}{A_{convex}} \tag{2}$$

$$\eta_M = 1 - \frac{m}{M} \tag{3}$$

$$\eta_\theta = 1 - \frac{\|\theta - \theta'\|}{\pi} \tag{4}$$

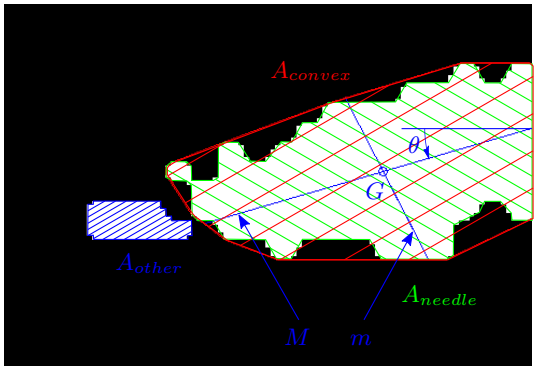


Fig. 3: Geometrical elements used for calculating the coefficients η_A, η_C, η_M and η_θ: the area of the needle (A_{needle}), of the smallest convex polygon containing the needle (A_{convex}) and of the other regions (A_{other}); the needle major (M) and minor (m) axes; and the needle orientation (θ).

The terms used for calculating these coefficients are graphically represented in Figure 3. Each of these parameters range from 0 to 1, where 1 corresponds to the ideal condition and 0 to the worst case scenario. The parameter η_A measures the fraction of the classified pixels that belong to the largest region. If the network output contains many regions, it strongly indicates that there are false positives among the detected pixels. Secondly, the parameter η_C measures the convexity of the needle region, by dividing the area of the needle region by the area of the smallest convex polygon containing the needle region. The parameter η_M analyzes the shape of the detected needle by computing how elongated the needle region is, provided by the ratio between the region's minor and major axes. Finally, the parameter η_θ measures the angular difference between the measured needle orientation and the previous orientation of the needle, estimated by the Kalman filter. Since the needle is a continuous body, this difference should not be too large, even if the needle curves inside the phantom.

Combining these four coefficients, we have created a performance measurement ϕ, given by

$$\phi = \eta_A^2 . \eta_C . \eta_M . \eta_\theta^{0.5} \tag{5}$$

, which also ranges from 0 to 1. The exponents associated to each coefficient where adjusted empirically, based on the degree of relevance of each coefficient to the overall performance measurement. The calculated ϕ is provided to the Kalman filter alongside the measured tip location p_M. In the case when the position uncertainty is high, and multiple N_i are provided to the MLP, the performance ϕ is used to select the best measurement and discard all the other ones.

E. Kalman filtering

In order to add robustness to the tracker algorithm, a first order Kalman filter was used to estimate the needle tip position and velocity. The state vector then corresponds to:

$$x_t = [\ p_x \quad p_y \quad \dot{p_x} \quad \dot{p_y}\ ]^T \tag{6}$$

The measurements vector Z_t is composed only by the two position coordinates of the needle tip. Then the dynamic model of the system can be described by

$$x_t = Fx_{t-1} + w_t \tag{7}$$

$$z_t = Hx_t + v_t \tag{8}$$

where

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

and considering the white noises w_t and v_t given by

$$w_t \sim N(0, Q) \tag{11}$$

$$v_t \sim N(0, R) \tag{12}$$

Since only the needle position is measured and the tracker does not actuate in the needle insertion system, the matrix H does not have any velocity term and also there is no control input matrix in the state update equation 7. The terms Q and R are the covariance matrices associated to the prediction and the measurement model, respectively. As the system is time invariant, Q was set to a diagonal constant matrix $Q = k_p I$, however R is changed at every iteration, to represent a sensor with variable variance. The equation for R was defined as $R = \sigma_t I$, with

$$\sigma_t = A + B(1 - \phi)^{\lambda} \tag{13}$$

The parameters A, B and λ need to be set in accordance with k_p to establish the correct relationship uncertainty between prediction and measurement.

