An Upgraded Siamese Neural Network for Motion Tracking in Ultrasound Image Sequences

Skanda Bharadwaj, *Member, IEEE*, Sumukha Prasad, *Graduate Student Member, IEEE*, and Mohamed Almekkawy, *Member, IEEE*

***Abstract*— Deep learning is heavily being borrowed to solve problems in medical imaging applications, and Siamese neural networks are the front runners of motion tracking. In this article, we propose to upgrade one such Siamese architecture-based neural network for robust and accurate landmark tracking in ultrasound images to improve the quality of image-guided radiation therapy. Although several researchers have improved the Siamese architecture-based networks with sophisticated detection modules and by incorporating transfer learning, the inherent assumptions of the constant position model and the missing motion model remain unaddressed limitations. In our proposed model, we overcome these limitations by introducing two modules into the original architecture. We employ a reference template update to resolve the constant position model and a linear Kalman filter (LKF) to address the missing motion model. Moreover, we demonstrate that the proposed architecture provides promising results without transfer learning. The proposed model was submitted to an open challenge organized by MICCAI and was evaluated exhaustively on the Liver US Tracking (CLUST) 2D dataset. Experimental results proved that the proposed model tracked the landmarks with promising accuracy. Furthermore, we also induced synthetic occlusions to perform a qualitative analysis of the proposed approach. The evaluations were performed on the training set of the CLUST 2D dataset. The proposed method outperformed the original Siamese architecture by a significant margin.**

***Index Terms*— Convolutional neural networks (CNNs), Kalman filter, Siamese networks, speckle tracking, ultrasound (US) image sequences.**

I. Introduction

MOTION tracking has wide range of applications in diagnostic ultrasound (US). It is used in several techniques, such as elasticity imaging [1], blood flow imaging [2]–[4], elastography [5], photoacoustic speckle tracking [6], phase-aberration correction [7], [8], and echocaridography [9]. Due to its wide range of applications, different motion tracking methods have been proposed [10]. They can be broadly classified as frequency-domain techniques [11], time-domain techniques [12]–[14], and optical flow-based techniques [15].

Speckles formed in US images have been commonly used for 2-D time-domain-based tissue motion tracking. Speckles, in US images, are formed by the combination of constructive and destructive interference of echoes from scatterers in the observed tissue. Speckle tracking was first used by Trahey *et al.* [3] for blood velocity imaging that employed correlation-based motion tracking technique, which was angle-independent. Combining signal correlation with local smoothness as prior information to improve displacement estimation motivates the development of the probabilistic algorithm, such as predictive search and Bayesian speckle tracking [13], [14]. Predictive search algorithms utilize the available prior information and then use certain predictive strategies to advance the estimation process. Chen *et al.* [12] proposed a predictive search strategy called "quality-guided" tracking, where an initial brute-force search is employed to obtain the prior information and then use a recursive search until all estimation locations are completed. On the other hand, McCormic *et al.* [13] first proposed to use the concept of Bayes theorem to regularize speckle tracking. It was an iterative algorithm where displacement estimation at one location can be gradually improved by taking information from its neighbors. Byram *et al.* [14] developed a similar but more general Bayesian framework for speckle tracking. One of their main contributions was the improvement of the discriminant ability by appropriately scaling the maximum likelihood function.

Furthermore, Ebbini [16] proposed phase coupled 2D speckle tracking algorithm, which couples the phase and magnitude gradients near the correlation peak to determine its coordinates with subsample accuracy in both axial and lateral directions. To overcome the limitations of relatively coarse lateral sampling, Almekkawy and Ebbini [17] proposed a multidimensional speckle tracking with subsample accuracy. To further improve the tracking accuracy, Rebholz *et al.* [18] proposed a 2-D iterative projection (TDIP) algorithm using the Riesz transform. The TDIP method performs iterative projections and uses the aggregate of these projected locations to estimate the motion. In [19], synthetic lateral phase was used to overcome the inherent limitation of speckle tracking, where lateral displacement estimates are much less accurate than axial displacement estimates. Other approaches include optical

Manuscript received June 5, 2021; accepted July 4, 2021. Date of publication July 7, 2021; date of current version November 23, 2021. *(Corresponding author: Skanda Bharadwaj.)*

The authors are with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: ssb248@psu.edu; sub1206@psu.edu; mka9@psu.edu).

