FEATURE FUSION-BASED SIAMESE REGION PROPOSAL NETWORK FOR ULTRASOUND TRACKING

Xinglong Zhu, Ruirui Kang, Yifan Wang, Danni Ai,
Tianyu Fu and Jingfan Fan

Beijing Engineering Research Center of Mixed Reality and
Advanced Display, School of Optics and Photonics,
Beijing Institute of Technology, Beijing 100081, China

ABSTRACT

Object tracking based on ultrasound image navigation can effectively reduce damage to healthy tissues in radiotherapy. In this study, we propose a deep Siamese network based on feature fusion. Whilst adopting MobileNetV2 as the backbone, an unsupervised training strategy is introduced to enrich the volume of the samples. The region proposal network module is designed to predict the location of the target, and a non-maximum suppression-based post-processing algorithm is designed to refine the tracking results. Moreover, the proposed method is evaluated in the Challenge on Liver Ultrasound Tracking dataset and the self-collected dataset, which proves the need for the improvement and the effectiveness of the algorithm.

KEYWORDS

Ultrasound tracking, Siamese network, Respiratory motion estimation, One-shot learning

1. INTRODUCTION

Respiratory motion negatively affects radiotherapy for liver tumors. Doctors typically enlarge the radiation margin to ensure that the tumor receives adequate radiation. However, enlarging the radiation margin can harm surrounding tissues [[1]]. Generally, patients are instructed to hold their breath during radiation. As completion of the radiotherapy in one breath-holding period is impossible, doctors stop the treatment frequently and retarget the tumor with the radiation source at the start of a new round of radiation treatments [[2], [3], [4], [5]]. This approach is time-consuming and difficult. Implantation of invasive markers was also attempted, but invasive surgery causes additional damage to patients [[6], [7], [8]].

In recent years, ultrasound navigation was utilized to predict the location of tumors in real-time, in which the radioactive source is controlled to follow a tumor's movement [[9]]. However, the acoustic reflectivity of liver tumors is similar to that of surrounding tissues [[1]], making it difficult to locate the tumor directly based on ultrasound images. Other anatomical structures were used to predict the location of tumors. Among them, liver vessels have an acoustic reflectivity contrasting that of surrounding tissues; thus, liver vessels are typically chosen as targets for ultrasound tracking [[10], [11]].

Previously, matching or registration algorithms were typically employed to track liver vessels [[12], [13], [14]]. Researchers introduced Siamese networks to ultrasound tracking, as such

David C. Wyld et al. (Eds): CMLA, CIoT, CSEIT, NCS, NLPD, SPM - 2021
pp. 219-232, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111517

networks excel in visual object tracking. Liu et al. (2019) proposed the cascaded SiamFC algorithm and designed a two-stage cascaded Siamese network to improve the tracking accuracy of the network, thereby ranking first in the Challenge on Liver Ultrasound Tracking(CLUST) 2015 competition [[15]].

Recently, a network architecture similar to AlexNet was widely applied as the backbone of network[[15], [16], [17], [18]]. This fact inspires us to apply a highly sophisticated architecture to ultrasound tracking. However, two major obstacles exist in the application of a very deep network in ultrasound tracking. Firstly, the lack of annotated data makes training a general model difficult. Secondly, distractors confuse trackers [[15]]. As it shows in Figure 1, the distractors in the left image are more similar to the target in the right image than that in the left image, because the appearance of the target changes, thereby making tracking difficult.

To overcome the two aforementioned problems, an unsupervised training strategy is introduced to expand the volume of the samples. MobileNetV2 is adopted as the backbone of the SiamRPN-based tracker and the output feature of the backbone is fused for better discrimination. A post-processing algorithm based on non-maximum suppression (NMS) is proposed to eliminate distractors.

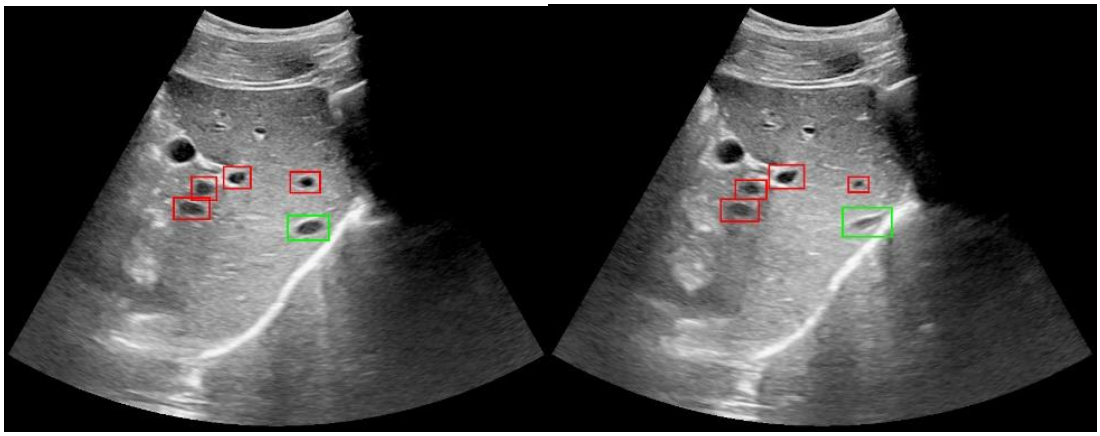

Figure 1. Two frames ofan ultrasound sequence, in which the green bounding boxes mark the targets, and the red bounding boxes mark the distractors.

