Computerized Medical Imaging and Graphics 76 (2019) 101639

Contents lists available at ScienceDirect

Computerized Medical Imaging and Graphics

journal homepage: www.elsevier.com/locate/compmedimag

Deep visual nerve tracking in ultrasound images

Mohammad Alkhatib[a,b,*], Adel Hafiane[a], Pierre Vieyres[b], Alain Delbos[c]

[a] INSA Centre Val de Loire, Laboratoire PRISME EA 4229, Bourges F-18000, France
[b] Université d'Orléans, Laboratoire PRISME EA 4229, Bourges F-18000, France
[c] Clinique Médipôle Garonne, Toulouse F-31036, France

ARTICLE INFO

Article history:
Received 22 November 2018
Received in revised form 19 February 2019
Accepted 31 May 2019

Keywords:
Nerve tracking
Visual tracking
Deep-learning
Ultrasound images
Regional anesthesia

ABSTRACT

Ultrasound-guided regional anesthesia (UGRA) becomes a standard procedure in surgical operations and pain management, offers the advantages of nerve localization, and provides region of interest anatomical structure visualization. Nerve tracking presents a crucial step for practicing UGRA and it is useful and important to develop a tool to facilitate this step. However, nerve tracking is a very challenging task that anesthetists can encounter due to the noise, artifacts, and nerve structure variability. Deep-learning has shown outstanding performances in computer vision task including tracking. Many deep-learning trackers have been proposed, where their performance depends on the application. While no deep-learning study exists for tracking the nerves in ultrasound images, this paper explores thirteen most recent deep-learning trackers for nerve tracking and presents a comparative study for the best deep-learning trackers on different types of nerves in ultrasound images. We evaluate the performance of the trackers in terms of accuracy, consistency, time complexity, and handling different nerve situations, such as disappearance and losing shape information. Through the experimentation, certain conclusions were noted on deep learning trackers performance. Overall, deep-learning trackers provide good performance and show a comparative performance for tracking different kinds of nerves in ultrasound images.

1. Introduction

Regional anesthesia (RA) is an important procedure used in medical operations. RA is performed by the anesthetist close to a nerve in order to mask the sensation of pain in that part of the human body, improving postoperative mobility, and facilitating earlier hospital discharge (Horlocker et al., 2003). Traditionally, RA performed with a blind guidance which increased the risks of block failure, nerve trauma, and local anesthetic toxicity (Tsui and Suresh, 2010). Ultrasound-guided regional anesthesia (UGRA) has become the current trend to perform regional anesthesia, due to several advantages of the ultrasound (US) imaging such as low cost, no radiation, real-time acquisitions, and portability (Woodworth et al., 2014). However, this procedure requires a long learning process and years of experience (Woodworth et al., 2014; Marhofer and Chan, 2007). It remains challenging for the anesthetist to maintain both the needle and the nerve region in the ultrasound plane at the same time. As such, the aim of this study is to develop a tool to assist the anesthetists with accurate nerve tracking procedure.

Tracking is one of the fundamental tasks in computer vision and image analysis, and it is used in a wide range of applications such as video surveillance, medical imaging, robotics, etc. Tracking is an easy task when the target objects are isolated and easily distinguishable from the background, but it is a very challenging task when the image suffers from illumination changes, shape deformation, object disappearance, viewpoint variation, etc. (Wang and Yeung, 2013).

In the literature, various methods have been proposed to address these problems, where visual tracking can be categorized into two models. Motion model that predicts the states of an object (Comaniciu et al., 2003; Pérez et al., 2002; Li et al., 2008), and an observation model that take into account object appearance information and corrects its predictions (Li et al., 2013). The observation model, which has more impact than the motion model (Wang et al., 2015), is divided into generative methods that search for the most similar regions to the tracked object (such as Ross et al., 2008; Wang et al., 2015; Santner et al., 2010), and discriminative models that use classifiers to differentiate between the tracked object and its surrounding areas (such as Kalal et al., 2012; Yang et al., 2014; Zhang et al., 2014a,b; Henriques et al., 2015).

Tracking in US images is a very challenging task due to the degradation of the visual property of US images. Various methods have been proposed in the literature regarding tracking in US

[*] Corresponding author at: INSA Centre Val de Loire, Laboratoire PRISME EA 4229, Bourges F-18000, France.
E-mail address: mohammad.alkhatib@etu.univ-orleans.fr (M. Alkhatib).

https://doi.org/10.1016/j.compmedimag.2019.05.007
0895-6111/© 2019 Elsevier Ltd. All rights reserved.

images. Guerrero et al. used an elliptical model with Kalman filter to track the center of vessels in US images (Guerrero et al., 2007). In Nascimento and Marques (2008), the authors tracked left ventricles in US images by using non-linear filters with a multiple model data association tracker. In Tang et al. (2012), Tang et al. used Markov random fields to track the tongue contour automatically in US images. In Novotny et al. (2007), the authors tracked medical instruments in three-dimensional US images by searching for long straight objects using the generalized Radon transform. Roussos et al. introduced a variant of active appearance modeling to detect and track the tongue in US images (Roussos et al., 2009). In Li et al. (2005), the authors tracked the tongue in US images by incorporating intensity information with edge gradient to improve active contour. Duan et al. proposed a region-based method for endocardium tracking in US images Duan et al. (2009). To the best of our knowledge, there is one study for tracking the nerves in US images. In Alkhatib et al. (2018), the authors introduced an extensive study on different kinds of trackers with different kinds of features to track the median nerve in US images.

Although the traditional visual trackers provide acceptable results and show good abilities to handle different scene situations, it is more beneficial to exploit recent trackers based on deep learning processes, since it has shown excellent performance in many computer vision applications, such as image classification (Krizhevsky et al., 2012) and recognition (He et al., 2016).

Recently, convolutional neural networks (CNN) (LeCun et al., 1998) have received significant attention in computer vision and machine learning applications such as object detection (Girshick et al., 2014), image classification (Krizhevsky et al., 2012), and image segmentation (Long et al., 2015). Motivated by these breakthroughs, several deep-learning based trackers have been developed in order to significantly improve the tracking performance. These works showed promising results for different tracking applications, such as Danelljan et al. (2016, 2017), Zhang et al. (2016, 2017) and Song et al. (2017).

In the literature, few methods introduced tracking using deep-learning in US images. In Carneiro and Nascimento (2013), the authors built a deep neural network observation distribution to track the left ventricle endocardium in US images. In Nascimento et al. (2016), The authors used deep neural networks to build a new observation model in a particle filter to track and segment the left ventricle in US images. To the best of our knowledge, this is the first deep-learning study on nerve tracking in US images.

Here, we introduce a deep-learning approach to robustly track nerve structures in US images. We conducted a comparative study of thirteen deep trackers for two types of nerves, median and sciatic.

The major contributions of this paper can be summarized as follows:

- Tracking of nerve structure in ultrasound images.
- Comparative study of recent deep-learning tracking techniques.
- Addressing new medical applications (Regional anesthesia).

The structure of our paper is as follow. Section 2 details the deep-learning trackers. Followed by experimental results and discussion in Section 3. The paper ends with final conclusions in Section 4.