The equations for updating the state using the Kalman filter can be given by:

$$\hat{x_t} = Fx_{t-1} \tag{14}$$

$$\hat{P_t} = FP_{t-1}F^T + Q \tag{15}$$

$$K_t = \hat{P}_t H^T (H\hat{P}_t H^T + R_t)^{-1} \tag{16}$$

$$x_t = \hat{x_t} + K_t(z_t - H\hat{x_t}) \tag{17}$$

$$P_t = (I - K_t H)\hat{P_t} \tag{18}$$

Equations 14 and 15 correspond to the prediction step, while equations 16 to 18 correspond to the correction step. At every measurement performed by the MLP, the Kalman filter is updated, by performing one prediction followed by one correction step (Eq. 14 to 18). The updated state (x_t) and variance (P_t) are used to determine the ROI for the next frame.

III. Experimental Characterization

In order to evaluate the proposed tracking system, experiments with artificial phantom were conducted. The goal of these experiments was to acquire a significant amount of ultrasound images for generating the MLP training data set and also for evaluating the complete tracking system after the MLP was trained. The acquired images were divided in two subsets, in order to ensure that the images used for testing the tracker were not used for training the MLP.

A. Experimental Setup

For acquiring the ultrasound images, a Nitinol (55%Ni-45%Ti metallic alloy) needle was manually inserted in a phantom tissue block. The phantom consisted of a modified recipe of ballistic gelatin, because its properties are similar to the human sot tissue and it provides good visualization with an ultrasound equipment. For increasing the heterogeneity of the produced phantom and result in a more realistic scenario, graphite powder and olives were added to the regular recipe of ballistic gelatin (made from gelatin and glycerin). Therefore the resulting images contained considerable speckle as well as obstacles for the needle insertion.

The equipment used to acquire the images was the SonixTOUCH from Ultrasound System Company, with a large bandwidth curved array transducer C5-2. The ultrasound probe was also operated manually, which means that the image plane was subject to small variations, rather than being strictly fixed. Since the whole insertion process was manual, the insertion speed of the needle was variable and unknown. This conditions diverge from the standard robot-assisted needle insertion setup, where there are more variables under control, being closer to the clinical case, where the physician perform the complete insertion manually.

B. Data set generation and MLP training

From these experiments, 7 usable videos were acquired, totalizing 1757 frames. Two of these videos (422 frames) were separated for testing and the other five (1335 frames) were used to train the MLP. In order to generate the data set, the needle tip position was marked in each frame of each video. After marking all frames, a set of input vector was generated, extracting one input N_i from each sequence of 4 consecutive frames.

The width of the ROI was 115 pixels and the width of the resized image was 23 pixels. Therefore, the input vector for the trained networks contained 2116 elements each. In order to increase the training data set and to make the MLP less orientation dependent, each set of 4 frames was rotated in an angle $\theta_R \epsilon \{0°, 90°, 180°, 270°\}$. Also, for increasing the variability of the training data set, a random offset was added to the crop position for each N_i, ranging from -43

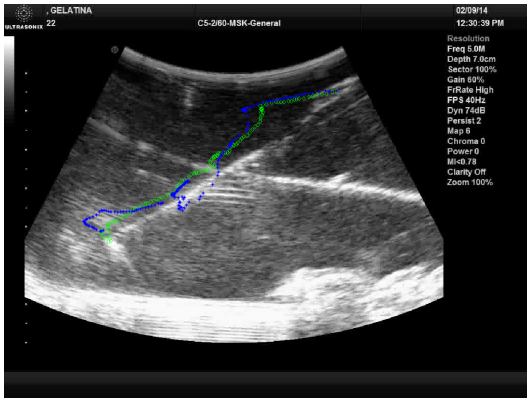


Fig. 4: Ground truth (green ◇) and estimated (blue ⋆) trajectory of the needle on the test video 1

to 43 pixels in x and y directions. This produced a total of 5340 training examples.

After generating all the input vectors, the target output images were generated, by marking the correct location of the needle in each selected ROI. The target images represent the likelihood of each pixel belonging to the needle, as described in section II-C and illustrated in Figure 2. After generating the training and test data sets, several networks were trained in order to find the training parameters, which resulted in the best performance. The best network, was a 4 layer MLP with 350 neurons in each hidden layer and 529 neurons in the output layer. Its performance in the test data set, evaluated by the Mean Squared Error was 0.0550. After thresholding, the average error was 0.0228, which corresponds to an average of 12 pixels wrongly classified in each 23x23 crop window.

C. Tracking Experiments

Once the MLP was trained, the described tracking algorithm was applied to the two test videos. In order to better evaluate the performance of the tracking system, the ground truth was generated by manually marking the needle tip position in each frame of the video.