In this work, we propose an original tracking algorithm based on feature fusion to solve the aforementioned problems. The contributions of this work are summarized below.

1) A network model based on feature fusionis proposed. Local features and semantic features are integrated to improve the discriminative ability of the algorithm.
2) A training strategy combining supervised and unsupervised methods is proposed. Unsupervised training methods can increase the volume of samples, thereby enabling the algorithm to learn substantial general features.
3) A post-processing algorithm based on NMS is proposed. Spatial information and temporal features are utilized to eliminate distractors, which can improve the accuracy and robustness of the tracker.

In the next section, the development of visual object tracking and ultrasound tracking is reviewed. The proposed method will be introduced in detail in section 3. The experiments and their results are reported in section 4. In section 5, the advantages and limitations are concluded.

2. RELATED WORKS

Visual target tracking is a classic computer vision problem. Tracking algorithms such as sparse coding [[19]], Kalman filters [[20]], mean shift [[21], [22]] and so on model the target then locate the most similar area in the search image. Algorithms using this tracking method are called generative algorithms. Such algorithms generally do not require training, but their performance depends on the parameters set empirically by the researcher. With the introduction of KCF [[23]], discriminative algorithms attracted researchers' attention. Discriminative algorithms focus on the difference between target and background. Compared with generative algorithms, discriminative algorithms pay attention to negative samples, which leads to better performance. With the increasing numbers of proposed datasets and benchmarks [[24], [25]], deep features based on statistics gradually replaced handcrafted features [[26], [27]]. Deep features are extracted by convolutional neural network (CNN), and weights of network are optimised based on a huge amount of data. Thus, deep features are more robust than handcrafted features. The combination of deep features with discriminative algorithms spawned many remarkable algorithms, such as MDNet [[28]], ECO [[27]], SiamFC [[29]] and SiamRPN++[[30]], which all achieved SOTA in competitions [[31], [32], [33]].

Ultrasound tracking algorithms combine the characteristics of ultrasound images and are affected by the development of object tracking algorithms for natural images.

A similar process can be seen in ultrasound tracking. Previously, tracking was generally considered as a registration or matching problem in ultrasound sequences [[1]]. Hallack et al. (2015) used LogDemons as a registration framework to solve the problem of target tracking [[12]]. Similarly, Shepard et al. (2017) employed image block matching to track a target [[13]]. Williamson et al. (2017) integrated dense optical flow, template matching, and image intensity information for hybrid tracking [[14]]. The aforementioned algorithms are training-free. However, as most matching and registration tasks are performed offline, such algorithms do not pay attention to real-time performance. Meanwhile, ultrasound tracking also draws on the development of visual object tracking. For example, in 2015, Kondo improved the KCF algorithm for ultrasound tracking [[34]]. Moreover, Shen et al. (2018) and Jeungyoon et al. (2019) adopted a CNN-like architecture to extract features and constructed correlation filters to process the features [[16], [17]]. Gomariz et al. (2018) added prior location information prediction to SiamFC [[18]]. Liu et al. (2019) proposed the cascaded SiamFC algorithm based onSiamFC, which won first place in the CLUST 2015 competition and has yet to be surpassed [[15]].

3. METHOD

The network structure proposed in this study is illustrated in Figure 2. The network uses MobileNetV2 [[35]] as the feature extractor. As the third-, fifth- and seventh-layer features outputted by the network have the same size, they can be stacked easily for feature fusion. Inspired by SiamRPN++ [[30]], a depth wise cross-correlation structure is adopted for the discrimination, and the two branches are designed for precise positioning. The difference between SiamRPN++ and the proposed network structure is that the stacked features are inputted directly into the two branches, which means that convolution layers are utilized to integrate the semantic and local features.

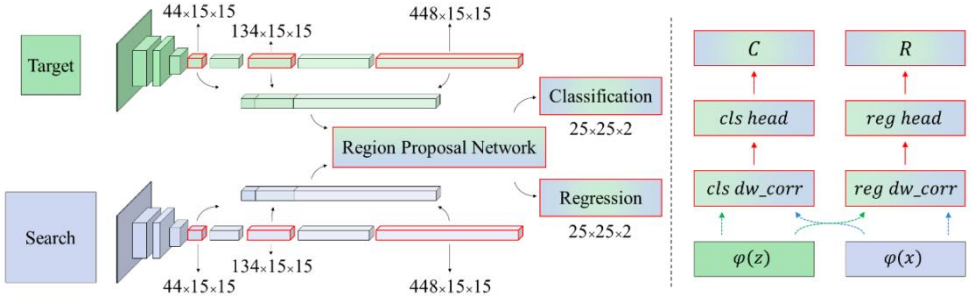

Figure 2.Theproposed network architecture; the left sideshows the network architecture; the right sideshows the classification and regression branches in RPN module.