2. Deep visual trackers

This paper aims to track nerves in US images using deep-learning methods, and as such these methods should be robust enough to track different nerve situations. The visual tracker starts by generating the target model in the first frame, then extracts features in the next frame to find the candidate models, and find the best match between target and candidates models. Most existing deep trackers use CNN either to generate appearance models, to match object model with its candidates, or to distinguish the object from the surrounding areas. Therefore in the following, 13 deep-learning tracking methods are discussed where CNN is used.

2.1. Continuous convolution operators tracker

Continuous convolution operators tracker (C-COT) (Danelljan et al., 2016) employed a new technique that solves the learning problem by using an interpolation model which enables the integration of multi-resolution feature maps. This model takes the advantages of performing the convolution in the continuous spatial domain to obtain high results in the visual tracking problem. To reach precise sub-pixel localization, C-COT obtains the predicted score as a continuous function. Furthermore, C-COT's multi-resolution feature maps facilitate each visual feature to choose the region-size independently.

C-COT integrates the feature map x to the continuous spatial domain $t \in [0, T)$ by introducing an interpolation model J_d for each feature channel d,

$$J_d\left\{x^d\right\}(t) = \sum_{n=0}^{N_d-1} x^d[n] b_d\left(t - \frac{T}{N_d} n\right) \tag{1}$$

where x^d represents visual features, N_d is each feature resolution, and b_d is an interpolation kernel.

To predict the matching scores of the target image region, the convolution operator $S_f\{x\}$ is parametrized by a set of continuous T-periodic multi-channel convolution filters f,

$$S_f\{x\}(t) = f * J\{x\} = \sum_{d=1}^{D} f^d * J_d\left\{x^d\right\} \tag{2}$$

where D is the number of domain dimensions, and the filters are learned by minimizing the following,

$$E(f) = \sum_{j=1}^{M} \alpha_j \left\| S_f\left\{x_j\right\} - y_i \right\|_{L^2}^2 + \sum_{d=1}^{D} \left\| w f^d \right\|_{L^2}^2 \tag{3}$$

where M is the number of training samples, α_j is the weight of sample x_j, y_i is the labeled detection scores of x_j sample and represented by periodically repeated Gaussian function, w parameter is used to reduce the effect of periodic assumption, and the L_2-norm is weighted classification error.

2.2. Efficient convolution operators

Efficient convolution operators (ECO) (Danelljan et al., 2017) is based on C-COT and aims to reduce the C-COT model size, the training set size, and excessiveness and sensitivity of the model update. In other words, ECO aims to reduce the redundancy in the sample set of C-COT.

Unlike C-COT which learns one separate filter for each feature channel, ECO provides a smaller set c of filters by using a factorized convolution operator P. Therefore, ECO replaces Eq. (2) by,

$$S_{Pf}\{x\}(t) = Pf * J\{x\} = \sum_{c,d} P_{d,c} f^c * J_d\left\{x^d\right\} = f * P^T J\{x\} \tag{4}$$

C-COT collect new samples in each frame, which leads to sampling set redundancy. For that, while preserving the samples diversity, ECO reduces their number in the learning phase by a generative model of the training sample space. The generative model finds the filter that minimizes the expected correlation error,

$$E(f) = \mathbb{E}\left\{ \left\| S_f\{x\} \right\| - y \|_{L^2}^2 \right\} + \sum_{d=1}^{D} \left\| wf^d \right\|_{L^2}^2 \tag{5}$$

The model strategy in C-COT is updated in each frame which leads to slow tracking procedure. ECO uses a sparse updating scheme to reach more fast and robust tracking. This scheme is concluded by updating the model once a sufficient change happens.

2.3. Convolutional network based tracker

Convolutional network based tracker (CNT) (Zhang et al., 2016) considered a simple two-layer convolutional network that developed a robust sparse representation for visual tracking. CNT uses CNN network to extract mid-level features from the image. These features are then distinguished into positive and negative ones by learning a classifier. Using a k-means algorithm, CNT initializes fixed filters by extracting a set of normalized patches from the target region. Around the target region, these filters define a set of feature maps which encode target local structural information. These feature maps construct the first layer, while the second layer consists of the encoded target local structural information. To adapt to target appearance variations, an online strategy is employed to update the model. The online strategy is a sparse representation which adapts a simple temporal low-pass filtering method,

$$c_t = (1-\rho)c_{t-1} + \rho\hat{c}_{t-1} \tag{6}$$

where c_t is the target template at frame t, ρ is a learning parameter, and $\hat{c}_{t-1}$ is the sparse representation of the tracked target at frame $t-1$.

2.4. Multi-domain convolutional neural networks

Multi-domain convolutional neural networks (MDNet) (Nam and Han, 2016) has five hidden layers which consist of shared layers and multiple branches of domain-specific layers. These domains correspond to individual training sequences and each branch is representing binary classification to identify the target in each domain. MDNet has K branches in the last fully connected layers, where each K branch contains a binary classification layer. These binary classification layers are to distinguish the target and the background in each domain. Using a large set of videos with tracking ground-truths, MDNet trains each domain to construct a target representation.

MDNet trained CNN using the stochastic gradient descent (SGD) method, wherein each iteration of each domain is handled separately. MDNet tracks the target by evaluating the randomly sampled candidate windows around the target region in the previous frame. While tracking, MDNet incorporates the shared layers in the pre-trained CNN with a new binary classification layer to develop a new network. In order to update the network, MDNet applies short-term and long-term update strategies. Furthermore, MDNet adapts bounding box regression technique for more precise target localization.

2.5. Structure-aware network

In general, structure-aware network (SANet) (Fan and Ling, 2017) follows the same strategy as MDNet but with an additional recurrent neural network (RNN) based structure for improving object representation. Fig. 1 shows the structure of SANet and how RNNs is utilized. Using multiple RNNs, SANet models object structure during learning then combines it into CNN. SANet consists of two fully connected layers and one fully connected classification layer which concatenated with the recurrent layers using a skip concatenation strategy. For training, target tracking, model update, and target localization, SANet adapts the same strategies as MDNet.

2.6. Fully-convolutional Siamese networks

Fully-convolutional Siamese networks (SiameFC) (Bertinetto et al., 2016) utilize a sliding-window evaluation by using a bilinear layer that computes the cross-correlation of its two inputs. SiameFC starts by initializing the interested object location (x') in the first frame and the search area in the next frame (z'). A convolutional embedding CNN function (f_ρ), with learnable parameter ρ, is used to represent each feature map for the inputs. To find the similarity between these inputs, SiameFC computes the cross-correlation between the two feature maps $f_\rho(x')$ and $f_\rho(z')$ using,

$$g_\rho(x', z') = f_\rho(x') \star f_\rho(z') \tag{7}$$

where ρ is a learnable parameter.

The goal of Eq. (7) is to find the best value between the two input feature maps. And to achieve this goal, SiameFC offline trains the network with a huge number of random targets taken from a large database of videos. A spatial map of labels c_i is assembled for each sample. The training is performed by minimizing an element-wise logistic loss over the training set,

$$arg \min_{\rho} \sum_{i} \ell(g_\rho(x'_i, z'_i), c_i) \tag{8}$$

The tracking process is performed by comparing the target object with the candidate's locations. These candidates are located in the search region which is centered in the same previous target location center but four-times bigger in size. The new target location is found by taking the candidate with the highest similarity score.