Digital Object Identifier 10.1109/TUFFC.2021.3095299

flow-based tracking [15], [20], [21], kernel-based mean-shift tracking [22], and correlation-based tracking [23], [24]. As a result of contributions from various groups, speckle tracking has been used in various techniques, such as measurement of ventricular torsion [25], quantifying tendon displacement [26], elastography [27], and echocardiography [28], [29].

Among the 2-D time-domain motion tracking techniques, one of the most commonly used algorithms for tracking in US images is block matching [30]. Researchers have worked on different applications of block matching including tracking of carotid artery wall motion [31], [32], subsample displacement estimation using Kriging interpolation [33], shear strain and motion amplitude within the arterial wall [34], and study of motion dynamics of carotid atheromatous plaque [35]. In any typical block matching-based tracking, a reference block (window/kernel) is defined in the first frame and is tracked in the subsequent frames. Any block in the subsequent frame that is subject to search is called the candidate block. The similarity between the reference block and the candidate block is quantified by defining a cost function. The best match is obtained by choosing a match with either minimum or maximum value of the cost function. Cross correlation and normalized cross correlation (NCC) are common cost functions used in motion tracking [36], [37]. The search algorithm to improve processing speed in the context of block-matching has also been proposed [38]. However, recent advancements in deep learning have proven to be extremely effective for similarity matching-based motion tracking.

Deep learning techniques have gained significant attention toward a number of imaging tasks, such as object recognition [39], visual object tracking [40], and image segmentation [41]. Convolutional neural networks (CNNs) are slowly being borrowed for medical imaging including US imaging [42]. Notably, CNNs are being extensively used for beamforming [43]–[45], US image segmentation [46], and image reconstruction [47], [48]. Deep learning has also been applied to motion estimation in US images. Dosovitskiy *et al.* [49] proposed a deep learning model called Flownet for end-to-end motion estimation. Peng *et al.* [27] proved that an updated model called Flownet2.0 was feasible for speckle tracking-based strain elastography. Based on the Flownet2.0 architecture, Kibria and Rivaz [50] applied CNN, called global US elastography network (GLUENet), to address the decorrelation in estimating displacement field. Furthermore, Tehrani and Rivaz [51] also applied deep learning for displacement estimation in US elastography. All the aforementioned deep learning models formulate optical flow estimation as a learning problem. Siamese architectures have also been explored to estimate motion in US images [52], [53].

In this article, we propose to adopt a Siamese network to perform motion tracking in US images. Specifically, we adopt the fully convolutional Siamese network (SiamFC) proposed by Bertinetto *et al.* [40]. In our associated conference paper [54] and presentations [55], [56], we proved that the Siamese network-based deep learning model could be used to track regions of interest (ROIs) in US image sequences. Despite the remarkable efficiency of SiamFC, it suffers two major limitations. First, it considers a constant position model and, therefore, is extremely sensitive to the deformation of the reference object. Second, it is a detection-based tracker and does not consider any motion model. Consequently, tracking of the reference object is solely dependent on the detections obtained in each frame. In order to overcome these limitations, an upgraded version of the SiamFC is proposed in this article. The two limitations of SiamFC are overcome with the introduction of two new modules into the original architecture. First, the template update module is introduced into SiamFC making it robust to the deformation of the reference object. This resolves the issue of the constant position model. Second, a linear motion model is introduced to the original architecture to address the issue of the missing motion model. The prediction from the linear motion model and the detections (measurement) from the SiamFC are combined using a linear Kalman filter (LKF) (henceforth, the second module is simply referred to as the LKF).