In addition to the network structure, a post-processing algorithm is proposed based on NMS, which plays a positive role in suppressing distractors.

3.1. Network Architecture

The proposed network structure uses the pretrained MobileNetV2 as the feature extractor. MobileNetV2 employs deep separable convolutions to construct an inverted residual block that maps the high-dimensional image space to the low-dimensional feature space. This design demonstrates a satisfactory balance between performance and computational cost. Meanwhile, the last few inverted residual blocks of MobileNetV2 output tensors with the same scale, which provide convenience for the feature fusion.

Feature fusion of different depths was proven to be effective for tracking. The stacked features are integrated into the RPN. Compared with SiamRPN++, the adjust layers are removed from the RPN module in the proposed structure. Based on experiments, removal of the adjustment layer can prevent overfitting. The depth-wise correlation layer first convolves the stacked features with a 3×3 kernel to 256 channels, integrating the feature output to each layer. After the correlation, fully convolutional layers are built as the head modules. The two-branch head modules predict the position and score for each subregion.

3.2. Mixed Training Strategy

As the network outputs the classification and regression results, the loss function of the network must consider the output of the two branches, and the loss value of the classification branch adopts a cross-entropy form.

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)],$$

where L_{cls} represents the loss value of the classification branch of the network, y_i is the result marked at coordinate i and $\hat{y}_i$ is the predicted value of the classification branch.

The loss of the regression branch usesL_1 loss, as follows:

$$L_{reg} = -\frac{1}{N}\sum_{i=1}^{N}|r_i - \hat{r}_i|,$$

where L_{reg} represents the loss value of the classification branch of the network, y_i is the result marked at coordinate i, and $\widehat{y_i}$ is the predicted value of the classification branch.

Finally, the network loss can be written as follows:

$$L = L_{cls} + L_{reg}.$$

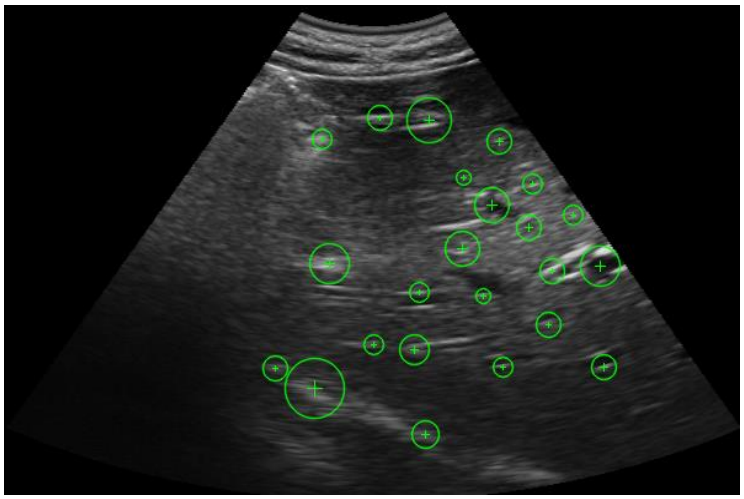

Figure 3. Data generated by the unsupervised strategy; the green crosses mark the key points extracted by SURF, and the green circles indicate the one-quarter size of the key points.

As a typical network using padding, the MobileNetV2 network does not have a shift-invariant characteristic. A large range of random shifts must be set during the training phase to prevent the network from collapsing into the center bias.

To increase the generalization ability of the network, unsupervised training is added to the previous training strategy. SURF algorithm is utilized to extract the key points from the ultrasound image then select the high response key points with large feature size and far from the ultrasound image boundary as the training sample. Figure 3 shows the key points extracted by an unsupervised strategy in an ultrasound image. During the training, the regions around the key points are cropped as target images and search images. The samples generated by the unsupervised algorithm and the samples manually labeled are mixed and added to the data loader. Without the unsupervised strategy, the tracker would propose objects likely to be vessels instead of objects likely to be the target. As the objects labeled in the dataset are nearly all vessels, the addition of the unsupervised strategy will prevent the algorithm from overfitting the labeled objects.

3.3. Tracking Inference Phase

To estimate the location of the target, the region of interest (ROI) is divided into 25×25 subregions, and the RPN outputs a score and a location for each subregion. The score represents the probability of the target appearing in the subregion, and the location indicates where the target is most likely to appear in the subregion.

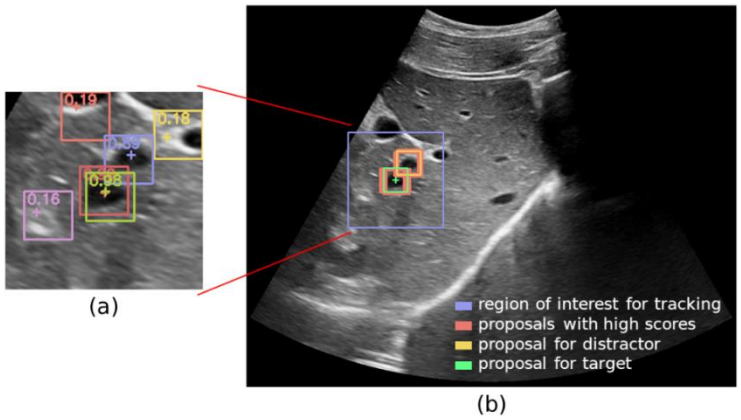

Figure 4. Tracking inference phase; image (a) shows several proposals generated by RPN; image (b) shows the post-processing algorithm dealing with the proposals.