The Figure 4 shows the ground truth and the estimated needle trajectory for the first test video. As it can be seen, the estimated trajectory diverged from the ground truth in certain moments, but the algorithm managed to track the needle until the end of the insertion. The estimation error, for each iteration, is plotted in the Figure 5. From this graph, it is possible to see that the final position estimation error was of 5.68 mm. The maximum estimation error during the complete insertion was of 14.92 mm

The Figure 6 presents the obtained trajectory for the second video. In this case, the algorithm lost the needle in the middle of the tracking process and did not manage to find it accurately, until the end of the video. The Figure 7 shows the estimation error for this test. As presented, until the frame 92, the error was below 4.1 mm, but after that the tracking system got lost, resulting in a final estimation error of 39.84 mm.

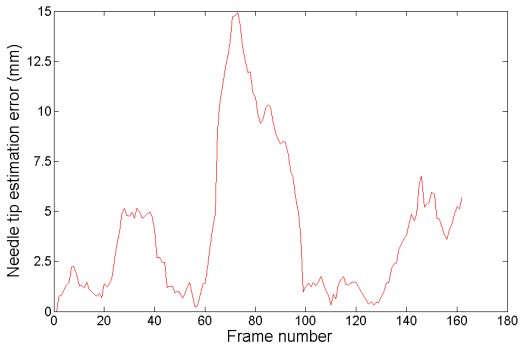


Fig. 5: Needle tip position error on the test video 1

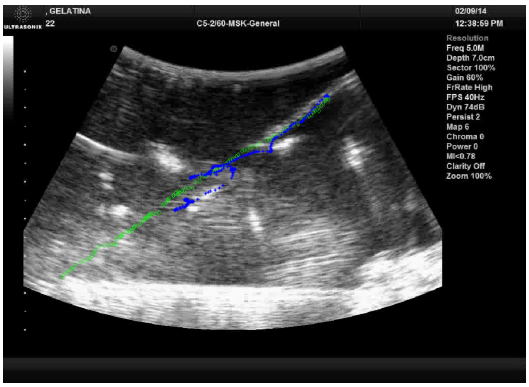


Fig. 6: Ground truth (green ◇) and estimated (blue ⋆) trajectory of the needle on the test video 2

IV. Discussions

Even though the obtained results for the tracking tests were not satisfactory, there are a series of discussions that deserve to be addressed. Firstly, it is necessary to evaluate the performance of the trained MLP. From a qualitatively analysis, which is adequately represented by Figure 2, it is possible to say that the results were truly satisfying. The network managed to detect the correct needle orientation and placement, even if the needle was not trivially distinguishable in the provided ROI. It is possible to assume that this result should be better than an untrained human subject. However, from a quantitatively analysis, it is seen that the average classification error of the MLP was of 12 pixels in 529, which corresponds to 300 pixels in a 115x115 ROI. If all the wrongly classified pixels were aligned in the same direction, this error could represent up to 40 mm of position estimation

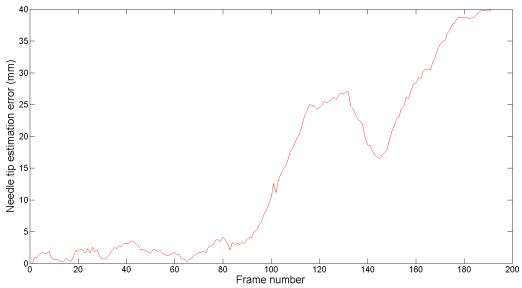


Fig. 7: Needle tip position error on the test video 2

error in a single measurement.

Despite of that, in the first test video, the maximum observed error was of 15 mm, as shows Figure 5. This indicates that the Kalman filtering in combination with the implemented performance evaluator is able to filter the worst measurements, producing a more reasonable result. It can also be seen from Figure 4 that the in two situations (frame 50 and frame 100), the estimated position largely deviated from the ground truth, but was able to correct itself. This is probably due to the implemented feature of increasing the search area in proportion to the state uncertainty.

In the second video, shown in Figure 6, the tracker lost the needle when encountering a large obstacle. This is obviously bad, but it must be noted that none of the videos used for training had a situation like that. This means that the poor performance of the MLP is a consequence of the bad quality and low variability of the acquired data. Even though, this experiments were enough to perform preliminary evaluations over the system, it becomes clear that it is necessary to acquire more and better ultrasound images to improve the performance of the MLP. One way to improve the quality of the ultrasound images would be to apply an anti-speckle or a band pass filter, in order to enhance the needle in the images.

