HAL
archives-ouvertes.fr

Nerve Contour tracking for Ultrasound-Guided Regional Anesthesia

Xavier Cortés, Donatello Conte, Pascal Makris

▶ To cite this version:

HAL Id: hal-02285803

https://hal.archives-ouvertes.fr/hal-02285803

Submitted on 13 Sep 2019

Nerve Contour tracking for Ultrasound-Guided Regional Anesthesia⋆

Xavier Cortés, Donatello Conte, and Pascal Makris

Université de Tours, Laboratoire d'Informatique Fondamentale et Appliquée de Tours
(LIFAT - EA 6300), 64 Avenue Jean Portalis, 37000 Tours, France
{xavier.cortes,donatello.conte,pascal.makris}@univ-tours.fr

Abstract. Ultrasound-Guided Regional Anesthesia is a technique to
provide regional anesthesia aided by ultrasound visualization of the re-
gion on which the anesthesia will be applied. A proper detection and
tracking of the nerve contour is necessary to decide where anesthesia
should be applied. If the needle is too far from the nerve contour, the
anesthesia could be ineffective, but if it touch the nerve could harm the
patient. In this paper we address a model to track nerve contours in ul-
trasonic videos to assist the doctors during Ultrasound-Guided Regional
Anesthesia procedures. The experimental results show that our model
performs good within an acceptable margin of error.

Keywords: Nerves · Contour Tracking · Ultrasound-Guided Regional
Anesthesia

1 Introduction

Regional anesthesia is an interesting alternative to general anesthesia during sur-
gical procedures. It reduces pain scores and postoperative complications. Typi-
cally, this technique has been performed by blind guidance of the needle to the
target nerve. However, this method of needle guidance increases the risk of block
failure, nerve trauma and local anesthetic toxicity [13]. To reduce these compli-
cations, the current trend is to use the Ultrasound-Guided Regional Anesthesia
(UGRA) technique. The UGRA technique has had an enormous impact on the
practice of regional anesthesia during the last years [10] becoming an emerging
and an innovative technique for anesthesia procedures.

The key problems with UGRA practice is the nerve localization and tracking
in ultrasound videos [15]. Some works have been published that address this
issue, in particular the detection of the nerve in a ultrasound image [12, 5, 4, 6].
Morevoer, nerve detection is typically a very time-consuming task, therefore is
not suitable in practice for UGRA, where real-time processing is needed. Besides,
to reduce the risk of trauma caused by touching nerves with the needle, the nerve

contour tracking has to be fast and precise. Therefore, classic object tracking algorithms that provide only bounding box of targets, are not sufficient to be used in practice.
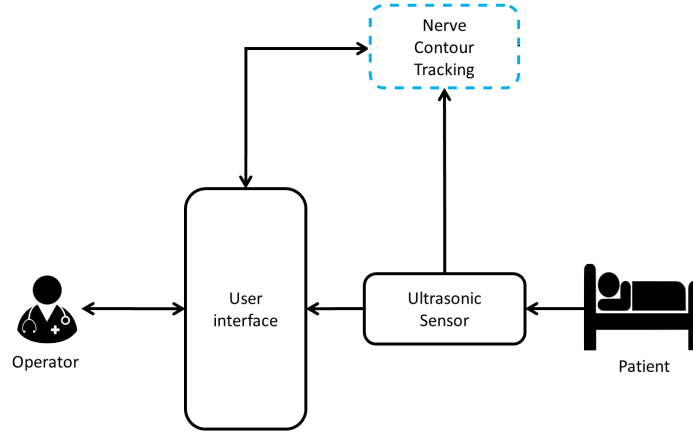

Fig. 1. General scheme of our framework.

For this reason, in this paper, we propose an attempt to solve the nerve contours tracking problem in ultrasound images aiming to assist the doctors during UGRA procedures. The proposed solution is real-time and quite precise. Even if each step of the approach are well-known image processing techniques, at the best of our knowledge, this is the first time that the whole framework is designed for nerve contour tracking in ultrasound images.

The rest of the paper is organized as follow : in Section 2, we describe the context within the tracking algorithm is situated; Section 3 presents the proposed method; in Section 4, we evaluate the method; finally, the paper ends with some conclusions and future perspectives (Section 5).

2   Framework description

The model presented in this paper aims to aid UGRA procedures by highlighting the contour of the nerve that we want to anesthetize.