2.7. Correlation filter network

Correlation filter network (CFNet) (Valmadre et al., 2017) based on Siamese networks by modifying the correlation filter learner as a differentiable layer in the deep neural network. The correlation filter block between the interested object location x' and the cross-correlation operator is set as,

$$h_{\rho,s,b}(x', z') = s\,w(f_\rho(x')) \star f_\rho(z') + b \tag{9}$$

where the scalar parameters s and b, the scale and the bias respectively, are used to make the score range suitable for logistic regression. z' is the search areas in the next frame. f_ρ is a convolutional embedding CNN function. Also, w is the standard convolution filter template computed by the convention filter block from the training feature map $x = f_\rho(x')$.

2.8. Discriminant correlation filters network

Discriminant correlation filters network (DCFNet) (Wang et al., 2017) is based on Siamese network and aimed to do simultaneously convolutional features learning and correlation tracking. DCFNet added a correlation layer for backpropagation to Siamese network using an object location probability heat map.

The discriminant correlation filter w can be obtained by,

$$\hat{\mathrm{w}}^l = \frac{\hat{\varphi}^l(\mathrm{x}) \odot \hat{\mathrm{y}}^*}{\sum_{k=1}^{D} \hat{\varphi}^l(\mathrm{x}) \odot (\hat{\varphi}^l(\mathrm{x}))^* + \lambda} \tag{10}$$

where $\hat{\mathrm{w}}^l$ presents filter w of channel l, $\varphi \in \mathbb{R}^{M\times N\times D}$ is the target patch features, $\mathrm{y} \in \mathbb{R}^{M\times N}$ is the ideal response, λ is the regularization coefficient constant, the $\hat{()}$ represents discrete Fourier

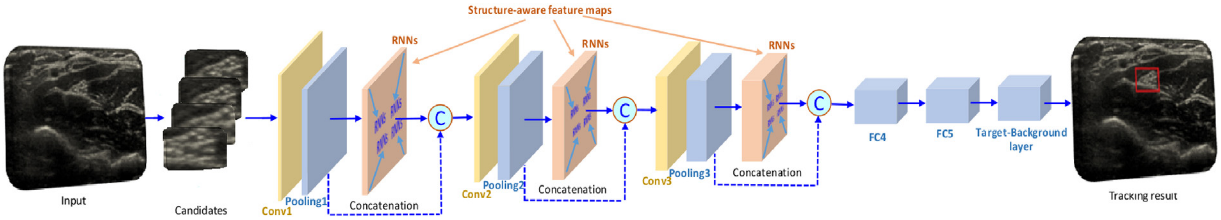


Fig. 1. Illustration of SANet tracker (Fan and Ling, 2017).

transform $\mathcal{F}$, $(^*)$ is the complex conjugate of a complex number y, and $\odot$ refers to Hadamard (element-wise) product.

In the new frame, DCFNet detection process starts by cropping a search window and finding the features inside it. Then DCFNet estimates target translation by using correlation response map maximum value,

$$g = \mathcal{F}^{-1}\left(\sum_{l=1}^{D} \hat{w}^{l*} \odot \hat{\varphi}^{l}(z)\right) \tag{11}$$

DCFNet uses incremental filter update which regards it as an RNN network. DCFNet incrementally updates the filter over time t during the online tracking which gives the advantages of maintaining only a small sample set. Filter update can be found using,

$$\hat{w}_p^l = \frac{\sum_{t=1}^{p} \beta \hat{y}^* \odot \hat{\varphi}^l(x_t)}{\sum_{t=1}^{p} \beta_t \left(\sum_{k=1}^{D} \hat{\varphi}^k(x_t) \odot \left(\hat{\varphi}^k(x_t)\right)^* + \lambda\right)} \tag{12}$$

where β_t is the impact of sample x_t.

2.9. Multi-task correlation particle filter

Multi-task correlation particle filter (MCPF) (Zhang et al., 2017) takes the benefits of controlling the particle sampling in particle filter using a multi-task correlation filter. The learning process of the correlation filter leads to using fewer particles which decreases the computation cost. MCPF learns the correlation filter jointly by including different feature inter-dependencies.

Multi-task correlation filter (MCF) learned z_k to differentiate discriminative training samples x_k of target from the background, where K is the features (CNN or HOG). To find the features with similar z_k which are more stable, MCF applies circular shifts to z_k, where all possible circular shifts of an image patch of $M \times N$ pixels are $x_{m,n} \in \{0, 1, \ldots, M-1\} \times \{0, 1, \ldots, N-1\}$ which represent the possible locations of the target object. After that, MCF learns the correlation filters for z_k using,

$$\min_{\{z_k\}_{k=1}^{K}} \sum_k \frac{1}{4\lambda} z_k^\top G_k z_k + \frac{1}{4} z_k^\top z_k - z_k^\top y + \gamma \left\|Z\right\|_{2,1} \tag{13}$$

where $Z = [z_1, z_2, \ldots, z_K]$ is obtained by gathering learned z_k of K different features, λ is a regularization parameter, G_k is equal to $X_k X_K^\top$, X_k donates all training samples with Gaussian function label y, γ is a tradeoff parameter between reliable reconstruction and joint sparsity regularization. Finally, to find multi-task correlation filter z_k for each type of feature, Eq. (13) is solved using the accelerated proximal gradient method.

MCPF first step consists of generating the particle, followed by predicting the particle location using the probabilistic framework. After that, MCF is applied to each particle with the aim of shifting the particles to a stable location using its circular shifts. Then MCPF updates the weights using the response map,

$$r = \sum_k \mathcal{F}^{-1}\left(\mathcal{F}(z_k) \odot \mathcal{F}\left(\left\langle y_t^i, \bar{x}\right\rangle\right)\right) \tag{14}$$

where z_k is the learned MCF, $\bar{x}$ is the target appearance model, y_t^i is the observation of particle i at time t, $\odot$ donates Hadamard (element-wise) product, and $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier transform and its inverse, respectively.

The optimal state can be formulated as,

$$E\left[s_t \mid y_{1:t}\right] \approx \sum_{i=1}^{n} w_t^i S_{mcf}\left(s_t^i\right) \tag{15}$$

where each particle s_t^i is shifted $S_{mcf}\left(s_t^i\right)$, and w_t^i is particle weights and is proportional to the response of the MCF.