Figure 4 shows the selection of proposals using NMS-based post-processing algorithms. Figure 4(a) shows several proposals generated by RPN. In Figure 4(a), the rectangles represent the subregions, the values annotated in the rectangles indicate the scores of the proposal, the crosses mark the proposed locations. The range and proposal of the same subregion are presented in the same color. Figure 4(b) shows the post-processing algorithm separates different proposals into targets, distractors, and redundant proposals that represent the same object. As there could be several proposals that represents the same objects, the key to improving the accuracy of the tracker is finding the proposal closest to the real target location among them. Inspired by the NMS algorithm, an appropriate algorithm for screening the proposals is designed. Firstly, the low-response proposals are excluded, which are not presented in Figure 4(b). Secondly, the proposals close to the proposal with a maximum response are filtered out, which are marked with red rectangles in Figure 4(b). Finally, the proposal closest to the tracking result of the previous frame is selected as the tracking result of the current frame, which is marked with a green rectangle in Figure 4(b). Those filtered out in the last step are marked in yellow rectangles in Figure 4(b). All rectangles in Figure 4(b) represent the location of the proposals.

NMS-based post-processing strategies can explicitly suppress distractors. When all the proposals have a low response, this strategy ensures that the tracker outputs an appropriate result. As the tracker will extract the ROI based on the result of the previous tracking frame, losing the target would be catastrophic for the subsequent tracking. The proposed method can effectively avoid this problem. Compared with the post-processing strategy based on the Hanning window, the proposed method is more robust.

4. EXPERIMENTS

The trained MobileNetV2 in SiamRPN++ is utilized as the initial weights of the backbone and fine-tune the network with the mixed training strategy mentioned in the previous section. The warm up learning rate is set to make the learning rate decay exponentially from 0.005 to 0.0005 during the training. In addition, the optimizer is SGD.

During the training process, a total of 20 epochs is performed. In the first 10 epochs, only the weights of the RPN are optimized. In the last 10 epochs, the last five inverted residual blocks and the RPN are optimized together. The sizes of the target images and search images are set to 127×127 and 255×255.

Following the standard of the CLUST 2015 and VOT 2015, the locations predicted by the tracker are compared with the ground truth. Criteria are calculated and plots are drawn below.

The proposed method is built with Python. And the experiments are implemented on a computer with an Intel i7 processor, 32 GB of RAM, and an Nvidia GTX 1080 graphic card.

4.1. Data

Experiments are performed on two datasets, which are, the published CLUST 2D Training Dataset and the self-collected dataset.

The CLUST dataset provides 24 2D ultrasound sequences with public annotations. The ultrasound sequences were acquired from patients during free breathing with various equipment, leading to sequences with different temporal and spatial resolutions. Approximately 10% to 13% of the frames in each sequence are annotated. Moreover, multiple targets may be labeled in a sequence. The dataset is annotated manually with the target location by three observers and verified by an additional observer. The ground truth of the dataset is the mean of the three manual annotations.

The self-collected dataset is acquired using a Philips scanner. The self-collected dataset consists of 10 sequences with temporal resolutions from 27 Hz to 30 Hz and spatial resolutions from $0.16\ \text{mm} \times 0.16\ \text{mm}$ to $0.27\ \text{mm} \times 0.27\ \text{mm}$. All the data are annotated following the method of CLUST dataset. When collecting ultrasound sequences for CLUST 2D Training Dataset, coughing and other emergencies sometimes cause discontinuities in the sequence. In the self-collected dataset, those discontinuous sequences are filtered out for better quality.

4.2. Evaluation Criteria

To evaluate the proposed method, two types of experiments are designed, which are, a cross-validation in the CLUST 2D Training Dataset and an evaluation in the self-collected dataset.

Specifically, a sixfold cross-validation in the CLUST 2D Training Dataset is performed. The dataset is divided into six groups. All the models are fine-tuned into five groups then evaluated in the remaining group. In the evaluation in the self-collected dataset, the model is fine-tuned first in the CLUST 2D Training Dataset.

To compare the various trackers comprehensively, two evaluation methods are designed. Following the criteria of the CLUST dataset, the trackers are evaluated in all the sequences with only one initialization [[1]]. Given annotations l_i and tracking results x_i for target i, tracking error E_i at time t is calculated as

$$E_i(t) = \|l_i - x_i\|,$$

where $\|\cdot\|$ represents the Euclidean distance. The tracking errors are summarised by the mean, standard deviation, and 95th percentile of the Euclidean distance for all the frames.

Inspired by the criteria of VOT 2015 [[31]], an evaluation experiment is designed using a different method. After initialization at the first frame, the tracker is reinitialized when it loses the target. The failures are counted to measure the robustness of the tracker. In addition, the average overlap between the predicted target bounding boxes and annotations is calculated, which is defined as accuracy.