The elements involved in the procedure addressed in this paper are: an ultrasonic transducer to scan the local region of the patients body where the anesthesia must be injected, the user interface that allows to the human operator to impose the initial bounding box and the nerve tracking module, which is the one that we propose here (see Figure 1).

An accurate detection of the contour is fundamental because the region that surrounds the nerves is the one where the anesthesia must be injected. If it is

too far from the nerve, it may be ineffective, while if it is inside the nerve, it could harm the patient.

The way in which this module works is related to a set of steps that are assisted by a human operator.

First, the ultrasonic transducer is manually placed on the region of the patients body on which the anesthesia has to be injected. Then, the human operator interacts with the system using a human-computer interface consisting of a screen, showing in real-time, the ultrasound video sequence that comes from the ultrasonic transducer. The operator, typically a doctor who has specific training to operate the system, imposes the initial bounding box where the nerve is located on the ultrasound sequence appearing in the video. The bounding box must contain the target nerve inside and exclude, as much as possible, all the elements that are not relevant, such as tendons, muscle fibers among others appearing in the video (Figure 2).
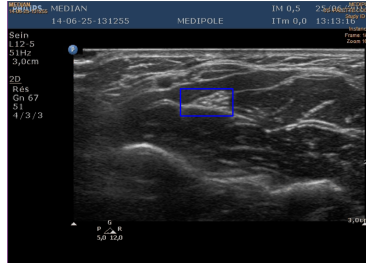

Fig. 2. An example of a bounding box (blue) limiting the region where the nerve contour is located.

3   Proposed model

To track the nerve contour we propose a procedure divided into three different steps that are processed iteratively for each frame of the video sequence after an initialization step in which the human operator imposes the bounding box. The first step consists in tracking the region inside the bounding box through the frames sequence. The second step is to reduce the noise of the region detected by the tracker in order to increase the quality of the contour detection and finally, the third step aims to detect the contour points inside the region bounded by the imposed box. In Figure 3 we show a general scheme of the model.

3.1   Bounding box tracking

The first step after the initialization of the bounding box by a human operator, is to track the region in the video where the nerve is located. This region is affected
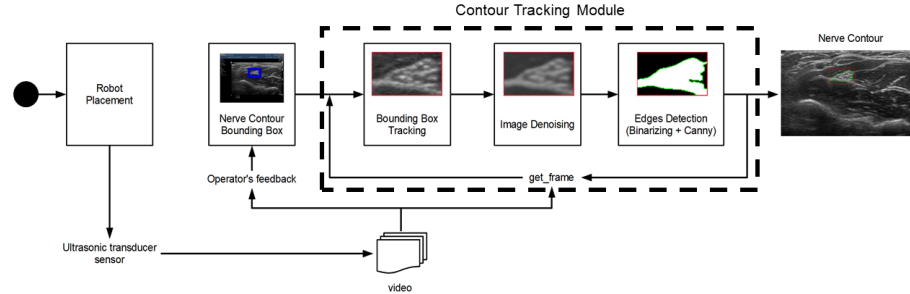
Fig. 3. General scheme of the contour tracking module.

by different deformations and translations because the shape and position of the nerves is not static in time.

The robustness of the tracking algorithm is crucial for the performance of the model. For this reason, we propose to evaluate our model using different state-of-the-art tracking algorithms in the experimental section.

3.2   Image denoising

Ultrasonic transducers may not be entirely accurate because of the intrinsic noise in the data captured by the sensors. Due to this, we propose to apply the *Fast Non-Local Means* (FNLM) [14] algorithm, before the detection of the nerves contour, in order to improve the performance of the method. FNLM is an algorithm that reduces the computational complexity of the original *Non-Local Means* (NLM) [1] algorithm, for smoothing the image. This complexity reduction is necessary if we want to apply the entire process in real-rime.

To give the basic principles and the complexity of NLM and FNLM, the denoiser algorithm applies, for each pixel of the image, a specific local filter built upon a neighborhood window. The computational complexity of the original NLM algorithm is $O(M^2 \cdot n^4)$ where M is the filter size and n is the number of pixels of the image. The fast version FNLM takes benefit of the Integral Image representation to reduce the number of computations needed for the filtering. Its complexity is $O(n^4)$, therefore it is independent from the filter size. Please, refer to [14] for more details.

We apply the denoiser algorithm only on the pixels within the bounding box provided by the previous step (tracking), because, by hypothesis, there is no nerve outside the box. This further reduces the computational cost of the processing.