MFCF updates MCF using an incremental strategy which only utilizes current frame new samples x_k,

$$\begin{aligned} \mathcal{F}(\bar{x}_k)^t &= (1-\eta)\,\mathcal{F}(\bar{x}_k)^{t-1} + \eta \mathcal{F}(x_k)^t \\ \mathcal{F}(z_k)^t &= (1-\eta)\,\mathcal{F}(z_k)^{t-1} + \eta \mathcal{F}(z_k)^t \end{aligned} \tag{16}$$

2.10. Hedged deep tracking

Hedged deep tracking (HDT) (Qi et al., 2016) is based on adaptive hedge method which solves the online learning problems in a multi-expert multi-round setting. HDT takes the advantages of VGG-Net (Simonyan and Zisserman, 2014) deep architecture to extract feature maps of convolutional layers from image regions. To generate response maps, each feature map is convolved by correlation filters to generate a weak tracker (expert). HDT hedges these trackers into a stronger one using an online decision-theoretical Hedge algorithm as shown in Fig. 2. At time t, the target position (x_t^*, y_t^*) can be found by the position with the best response,

$$\left(x_t^*, y_t^*\right) = \sum_{k=1}^{K} w_t^k \cdot \left(x_t^k, y_t^k\right) \tag{17}$$

where w_t^k is the weight at time t for expert k. w_t^k uses new samples $\bar{\mathcal{X}}^k$ in the current frame to update the previous models which reflect each experts decision loss,

$$\begin{aligned} \mathcal{Z}_{*,*,d}^k &= \frac{\mathcal{Y}}{\bar{\mathcal{X}}^k . \bar{\mathcal{X}}^k + \lambda} \odot \bar{\mathcal{X}}_{*,*,d}^k \\ \mathcal{W}_t^k &= (1-\eta)\mathcal{W}_{t-1}^k + \eta \mathcal{Z}_t^k \end{aligned} \tag{18}$$

where $\mathcal{W}_t^k$ is the kth filter which modeled in the Fourier domain, $\bar{\mathcal{X}}$ represents the new samples in the current frame kth convolutional layer, $\mathcal{Y}$ is a 2D Gaussian distribution with zero mean and standard deviation proportional to the target size, $\odot$ denotes the Hadamard (element-wise) product, λ is a tradeoff parameter, and η is the learning rate.

2.11. Hierarchical convolutional features tracker

Hierarchical convolutional features tracker (HCFT) (Ma et al., 2015) utilizes large-scale datasets to learn rich feature hierarchies of CNNs. HCFT is robust to appearance variations due to keeping target objects semantics in the last convolutional layers. Target

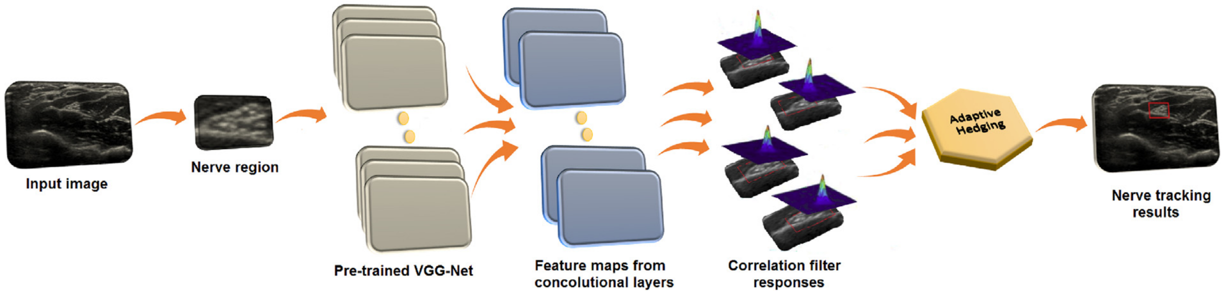


Fig. 2. Illustration of HDT tracker (Qi et al., 2016).

appearance is encoded using a correlation filter on each convolutional layer. HCFT uses the hierarchies of convolutional layers to find the maximum responses that locate the target.

HCFT employs VGG-Net (Simonyan and Zisserman, 2014) convolutional feature maps to encode target appearance. Followed by finding the maximum response using correlation filters in the frequency domain,

$$W^d = \frac{Y \odot \bar{X}^d}{\sum_{i=1}^{D} X^i \odot \bar{X^i} + \lambda} \tag{19}$$

where D is the number of channels, Y is a Fourier transform of a Gaussian function y, X is the feature vector of the d channel, the bar refers to the complex conjugation, $\odot$ is Hadamard (element-wise) product, and λ is a regularization parameter.

HCFT finds the response map in the new frame by providing the feature vector ‡ ($\bar{Z}$ using Fourier transform) on the lth layer of a patch,

$$f_l = \mathcal{F}^{-1}\left(\sum_{d=1}^{D} W^d \odot \bar{Z}^d\right) \tag{20}$$

The new target location is set to the patch with the maximum response. HCFT update the model using moving average technique for the numerator A^d and denominator B^d of the correlation filter W^d,

$$A_t^d = (1-\eta)A_{t-1}^d + \eta Y \odot \bar{X}_t^d,$$

$$B_t^d = (1-\eta)B_{t-1}^d + \eta \sum_{i=1}^{D} X_t^i \odot \bar{X}_t^i, \tag{21}$$

$$W_t^d = \frac{A_t^d}{B_t^d + \lambda}$$

where t is the frame index and η is a learning rate.

2.12. Convolutional residual tracker

Convolutional residual tracker (CREST) (Song et al., 2017) uses end-to-end training and reformulates discriminant correlation filters (DCFs) as a one-layer convolutional neural network. The residual learning process is adopted to reduce model degradation and to be more invariant to appearance changes.

DCF learns a filter W by solving the minimization problem

$$\arg\min_{W} \left\|W * X - Y\right\|^2 + \lambda\left\|W\right\|^2 \tag{22}$$

where X is the input sample, Y is its corresponding Gaussian function label, and λ is the regularization parameter.

CREST initializes the model by extracting features, mapping the response, and setting the distribution parameters in the base and residual layers. For the new frame, CREST generates the response map and sets the new target location to the patch with the maximum response. CREST handles scale changes by extracting different scale patches around target new location center. These patches are sent to the response map and the patch with the maximum response sets to be the target new scale.

CREST initializes a ground-truth response map in each frame and the search patch is adopted as a training patch. These ground-truth maps and the training patch update online the network.

2.13. Deep-learning tracker

Deep-learning tracker (DLT) (Wang and Yeung, 2013) adopts the neural networks outputs as object features to localize the target object. DLT learns generic image features using stacked de-noising autoencoder (SDAE). Followed by transferring the offline training to online tracking. Using continuously tuned classification neural networks and feature extractor, DLT adapted to target object appearance changes.

DLT starts by collecting features from the target and its surrounding areas, then pass it to SDAE for offline training. For the new frame, DLT adopts particle filter technique by spreading particles around the previous target location. Using the network, a confident map is determined for each particle and the new location sets to the particle with the highest confidence. If all particles have the same confidence, this concludes an appearance change of the target and the network should be tuned again.

3. Experiments, results and discussion

In UGRA, the anesthetist starts by using the US probe to scan a part of the body back and forth in order to locate and track the nerve. This step is important to stabilize the probe in a good position to visualize the nerve and insert the needle.

In this paper, we conduct the experiment on two different nerves which have different characteristics, median and sciatic. The median nerve is one of the major nerves in the arm, it starts from the brachial plexus to innervate the intrinsic muscles of the hand. The sciatic nerve is located in the leg, more specifically in the popliteal fossa. The median nerve presents a circular, oval or elliptic shape, whereas the sciatic nerve sometimes presents an irregular shape which makes it harder to visualize and to track (Heinemeyer and Reimers, 1999).