The expected average overlap (EAO) is the average of the expected overlap in an interval $[N_l, N_h]$ of typical sequence lengths. The expected overlap of $N_s$is calculated by averaging the overlap in all available N_s-length sequences. To get the typical sequence length, the probability density function (PDF) of the sequence lengths is computed via kernel density estimation. Figure 5 presents the estimated PDF of the sequence lengths in the self-collected dataset. As it shows, the typical length is in the interval of $[121, 308]$. The probability of the sequence length being a typical length is 50%.

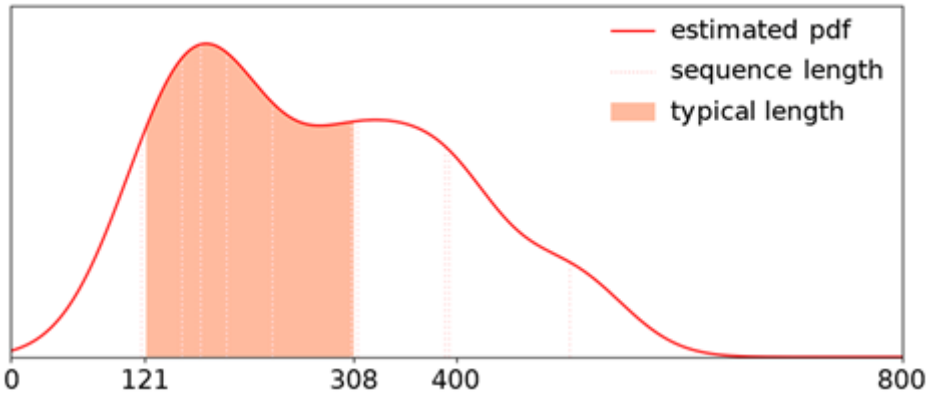

Figure 5. Estimated PDF of sequence lengths in the self-collected dataset; the sequence length in the dataset is marked by dotted lines.

4.3. Cross Validation in CLUST Dataset

A sixfold cross-validation is performed in the CLUST 2D Training Dataset to compare the performance of the proposed model with that of several representative methods. Figure 6 presents the success plots and precision plots of the proposed method and several representative methods, including SiamRPN++, SiamFC, DiMP18, PrDiMP18, and KYS, and shows that the proposed method generates superior results in terms of overlap success.

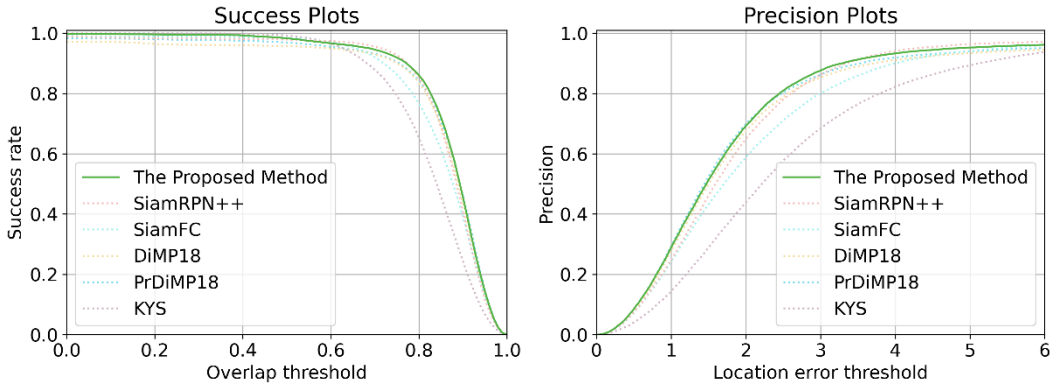

Figure 6. Success plots and precision plots of the proposed method and several representative methods. All the trackers are evaluated in the CLUST dataset via cross-validation.

Table 1 reports the mean, standard deviation, and 95th percentile of the tracking error of each tracker evaluated in the CLUST dataset via cross-validation. Table 2 shows the accuracy, failure, and EAO of each tracker evaluated in the CLUST dataset via cross-validation. The tables reveal that the proposed method obtains a minimal mean error and maximal accuracy.

Table 1. Mean error (Mean), standard deviation (Std) and 95th percentile (TE95th) of each tracker evaluated in CLUST dataset via cross validation

Tracker	Mean (mm)	Std (mm)	TE95th (mm)
The Proposed Method	**0.8582**	1.7042	1.9410
SiamRPN++	1.3622	4.7684	**1.7851**
SiamFC	1.4086	3.3518	2.5030
DiMP18	1.3864	3.8717	2.5193
PrDiMP18	1.5818	4.5768	2.7775
KYS	1.0856	**0.9283**	2.5882

Table 2. Accuracy, failure and EAO of each tracker evaluated in CLUST dataset via cross validation

Tracker	Accuracy	Failure	EAO
The Proposed Method	**0.8690**	7	0.8322
SiamRPN++	0.8655	8	0.8492
SiamFC	0.8479	14	0.8120
DiMP18	0.8449	**2**	**0.8541**
PrDiMP18	0.8564	27	0.7698
KYS	0.8230	**2**	0.8207

4.4. Evaluation of Trackers in Self-collected Dataset

The trackers are further evaluated by fine-tuning them in the CLUST dataset then evaluating them in the self-collected dataset. Figure 7 presents the success plots and precision plots of the proposed method and several representative methods, including SiamRPN++, SiamFC, DiMP18, PrDiMP18, and KYS, and shows that the proposed method demonstrates the best performance.