3.3   Contour detection

The final step is to detect the points belonging to the nerve contour. To do this we propose a combination of thresholding and edges detection.

First, we apply a basic image thresholding [11] to binarize the region inside the bounding box, and next we apply the Canny Edges detection algorithm [3] to detect the borders on the binarized image. The detected edges are the contour points returned by the model.

4   Experimental evaluation

In this section we present the experimental results.

To evaluate the performance we compare the contours found by our model with the ground-truth contours that has been manually labeled by a human expert frame by frame on a grey-scale video of 659 frames captured by an ultrasonic transducer.

The metric used to evaluate the error between the contour found by the model and the ground-truth contour is the Mean Euclidean Distance (MED) defined as the mean distance in pixels between each point of the first contour with respect to the closest point of the second contour. Since this metric is not symmetric we evaluate the distance in both senses and calculate the mean.

We have compared the performance of our model using the following tracking algorithms (Section 3.1), that are know to be among the best in the scientific literature: *Online Multiple Instance Learning* (MIL) [2], *Median Flow* (MF) [16], *Kernel Correlation Filter* (KCF) [8, 9] and *Real-time Tracking via On-line Boosting* (RTB) [7].

In table 1 we show the mean, the maximum and the minimum MED results using different combinations of denoising and tracking algorithms. The performance are similar in terms of accuracy for MIL, KCF and RTB algorithms while for MF is significantly worse. Using the FNLM algorithm to eliminate noise (Section 3.2) we increase performance in all cases. By way of illustration in Figure 4, we show some ultrasound images and the tracked contours in order to show the behavior of our model with different configurations.

Table 1. Comparative table of results with different tracking algorithms. Best results highlighted in bold.

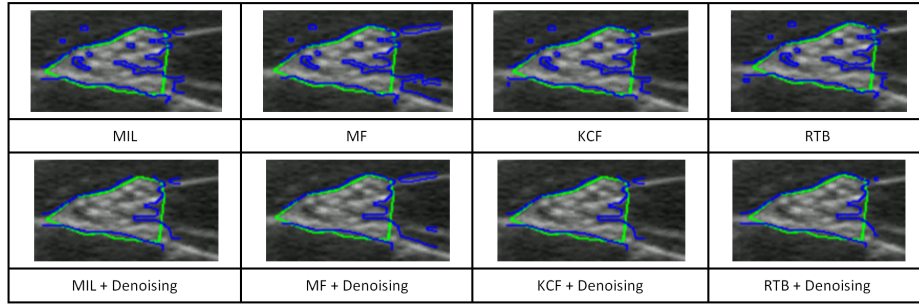| Tracking algorithm	Mean	Max	Min
MIL	4.30	6.12	3.41
MIL + Denoising	3.95	6.10	2.14
MF	6.09	11.06	3.32
MF + Denoising	5.76	11.27	**2.13**
KCF	4.20	6.36	3.23
KCF + Denoising	**3.73**	6.15	**2.13**
RTB	4.18	6.04	3.21
RTB + Denoising	3.75	**6.01**	2.15

Fig. 4. Nerve contour examples using different configurations. Green: ground-truth contour. Blue: contour found by the algorithm.

The second requirement that our model must satisfy is to be able to operate in real time or close to it. For this reason, in Figure 5, we show frame rate comparison for different tracking algorithms with and without denoising application. We have executed our experiments using a Python interpreter running on Windows 10, with an Intel i7 processor at 2.6 GHz and 16 GB of RAM. The vertical bars represent the average number of Frames Per Second (FPS) that each configuration can process in our experiments.
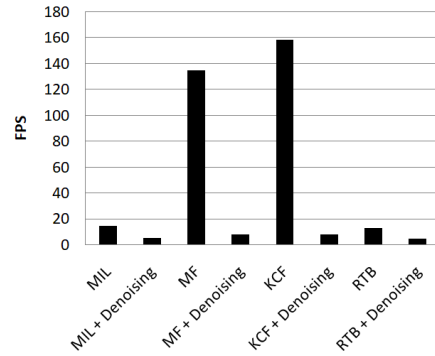

Fig. 5. Average number of Frames Per Second using different configurations.

On one hand, we observe that applying the FNLM in order to eliminate the noise increases significantly the runtime required to process each frame. On the other hand we see that the fastest algorithms are the MF and the KFC, while the MIL and the RTB have worse performance. The configurations that are able to process all the frames in real time assuming videos at 25 FPS in our hardware configuration are the MF and KCF when the noise is not removed using the FNLM algorithm (Section 3.2).