This study shows the feasibility of nerve tracking in US images using deep-learning approaches. This experiment provides a performance comparison and evaluation of deep-learning approaches for nerve tracking in ultrasound images. Each method is analyzed in term of accuracy, consistency, time cost, and handling different nerve situations. In this section, we first describe the used dataset and setup, then analyze and discuss results and performances.

3.1. Dataset and setup

Experiments were conducted on sonographic videos of the median and sciatic nerves obtained from 42 anonymous adult patients using an ultrasound machine with an MHz transducer frequency (one video per patient). A total number of 10,337 ultrasound images of the nerve were used, which include 25 videos of median nerves with an average of 335 image per video, and 17 videos of sciatic nerves with an average of 120 image per video. The dataset is ethically approved, and it was acquired in real conditions at the Medipole Garonne hospital in Toulouse (France). The ground-truth was provided by two regional anesthesia experts.

It is well known that the visual properties of US images are degraded by many effects such as artifacts, signal degradation, and speckle noise. These are caused by the coherent source and noncoherent detector of echo ultrasound imaging systems. With the aim of performing in real-time, nerve tracking performed directly on the original US image without any prior image enhancements.

The experiments were carried out with a core 7 Duo 3.50 GHz processor with 32GB RAM under Matlab. In this experiment, we conduct a comprehensive experimental evaluation of 13 deep visual trackers introduced in Section 2 and one visual tracker for nerve tracking from Alkhatib et al. (2018). In Alkhatib et al. (2018), the authors tracked the median nerve using hand crafted features, where AMBP texture descriptor with Particle filter showed the best results. In this paper, AMBP-PF is tested, as the best method obtained in Alkhatib et al. (2018), to provide a comparative study between hand crafted features and deep-learning features. The 13 deep visual trackers have achieved top performance on OTB-100 (Wu et al., 2015), TC-128 (Liang et al., 2015) and VOT2015 (Kristan et al., 2015) datasets. For evaluating each method, the same parameters provided by the original papers were used along with the source codes that have been made available by the original authors. In this experiment, VGG-Net (Simonyan and Zisserman, 2014), very deep convolutional networks (up to 19 layers) are adopted for feature extraction. Overall, this experiment shows the benefits of exploiting deep-learning visual tracking in US images.

3.2. Results

For more accurate and extensive results, two evaluation procedures are performed in this experiment. The first accuracy evaluation is assessed by the bounding box overlap ratio between the estimated nerve position and the ground-truth. The overlap ratio is based on pixels percentage in the intersection area. The second evaluation methodology is precision and success plots (Wu et al., 2015). The trackers are ranked in terms of distance precision (DP) and area under the curve (AUC), respectively. The precision plot represents the percentage that the center location error is below a predefined threshold (20 pixels in this experiment), where the center location error uses the average Euclidean distance between target estimated center and ground-truth center. The success plot shows if the target is being tracked successfully by finding if the overlap score between the estimated bounding box and the ground-truth is larger than a predefined threshold (0.5 in this experiment).

Fig. 3 illustrates the tracking methods accuracy for median nerve, where ECO, C-COT, and SANet achieved the best results, while other methods suffer from less stability and less performance accuracy. Also, it can be seen that CREST gave good results but with less stability. On the other hand, compared to a median nerve, a sciatic nerve is harder to track due to its location, shape, and appearance which is almost the same as the surrounding areas in certain frames. In Fig. 5, ECO provided the best performance and it can be seen that HDT and MCPF gave a good performance but with less stability. Fig. 4 shows qualitative results of tracking median and sciatic nerves using ECO method. As we have previously stated, Fig. 4 provides the ground-truth.

Table 1
Tracking scores (%) comparison between the proposed tracking methods.

Method	Nerves		
	Median	Sciatic	Overall
C-COT (Danelljan et al., 2016)	**0.94**	0.73	0.84
ECO (Danelljan et al., 2017)	**0.94**	**0.80**	**0.87**
CNT (Zhang et al., 2016)	0.79	0.73	0.76
MDNet (Nam and Han, 2016)	0.93	0.73	0.82
SANet (Fan and Ling, 2017)	**0.94**	0.76	0.85
SiameFC (Bertinetto et al., 2016)	0.82	0.74	0.78
CFNet (Valmadre et al., 2017)	0.85	0.77	0.81
DCFNet (Wang et al., 2017)	0.86	0.73	0.79
MCPF (Zhang et al., 2017)	0.89	0.79	0.84
HDT (Qi et al., 2016)	0.87	**0.80**	0.83
HCFT (Ma et al., 2015)	0.85	0.78	0.82
CREST (Song et al., 2017)	0.92	0.73	0.83
DLT (Wang and Yeung, 2013)	0.86	0.75	0.80
PF-AMBP (Alkhatib et al., 2018)	0.87	0.71	0.82

Notes: The highest score is represented in bold.

Fig. 6 reports the precision plot and the success plot over the median and sciatic nerves videos, where it illustrates a comparison of all deep trackers. DP and AUC scores for each tracker are shown in the figure legend. For the median nerve and among the compared methods, ECO tracker provides the best results with DP and AUC scores of 95% and 75%. SANet tracker achieves the second best results in both DP and AUC scores. For sciatic nerve, ECO and HDT trackers reach more than 72% for DP score and more than 61% for AUC score. Overall, it can be noticed ECO tracker outperforms other trackers and achieves the best results in both precision and success plots.

3.3. Discussion

In this paper, we addressed a challenging problem in UGRA which is nerve tracking. To deal with this problem several CNN-based methods have been introduced. Table 1 depicts tracking methods performance for median and sciatic nerves. For the median nerve, using ECO provides the best results, where these results are obtained due to transferring prior visual via pre-training and capturing any appearance changes via online learning. C-COT adopts the same maneuver as ECO, but ECO provides a better generalization of the target by avoiding the over-fitting. Other good trackers for median nerve are SANet and MDNet which achieved a good score caused by using a particle filter framework in its design. As well as this, SANet incorporates with an RNN scheme which leads to an increase in the tracking accuracy. For the sciatic nerve, using ECO or HDT achieves the best results, where the HDT results are obtained as a result of its hedging properties.

CNT tracker uses one convolutional layer, while others use deeper convolutional layers such as ECO and HCFT. In this experiment, it was observed that using more deep layers results in a better performance and improves the tracking accuracy.