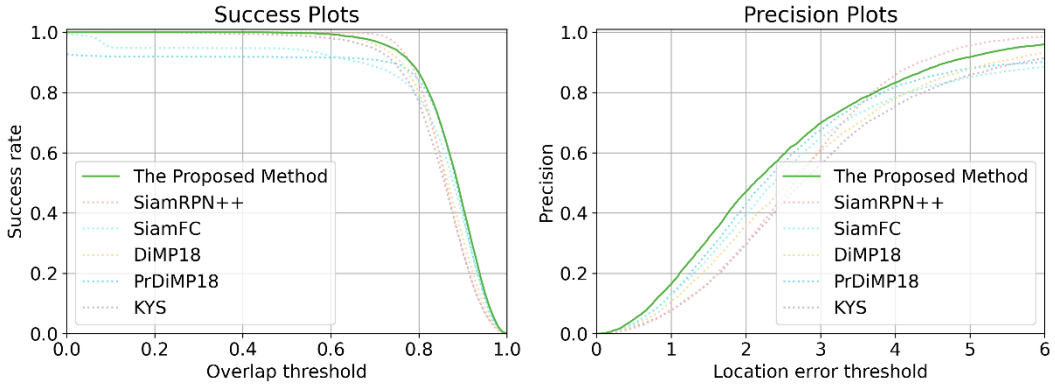

Figure 7. Success plots and precision plots of the proposed method and several representative methods. All the trackers are trained in the CLUST dataset and evaluated in the self-collected dataset.

Table 3 reports the mean, standard deviation, and 95th percentile of the tracking error of each tracker evaluated in the self-collected dataset. Table 4 shows the accuracy, failure, and EAO of each tracker evaluated in the self-collected dataset. Compared with the other methods, the proposed method obtains the best mean error, accuracy, and EAO.

Table 3. Mean error (Mean), standard deviation (Std) and 95th percentile (TE95th) of each tracker trained in the CLUST dataset and evaluated in the self-collected dataset

Tracker	Mean (mm)	Std (mm)	TE95th (mm)
The Proposed Method	**0.5745**	0.4369	1.4048
SiamRPN++	0.6298	**0.3503**	**1.2469**
SiamFC	1.1444	2.3535	8.0855
DiMP18	0.6591	0.4800	1.5631
PrDiMP18	0.8305	0.5736	1.9257
KYS	0.7258	0.5895	1.7611

Table 4. Accuracy, failure and EAO of each tracker trained in the CLUST dataset and evaluated in the self-collected dataset

Tracker	Accuracy	Failure	EAO
The Proposed Method	**0.8750**	**0**	**0.8782**
SiamRPN++	0.8597	**0**	0.8642
SiamFC	0.8218	2	0.8214
DiMP18	0.8567	**0**	0.8002
PrDiMP18	0.8357	51	0.3188
KYS	0.8445	**0**	0.8494

4.5. Visualisation

The tracking results of the proposed method and SiamRPN++ from the cross-validation in the CLUST dataset are visualized for a representative example. The tracking trail of the trackers is plotted and several frames are posted with tracking results in Figure 8.Frames 0570 to 0652 show a distractor approaching the target, leading SiamRPN++ to drift whilst the proposed method tracks steadily. As shown in Figure 8, the proposed method exhibits superior capability in distinguishing the target from the distractors, thereby benefitting from the NMS-based post-processing.