Evaluation of the proposed model was performed by tracking landmarks in US images acquired during image-guided radiation therapy. The motion caused due to the patient's respiration negatively affects image-guided therapy. Forcing patients to hold their breath is often unreliable [57]. Furthermore, challenging scenarios posed by the US images complicate the tracking procedure, as represented in Fig. 1. The scenarios include low signal-to-noise-ratio images, poor foreground–background distinction, multiple landmarks with similar appearance, and shadowing caused by airflow. Thus, robust and accurate tracking is of vital importance in image-guided radiation therapy. The performance of the proposed model was submitted to the MICCAI CLUST challenge [57], [58] for evaluation and was compared with the existing state-of-the-art method [52] and several other approaches [53], [59]–[66]. It should be noted that our model does not incorporate transfer learning. That is, the network is not trained with US images, unlike the other CNN-based methods that entered the challenge. Network weights from the original architecture are retained. The proposed model is a class-agnostic tracker and can be used in motion tracking of any landmark based on the user's choice.

II. Materials and Methods

A. Fully Convolutional Siamese Neural Network

In this article, we adopt the SiamFCs developed by Bertinetto *et al.* [40], which is briefly explained in this section. Siamese networks or twin networks are two artificial networks working in parallel for two different input vectors (in our case, images). Both networks apply identical transformation (use same weights) on both inputs to perform similarity matching on the two input images. Fig. 2 represents a pair of Siamese networks. Motivation for the use of Siamese networks arises from the scarcity of available labeled data for learning. This problem is typically called one-shot learning [67]. Facial recognition [68] best explains the need for one-shot learning. Collecting several labeled data to classify every face is absurd and infeasible. Therefore, the problem is to recognize another instance of the same class with just one available labeled data. In the Siamese architecture, the same network is used in

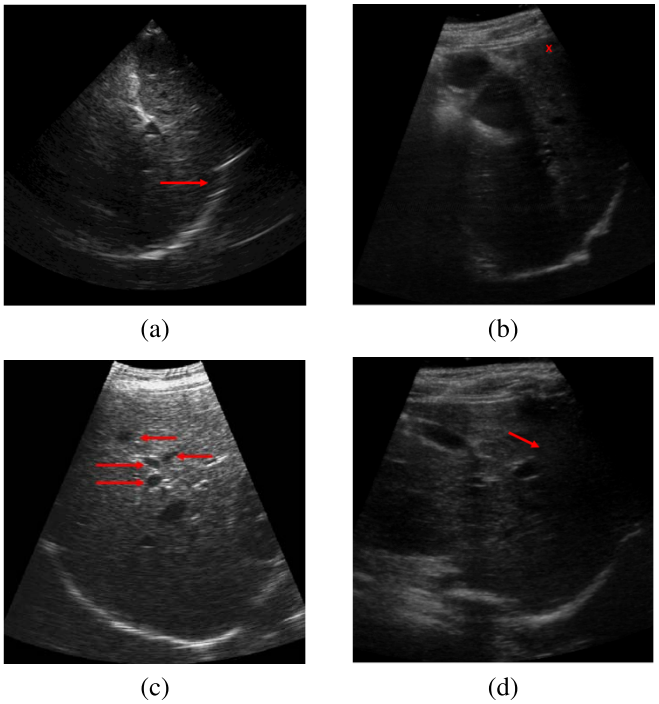


Fig. 1. Sample frames from different image sequences of CLUST 2D dataset that pose challenging situations for accurate landmark tracking. (a) Image with low signal-to-noise ratio. (b) Image in which the landmark marked in red "x" is almost indistinguishable form the background. (c) Landmarks that have very similar appearance models. (d) Air-shadowing.

parallel to generate feature maps of two input images and learn a distance function (represented by the *Merge* block in Fig. 2) between the two feature representations.