For the median nerve and comparing between CNN-based deep trackers and traditional (hand crafted features) trackers such as particle filter (PF) with adaptive median binary pattern (AMBP) features (Alkhatib et al., 2018), PF-AMBP achieves good results and outperforms few deep-learning trackers. While for the sciatic nerve, the CNN-based deep tracking algorithms achieved a better performance than the traditional trackers in terms of accuracy and stability thanks to deep features strong representation. Finally, it can be observed that ECO tracker provides the best results among CNN-based deep trackers for both median and sciatic nerve tracking and gives the best stable results. Overall, the accuracy of CNN-based

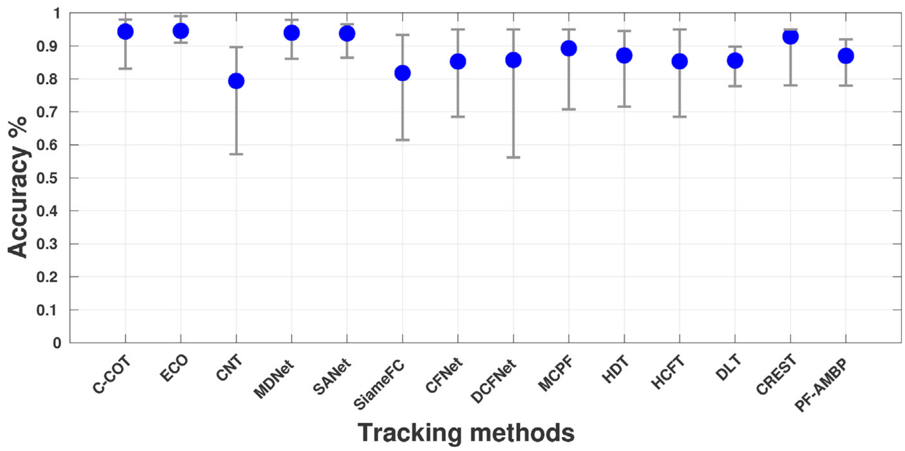


Fig. 3. The performance of deep-learning trackers for the median nerve, where accuracy and stability are shown.

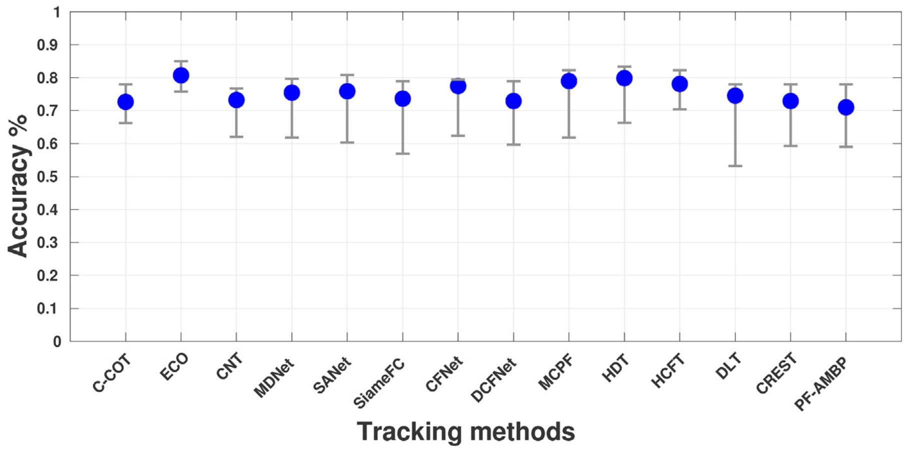


Fig. 5. The performance of deep-learning trackers for the sciatic nerve, where accuracy and stability are shown.

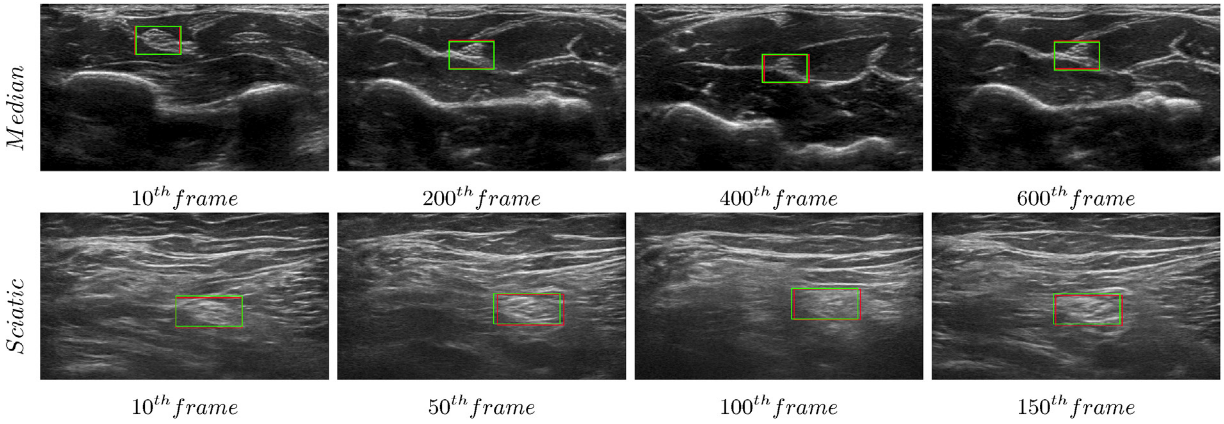
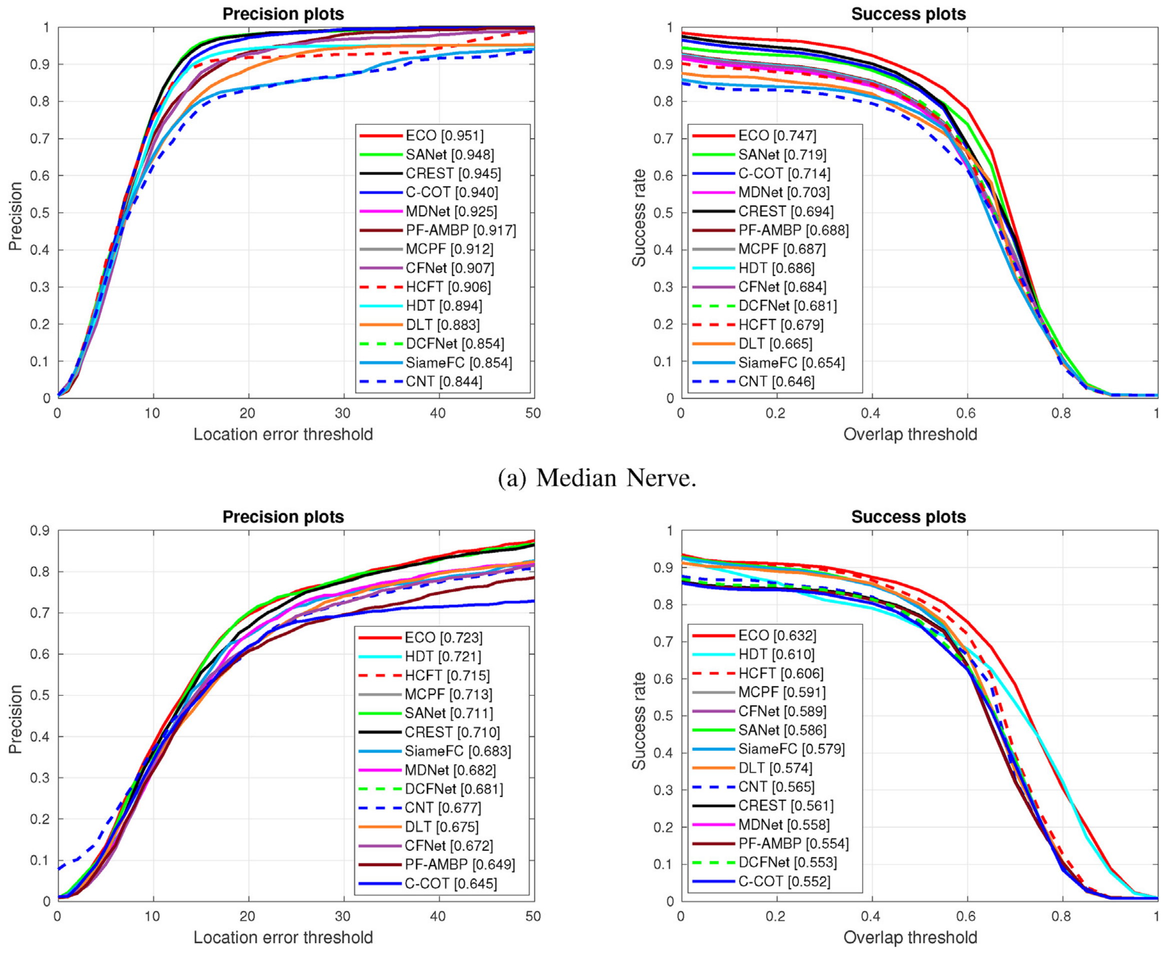


Fig. 4. Nerve tracking using ECO tracker on median nerve and sciatic nerve (red rectangle for ECO method and a green rectangle for the ground-truth). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

deep trackers is competitive and provides good performance for tracking the median and sciatic nerves.