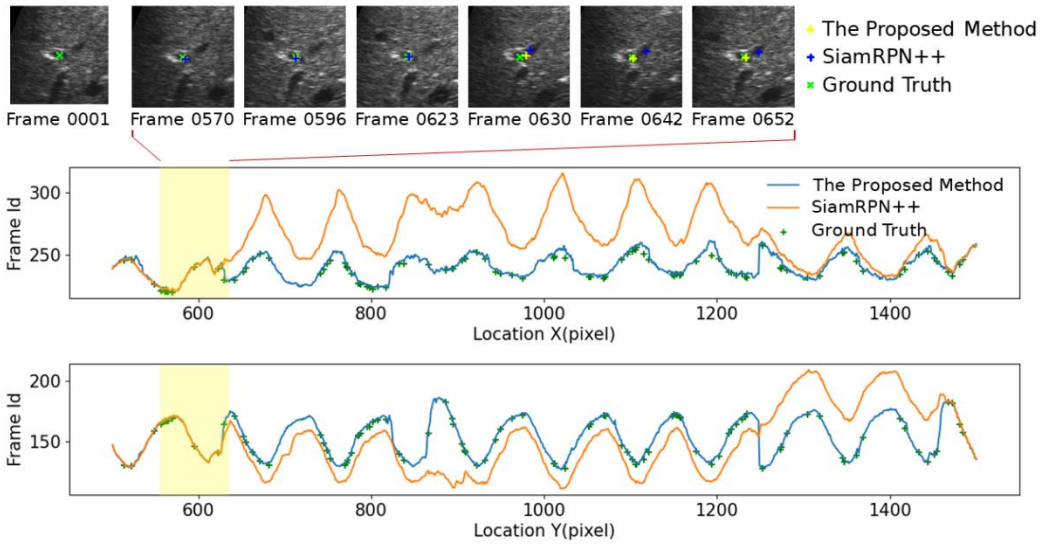

Figure 8. Tracking trail plot and tracking results of several frames.

Table 5 reveals the speed of the proposed method and several representative methods. The proposed method is the second fastest tracker among all the listed trackers and achieves real-time, with 62.12 FPS.

Table 5. Speed of proposed method and several representative methods

Tracker	Speed (FPS)
The Proposed Method	62.12
SiamRPN++	61.14
SiamFC	**105.03**
DiMP18	31.59
PrDiMP18	19.98
KYS	17.69

4.6. Ablation Study

In this section, the effects of different network factors are validated. Several trackers composed of different factors are evaluated using the same two methods employed in the previous experiments. Table 6 shows several criteria of each tracker evaluated via cross-validation. Table 7 presents several criteria of each tracker evaluated in the self-collected dataset. The tables reveal that the unsupervised strategy, feature fusion, and NMS post-processing contribute to the satisfactory performance of the proposed method.

Table 6: Network factors, accuracy and failure of several trackers evaluated in CLUST dataset via cross validation

Tracker	Unsupervised strategy	Feature fusion	Using NMS	Accuracy	Failure
The Proposed Method	√	√	√	0.8690	7
The Proposed Method (only trk)		√	√	0.8340	43
The Proposed Method (no fuse)	√		√	0.8655	8
The Proposed Method (no nms)	√	√		0.8472	24
SiamRPN++				0.8655	8

Table 7: Network factors, accuracy and failure of several trackers trained in CLUST dataset and evaluated in self-collected dataset

Tracker	Unsupervised strategy	Feature fusion	Using NMS	Accuracy	Failure
The Proposed Method	√	√	√	0.8750	0
The Proposed Method (only trk)		√	√	0.8041	0
The Proposed Method (no fuse)	√		√	0.8597	0
The Proposed Method (no nms)	√	√		0.8746	0
SiamRPN++				0.8597	0

5. CONCLUSIONS

In this paper, the obstacles for utilizing a highly sophisticated architecture in ultrasound tracking are analyzed. Firstly, an unsupervised training strategy is introduced to solve the problem of the lack of labeled data. Secondly, an RPN module is employed to predict the possible location of the target. Thirdly, feature fusion and NMS-based post-processing are proposed to improve the algorithm's robustness to distractors. Finally, an end-to-end network architecture is built with a

unique training strategy. Moreover, a large number of experiments are conducted based on the CLUST and the self-collected dataset, which proves our improvement of the performance of the algorithm.

This work provides a solution to the problem of applying very deep network structures to ultrasound tracking. In addition to the unsupervised training method that solves the lack of samples, other improvements are also proven to refine the accuracy of the algorithm. The proposed method gets accurate tracking results with a speed of 62.12 fps, which is surplus to ensure the real-time performance of the system.

For the lack of utilizing prior knowledge of ultrasound sequences, there is still much room to improve the proposed method. As our algorithm suppresses the distractors with the continuity of the sequence, the problem of distractors is not solved completely. And it also leads to dependence on the continuity of the sequence, which is difficult to guarantee during ultrasound acquisition. In the future, we will combine the characteristics of ultrasound images and the temporal and spatial characteristics of respiratory motion to further improve the accuracy and robustness of the algorithm.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation Program of China (81627803, 81871374)