5   Conclusions and future work

The work presented in this paper is part of a bigger project aiming to provide anesthesia to medical patients in a framework supervised and operated by a human expert. In this paper we have presented the module referred to tracking of nerves contours.

Due to the design specifications of the project we have two main requirements to accomplish. The first one is to minimize the error as much as possible since this is a critical task that may result in injury to the patient or make anesthetic ineffective if the contour detection is not accurate. The second main requirement is to be able to track the contour in real time.

On one hand, the preliminary experimental results are very hopeful in terms of accuracy using different tracking algorithms. On the other hand, the runtime experiments show that when we remove the noise appearing in the frame we are far to be able to run the model in real-time using our hardware configuration.

As future work we propose to create a complete database with several videos and its ground-truth contour correspondences in order to exhaustively validate the model providing to the community a new benchmark to compare contour tracking algorithms. We plan also to define a new contour tracking algorithm, based on spatio-temporal continuity, in order to increase accuracy and to decrease runtimes.

References

1. Antoni Buades, Bartomeu Coll, J.M.M.: A review of image denoising algorithms, with a new one. In: Multiscale Modeling & Simulation. vol. 4, pp. 490–530 (2005)
2. Boris Babenko, Ming-Hsuan Yang, S.J.B.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, CVPR. pp. 983–990. IEEE (2009)
3. Canny, J.: A computational approach to edge detection. In: Pattern Analysis and Machine Intelligence. vol. 8, pp. 679–698. IEEE (1986)
4. Hadjerci, O., Hafiane, A., Conte, D., Markis, P., P.Vieyres, Delbos., A.: Ultrasound median nerve localization by classification based on despeckle filtering and feature selection. In: 2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015. pp. 4155–4159 (2015)
5. Hadjerci, O., Hafiane, A., Markis, P., Conte, D., Vieyres, P., Delbos., A.: Nerve detection in ultrasound images using median gabor binary pattern. In: Image Analysis and Recognition Lecture Notes in Computer Science. vol. II, pp. 803–806 (2014)
6. Hadjerci, O., Hafiane, A., Vieyres, P., Conte, D., Makris, P., Delbos, A.: On-line learning dynamic models for nerve detection in ultrasound videos. In: Image Processing (ICIP), 2016 IEEE International Conference on. pp. 131–135. IEEE (2016)
7. Helmut Grabner, M.G., Bischof, H.: Real-time tracking via on-line boosting. In: British Machine Vision Conference, BMVC. vol. 1 (2006)
8. J. F. Henriques, R. Caseiro, P.M., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision, ECCV (2012)

9. M. Danelljan, F.S. Khan, M.F., van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Computer Vision and Pattern Recognition, CVPR. pp. 1090–1097. IEEE (2014)

10. Marhofer, P., Willschke, H., Kettner, S.: Current concepts and future trends in ultrasound-guided regional anesthesia. Current Opinion in Anesthesiology **23**(5), 632–636 (2010)

11. Mehmet Sezgin, B.S.: Survey over image thresholding techniques and quantitative performance evaluation. In: Electronic Imaging. pp. 146–168 (2004)

12. Thouin, E., Hafiane, A., Vieyres, P., Xylourgos, N., Triantafyllidis, G., Papadourakis, G.: Nerve region segmentation for ultrasound guided local regional anaesthesia (lra). Mediterranean Conference on Information Systems (2011)

13. Tsui, B.C., Suresh, S.: Ultrasound imaging for regional anesthesia in infants, children, and adolescentsa review of current literature and its application in the practice of extremity and trunk blocks. Anesthesiology: The Journal of the American Society of Anesthesiologists **112**(2), 473–492 (2010)

14. Venkateswarlu Karnati, Mithun Uliyar, S.D.: Fast non-local algorithm for image denoising. In: International Conference on Image Processing, ICIP. pp. 3873–3876 (2009)

15. Woodworth, G.E., Chen, E.M., Horn, J.L.E., Aziz, M.F.: Efficacy of computer-based video and simulation in ultrasound-guided regional anesthesia training. Journal of clinical anesthesia **26**(3), 212–221 (2014)

16. Zdenek Kalal, K.M., Matas, J.: A computational approach to edge detection. In: International Conference on Pattern Recognition, ICPR. pp. 2756–2759. IEEE (2010)