Time complexity is considered a crucial point and an important aspect for visual tracking, especially in medical applications. The used dataset was recorded at 20 *frames*/*s*. Table 2 demonstrates the running time for each method where it shows that DCFNet provides the best processing time. While ECO is slow, C-COT, SANet, and HDT are much slower.

Important aspects affect the running time for CNN-based deep tracking algorithms, which are the number of layers and model update strategy. Some trackers use more deep layers while others use fewer layers which make the tracker run faster. The other important aspect is the tracker model update strategy, where it can be noticed that updating the model after each frame is time-consuming. For that, ECO updates its model every few frames which make the process run faster. Another strategy to update the model is using Siamese network to model prior information that accelerates



(a) Median Nerve.

(b) Sciatic Nerve.

Fig. 6. Precision plot and success plot over 42 US videos on 14 tested trackers.

Table 2
Tracking speed (*spf*) between the proposed tracking methods

Method	Nerves	
	Median	Sciatic
C-COT (Danelljan et al., 2016)	0.65	0.67
ECO (Danelljan et al., 2017)	0.13	0.14
CNT (Zhang et al., 2016)	0.63	0.65
MDNet (Nam and Han, 2016)	1.00	1.20
SANet (Fan and Ling, 2017)	1.63	1.74
SiameFC (Bertinetto et al., 2016)	0.26	0.31
CFNet (Valmadre et al., 2017)	0.45	0.49
DCFNet (Wang et al., 2017)	**0.04**	**0.05**
MCPF (Zhang et al., 2017)	0.60	0.66
HDT (Qi et al., 2016)	0.89	1.30
HCFT (Ma et al., 2015)	0.27	0.31
CREST (Song et al., 2017)	0.9	1.5
DLT (Wang and Yeung, 2013)	0.33	0.42
PF-AMBP (Alkhatib et al., 2018)	0.59	0.63

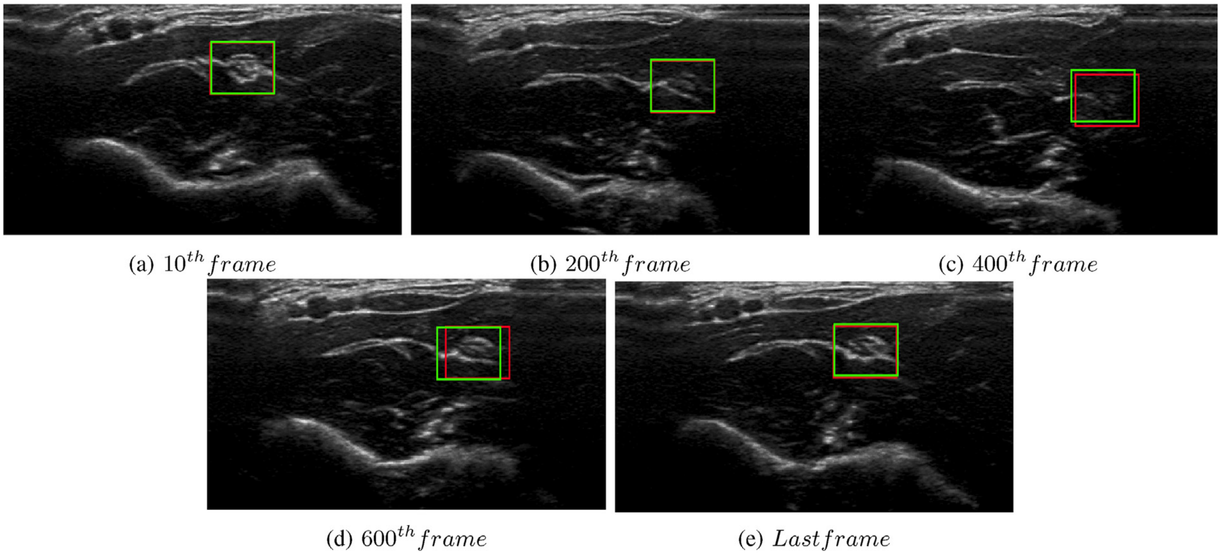
Note: The highest score is represented in bold.

the running process such as CFNet, DCFNet, and SiameseFCs. While ECO is not the fastest method but, at the same time, it provides a good trade-off between tracking accuracy and time complexity.

The experiments faced some challenges when the nerve disappeared or appeared to be almost as identical as the surrounding areas. Losing the nerve and failing to re-track it leads to tracking failure, which makes this challenge significant. The MCPF tracker uses particle filter principle which gives it the ability to re-track the nerve in case of disappearance. Other trackers expand their localization to re-track the nerve once it appears again such as ECO and DCFNet. On the other hand, CREST tracker failed to re-track the nerve rapidly after it appeared again. Fig. 7 shows an example of how MCPF succeeds in tracking the nerve even when the nerve almost disappeared.

4. Conclusion

Accurate and consistent nerve tracking is essential for safe and efficient Ultrasound-Guided Regional Anesthesia operation. In this paper, we perform nerve tracking in ultrasound images using deep-learning techniques. Recent deep-learning trackers are introduced to track nerve regions in ultrasound images. Different procedures were used to evaluate and compare the deep-learning trackers to demonstrate the effectiveness, robustness, and speed of the deep trackers. In this study, nerve tracking was performed directly on the original environment ultrasound images without any prior image enhancements which makes it a very challenging task. Our findings show that ECO, SANet, and C-COT outperform other techniques for tracking the median nerve, and ECO, HDT, and MCPF for tracking the sciatic nerve. Overall, deep-learning trackers showed good performance and handled noise suppression without pre-filtering the images. In future work, tracking techniques will be assessed on other types of nerves in order to improve the performance.


(a) $10^{th} frame$

(b) $200^{th} frame$

(c) $400^{th} frame$

(d) $600^{th} frame$

(e) $Last frame$

Fig. 7. Nerve tracking using MCPF tracker. Although the existence of nerve disappearance, the tracker succeeded to predict the nerve location (red rectangle for MCPF tracker and a green rectangle for the ground-truth). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Conflict of interest statement

No conflict of interest.