REFERENCES

[1] De Luca V, Banerjee J, Hallack A, et al. Evaluation of 2D and 3D ultrasound tracking algorithms and impact on ultrasound-guided liver radiotherapy margins[J]. Medical physics, 2018, 45(11): 4986-5003.
[2] Boda-Heggemann J, Knopf A C, Simeonova-Chergou A, et al. Deep inspiration breath hold—based radiation therapy: a clinical review [J]. International Journal of Radiation Oncology* Biology* Physics, 2016, 94(3): 478-492.
[3] Péguret N, Ozsahin M, Zeverino M, et al. Apnea-like suppression of respiratory motion: First evaluation in radiotherapy[J]. Radiotherapy and Oncology, 2016, 118(2): 220-226.
[4] Parkes M J, Green S, Stevens A M, et al. Safely prolonging single breath-holds to> 5 min in patients with cancer; feasibility and applications for radiotherapy[J]. The British journal of radiology, 2016, 89(1063): 20160194.
[5] Parkes M J. Breath-holding and its breakpoint [J]. Experimental physiology, 2006, 91(1): 1-15.
[6] Takao S, Miyamoto N, Matsuura T, et al. Intrafractional baseline shift or drift of lung tumor motion during gated radiation therapy with a real-time tumor-tracking system [J]. International Journal of Radiation Oncology* Biology* Physics, 2016, 94(1): 172-180.
[7] Hunt M A, Sonnick M, Pham H, et al. Simultaneous MV-kV imaging for intrafractional motion management during volumetric-modulated arc therapy delivery[J]. Journal of applied clinical medical physics, 2016, 17(2): 473-486.
[8] Iwata H, Ishikura S, Murai T, et al. A phase I/II study on stereotactic body radiotherapy with real-time tumor tracking using CyberKnife based on the Monte Carlo algorithm for lung tumors[J]. International journal of clinical oncology, 2017, 22(4): 706-714.
[9] Şen H T, Bell M A L, Zhang Y, et al. System integration and in vivo testing of a robot for ultrasound guidance and monitoring during radiotherapy[J]. IEEE Transactions on Biomedical Engineering, 2016, 64(7): 1608-1618.
[10] De Luca V, Benz T, Kondo S, et al. The 2014 liver ultrasound tracking benchmark[J]. Physics in Medicine & Biology, 2015, 60(14): 5571.
[11] De Luca V, Székely G, Tanner C. Estimation of large-scale organ motion in B-mode ultrasound image sequences: a survey [J]. Ultrasound in medicine & biology, 2015, 41(12): 3044-3062.

[12] Hallack A, Papiez B W, Cifor A, et al. Robust liver ultrasound tracking using dense distinctive image features[J]. MICCAI 2015 Challenge on Liver Ultrasound Tracking, 2015: 28-35.
[13] Shepard A J, Wang B, Foo T K F, et al. A block matching based approach with multiple simultaneous templates for the real-time 2D ultrasound tracking of liver vessels [J]. Medical physics, 2017, 44(11): 5889-5900.
[14] Williamson T, Cheung W, Roberts S K, et al. Ultrasound-based liver tracking utilizing a hybrid template/optical flow approach [J]. International journal of computer assisted radiology and surgery, 2018, 13(10): 1605-1615.
[15] Liu F, Liu D, Tian J, et al. Cascaded one-shot deformable convolutional neural networks: Developing a deep learning model for respiratory motion estimation in ultrasound sequences [J]. Medical Image Analysis, 2020, 65: 101793.
[16] Shen C, Shi H, Sun T, et al. An Online Learning Approach for Robust Motion Tracking in Liver Ultrasound Sequence[C]//Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, Cham, 2018: 440-451.
[17] Jeungyoon, L., Euisuk, C., Tai-Kyong, S., Combination of RCNN and KCF for Landmark Tracking in 2D Ultrasound Sequence of Liver. IEEE Engineering in Medicine & Biology Society,
[18] Gomariz A, Li W, Ozkan E, et al. Siamese networks with location prior for landmark tracking in liver ultrasound sequences[C]//2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE, 2019: 1757-1760.
[19] Zhang S, Yao H, Sun X, et al. Sparse coding based visual tracking: Review and experimental comparison [J]. Pattern Recognition, 2013, 46(7): 1772-1788.
[20] Xu S, Chang A. Robust object tracking using Kalman filters with dynamic covariance [J]. Cornell University, 2014: 1-5.
[21] Zeng H, Chen J, Cui X, et al. Quad binary pattern and its application in mean-shift tracking[J]. Neurocomputing, 2016, 217: 3-10.
[22] Vojir T, Noskova J, Matas J. Robust scale-adaptive mean-shift for tracking[J]. Pattern Recognition Letters, 2014, 49: 250-258.
[23] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 37(3): 583-596.
[24] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2013: 2411-2418.
[25] LIRIS F. The Visual Object Tracking VOT2014 challenge results[J].
[26] Danelljan M, Robinson A, Khan F S, et al. Beyond correlation filters: Learning continuous convolution operators for visual tracking[C]//European conference on computer vision. Springer, Cham, 2016: 472-488.
[27] Danelljan M, Bhat G, Shahbaz Khan F, et al. Eco: Efficient convolution operators for tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 6638-6646.
[28] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 4293-4302.
[29] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking[C]//European conference on computer vision. Springer, Cham, 2016: 850-865.
[30] Li B, Wu W, Wang Q, et al. Siamrpn++: Evolution of siamese visual tracking with very deep networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4282-4291.
[31] Kristan M, Matas J, Leonardis A, et al. The visual object tracking vot2015 challenge results[C]//Proceedings of the IEEE international conference on computer vision workshops. 2015: 1-23.
[32] Kristan M, Leonardis A, Matas J, et al. The visual object tracking vot2017 challenge results[C]//Proceedings of the IEEE international conference on computer vision workshops. 2017: 1949-1972.
[33] Kristan M, Matas J, Leonardis A, et al. The seventh visual object tracking vot2019 challenge results[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019: 0-0.
[34] Kondo S. Liver ultrasound tracking using kernelized correlation filter with adaptive window size selection[C]//MICCAI workshop: challenge on liver ultrasound tracking. 2015: 13-19.

[35] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.