It is also important to note that the initial complexity of the task was high. In the performed experiments, the insertion velocity is unknown and it varies during the insertion. We also assume that the initial velocity is unknown since we only know the tissue entry point beforehand. Besides that, the proposed tracking system is completely passive, which means the algorithm can not actuate the ultrasound probe if the needle leaves the image plane.

Having pointed all these difficulties, concerning the lack of control inputs and the limited data for training, it is possible to state that the obtained results were promising. Even if the performed experiments were still very preliminary, they suggest that if we had more data available or information about the insertion speed, the proposed MLP tracking approach could provide good results.

V. Conclusions

In this paper, we presented a neural network based flexible needle tracker for ultrasound images. The conceived method combines the classification power of the MLP, with stochastic filtering for discarding poor measurements and increase robustness. We have also collected images for generating a training data set and evaluated our system with new data in a complete needle tracking test. Even if the results are still not satisfactory, the preliminary experiments here described suggests this method could provide better results, if a larger and more diverse data set was available.

In future works, we intend to produce a new data set, increasing both the volume and the variability of our data. We would also like to incorporate noise reduction filters in our system in order to enhance the needle in the ultrasound images. Finally, we intend to evaluate the possibility of coupling the ultrasound probe to a robotic manipulator in order to be able to actuate in the system and guarantee the needle would always be in the image plane.

References

[1] G. Durkan, N. Sheikh, P. Johnson, A. Hildreth, and D. Greene, "Improving prostate cancer detection with an extended-core transrectal ultrasonography-guided prostate biopsy protocol," *BJU International*, vol. 89, no. 1, pp. 33–39, 2002.

[2] N. Abolhassani, R. V. Patel, and M. Moallem, "Needle insertion into soft tissue: A survey," *Medical Engineering & Physics*, vol. 29, no. 4, pp. 413–431, 2007.

[3] R. J. Webster, III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *Int. J. Rob. Res.*, vol. 25, no. 5-6, pp. 509–525, May 2006.

[4] K. Reed, V. Kallem, R. Alterovitz, K. Goldberg, A. Okamura, and N. Cowan, "Integrated planning and image-guided control for planar needle steering," in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS EMBS International Conference on*, oct. 2008, pp. 819 –824.

[5] T. Wedlick and A. Okamura, "Characterization of pre-curved needles for steering in tissue," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, sept. 2009, pp. 1200 –1203.

[6] M. Bernardes, B. Adorno, P. Poignet, and G. Borges, "Semi-automatic needle steering system with robotic manipulator," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 1595 –1600.

[7] J. Engh, G. Podnar, S. Khoo, and C. Riviere, "Flexible needle steering system for percutaneous access to deep zones of the brain," in *Bioengineering Conference, 2006. Proceedings of the IEEE 32nd Annual Northeast*, april 2006, pp. 103 – 104.

[8] G. Vrooijink, M. Abayazid, and S. Misra, "Real-time three-dimensional flexible needle tracking using two-dimensional ultrasound," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 1688–1693.

[9] Z. Neubach and M. Shoham, "Ultrasound-guided robot for flexible needle steering," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 4, pp. 799–805, 2010.

[10] Z. Y., C. C., and L. H., "Automatic needle detection and tracking in 3d ultrasound using an roi-based ransac and kalman method," *Ultrasound Imaging*, vol. 35, no. 4, pp. 283 – 306, Oct 2013.

[11] Q. Huang and R. Dony, "Neural network texture segmentation in equine leg ultrasound images," in *Electrical and Computer Engineering, 2004. Canadian Conference on*, vol. 3, May 2004, pp. 1269–1272 Vol.3.

[12] S.-F. Huang, Y.-C. Chen, and W. K. Moon, "Neural network analysis applied to tumor segmentation on 3d breast ultrasound images," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, May 2008, pp. 1303–1306.

[13] T. Rocha and A. Geraldes, "Flexible needles detection in ultrasound images using a multi-layer perceptron network," in *Biosignals and Biorobotics Conference (BRC), 2014 ISSNIP*, May 2014, pp. 1–6.