Acknowledgements

This work is part of the DANIEAL 2 project supported by Centre Val de Loire Region (France) grant 2016-00108375. We would like to thank Region Centre Val de Loire for supporting the work.

References

Alkhatib, M., Hafiane, A., Tahri, O., Vieyres, P., Delbos, A., 2018. Adaptive median binary patterns for fully automatic nerves tracking in ultrasound images. Comput. Methods Programs Biomed. 160, 129–140.

Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H., 2016. Fully-convolutional Siamese networks for object tracking. In: European Conference on Computer Vision, Springer, pp. 850–865.

Carneiro, G., Nascimento, J.C., 2013. Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data. IEEE Trans. Pattern Anal. Mach. Intell. 99 (1), 1.

Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25 (5), 564–577.

Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M., 2016. Beyond correlation filters: learning continuous convolution operators for visual tracking. In: European Conference on Computer Vision, Springer, pp. 472–488.

Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M., 2017. Eco: efficient convolution operators for tracking. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 21–26.

Duan, Q., Angelini, E.D., Herz, S.L., Ingrassia, C.M., Costa, K.D., Holmes, J.W., Homma, S., Laine, A.F., 2009. Region-based endocardium tracking on real-time three-dimensional ultrasound. Ultrasound Med. Biol. 35 (2), 256–265.

Fan, H., Ling, H., 2017. Sanet: structure-aware network for visual tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, pp. 2217–2224.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580–587.

Guerrero, J., Salcudean, S.E., McEwen, J.A., Masri, B.A., Nicolaou, S., 2007. Real-time vessel segmentation and tracking for ultrasound imaging applications. IEEE Trans. Med. Imaging 26 (8), 1079–1090.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

Heinemeyer, O., Reimers, C.D., 1999. Ultrasound of radial, ulnar, median, and sciatic nerves in healthy subjects and patients with hereditary motor and sensory neuropathies. Ultrasound Med. Biol. 25 (3), 481–485.

Henriques, J.F., Caseiro, R., Martins, P., Batista, J., 2015. High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37 (3), 583–596.

Horlocker, T.T., Wedel, D.J., Benzon, H., Brown, D.L., Enneking, K.F., Heit, J.A., Mulroy, M.F., Rosenquist, R.W., Rowlingson, J., Tryba, M., et al., 2003. Regional anesthesia in the anticoagulated patient: defining the risks (the second ASRA consensus conference on neuraxial anesthesia and anticoagulation). Reg. Anesth. Pain Med. 28 (3), 172–197.

Kalal, Z., Mikolajczyk, K., Matas, J., 2012. Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. 34 (7), 1409–1422.

Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R., 2015. The visual object tracking vot2015 challenge results. Proceedings of the IEEE International Conference on Computer Vision Workshops, 1–23.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Li, M., Kambhamettu, C., Stone, M., 2005. Automatic contour tracking in ultrasound images. Clin. Linguist. Phonet. 19 (6-7), 545–554.

Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M., 2008. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. IEEE Trans. Pattern Anal. Mach. Intell. 30 (10), 1728–1740.

Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D., 2013. A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol. 4 (4), 58.

Liang, P., Blasch, E., Ling, H., 2015. Encoding color information for visual tracking: algorithms and benchmark. IEEE Trans. Image Process. 24 (12), 5630–5644.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440.

Ma, C., Huang, J.-B., Yang, X., Yang, M.-H., 2015. Hierarchical convolutional features for visual tracking. Proceedings of the IEEE International Conference on Computer Vision, 3074–3082.

Marhofer, P., Chan, V.W., 2007. Ultrasound-guided regional anesthesia: current concepts and future trends. Anesth. Analg. 104 (5), 1265–1269.

Nam, H., Han, B., 2016. Learning multi-domain convolutional neural networks for visual tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 4293–4302.

Nascimento, J.C., Marques, J.S., 2008. Robust shape tracking with multiple models in ultrasound images. IEEE Trans. Image Process. 17 (3), 392–406.

Nascimento, J.C., Carneiro, G., Freitas, A., 2016. Tracking and segmentation of the endocardium of the left ventricle in a 2D ultrasound using deep learning architectures and Monte Carlo sampling. Biomedical Image Segmentation: Advances and Trends, 387.

Novotny, P.M., Stoll, J.A., Vasilyev, N.V., Pedro, J., Dupont, P.E., Zickler, T.E., Howe, R.D., 2007. GPU based real-time instrument tracking with three-dimensional ultrasound. Med. Image Anal. 11 (5), 458–464.

Pérez, P., Hue, C., Vermaak, J., Gangnet, M., 2002. Color-based probabilistic tracking. In: European Conference on Computer Vision, Springer, pp. 661–675.

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.-H., 2016. Hedged deep tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4303–4311.

Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H., 2008. Incremental learning for robust visual tracking. Int. J. Comput. Vision 77 (1-3), 125–141.

Roussos, A., Katsamanis, A., Maragos, P., 2009. Tongue tracking in ultrasound images with active appearance models. In: 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE, pp. 1733–1736.

Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H., 2010. Prost: parallel robust online simple tracking. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 723–730.

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R.W., Yang, M.-H., 2017. Crest: convolutional residual learning for visual tracking. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, pp. 2574–2583.

Tang, L., Bressmann, T., Hamarneh, G., 2012. Tongue contour tracking in dynamic ultrasound via higher-order MRFs and efficient fusion moves. Med. Image Anal. 16 (8), 1503–1520.

Tsui, B.C., Suresh, S., 2010. Ultrasound imaging for regional anesthesia in infants, children, and adolescents – a review of current literature and its application in the practice of extremity and trunk blocks. Anesthesiology: J. Am. Soc. Anesthesiol. 112 (2), 473–492.

Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H., 2017. End-to-end representation learning for correlation filter based tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 5000–5008.

Wang, N., Yeung, D.-Y., 2013. Learning a deep compact image representation for visual tracking. Advances in Neural Information Processing Systems, 809–817.

Wang, N., Shi, J., Yeung, D.-Y., Jia, J., 2015. Understanding and diagnosing visual tracking systems. Proceedings of the IEEE International Conference on Computer Vision, 3101–3109.

Wang, Q., Gao, J., Xing, J., Zhang, M., Hu, W., 2017. DCFNet: Discriminant Correlation Filters Network for Visual Tracking. arXiv:1704.04057.

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

Wu, Y., Lim, J., Yang, M.-H., 2015. Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. 37 (9), 1834–1848.

Yang, F., Lu, H., Yang, M.-H., 2014. Robust superpixel tracking. IEEE Trans. Image Process. 23 (4), 1639–1651.

Zhang, K., Zhang, L., Yang, M.-H., 2014a. Fast compressive tracking. IEEE Trans. Pattern Anal. Mach. Intell. 36 (10), 2002–2015.

Zhang, J., Ma, S., Sclaroff, S., 2014b. Meem: robust tracking via multiple experts using entropy minimization. In: European Conference on Computer Vision, Springer, pp. 188–203.

Zhang, K., Liu, Q., Wu, Y., Yang, M.-H., 2016. Robust visual tracking via convolutional networks without training. IEEE Trans. Image Process. 25 (4), 1779–1792.

Zhang, T., Xu, C., Yang, M.-H., 2017. Multi-task correlation particle filter for robust object tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1, 3.