In the context of motion estimation, Siamese architectures formulate the problem as convolutional feature cross correlation between the reference block (usually defined as the ROI in the first frame of the image sequence) and the candidate blocks within a predefined search region. As mentioned earlier, similarity matching in US images is performed mostly using block-matching techniques, where a cost function, such as mean absolute difference (MAD), mean squared error (MSE), or NCC, is used to calculate the similarity between the reference block and the candidate block in an iterative fashion. The predefined search region is searched exhaustively to obtain the best-matching candidate block. Consequently, the cost function is calculated in every iteration, making the method computationally expensive.

Bertinetto *et al.* [40] employ a deep learning approach to solve the similarity learning problem. A deep CNN is trained in an initial off-line phase to solve the similarity learning problem. During tracking, the CNN is simply evaluated to obtain the best-matching candidate block. Specifically, a Siamese neural network is trained to locate the reference block within a larger search region of the subsequent frame. Siamese networks are advantageous in which they use pretrained networks and can work as class-agnostic trackers. Deep learning architectures without pretrained networks (i.e., the networks that learn online) can only learn data that are derived exclusively

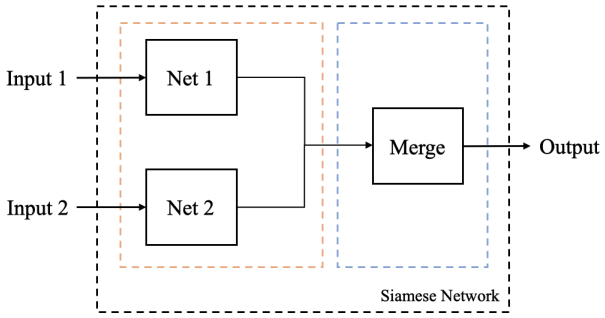


Fig. 2. Schematic of the Siamese network. The Siamese network takes two vectors (images) as inputs and applies identical transformation to both inputs. The networks *Net 1* and *Net 2* are identical. *Net 1* is applied on the labeled data *Input 1*, and *Net 2* is applied on the candidate *Input 2*. A *Merge* layer is then used to combine the two representations.

from the input video alone. This limits the network from learning more advanced models. In addition, such methods will apply stochastic gradient descent to update their model and fail to operate in real time.

In SiamFC, a function $f(z, x)$ is a Siamese architecture-based deep CNN, which learns to compare the reference block z and the candidate block x. A high score is returned if both blocks represent the same (or similar) object, and a low score otherwise. This function is applied to all possible candidate blocks within a predefined search region. Since the architecture is fully convolutional (a network without any dense layers), it allows us to input a larger search region and compute the similarity for all translated candidate blocks in a single evaluation. The two networks—*Net 1* and *Net 2*—shown in Fig. 2 represent the same CNN that resembles the architecture proposed by Krizhevsky *et al.* [39]. Consequently, identical transformation (φ) is applied on both inputs *Input 1* and *Input 2* (in our case, z and x, respectively), as shown in Fig. 2. Their representations are then combined using another function g, as shown in the following equation:

$$f(z, x) = g(\varphi(z), \varphi(x)) \tag{1}$$

where g can simply be a distance or a similarity metric. Specifically, in SiamFC, the output feature maps obtained by applying the transformation φ on both inputs are combined using a cross correlation layer. Since the search region is larger than z, the output of this network is a score map corresponding to the number of candidate blocks within the search region. Fig. 3 represents the detailed architecture of SiamFC.

As mentioned earlier, training of SiamFC in [40] was done in an initial off-line phase using the dataset of annotated videos. It should be noted that the network was trained using the ILSVRC dataset [69] that consists of camera images of real-world objects, such as animals, vehicles, and household items, and the network was not retrained using US images for our application. Furthermore, SiamFC does not update the template model or incorporate any motion model. Although researchers have improved the existing architecture using transfer learning with sophisticated models [52], [53], the aforementioned limitations remain unaddressed.

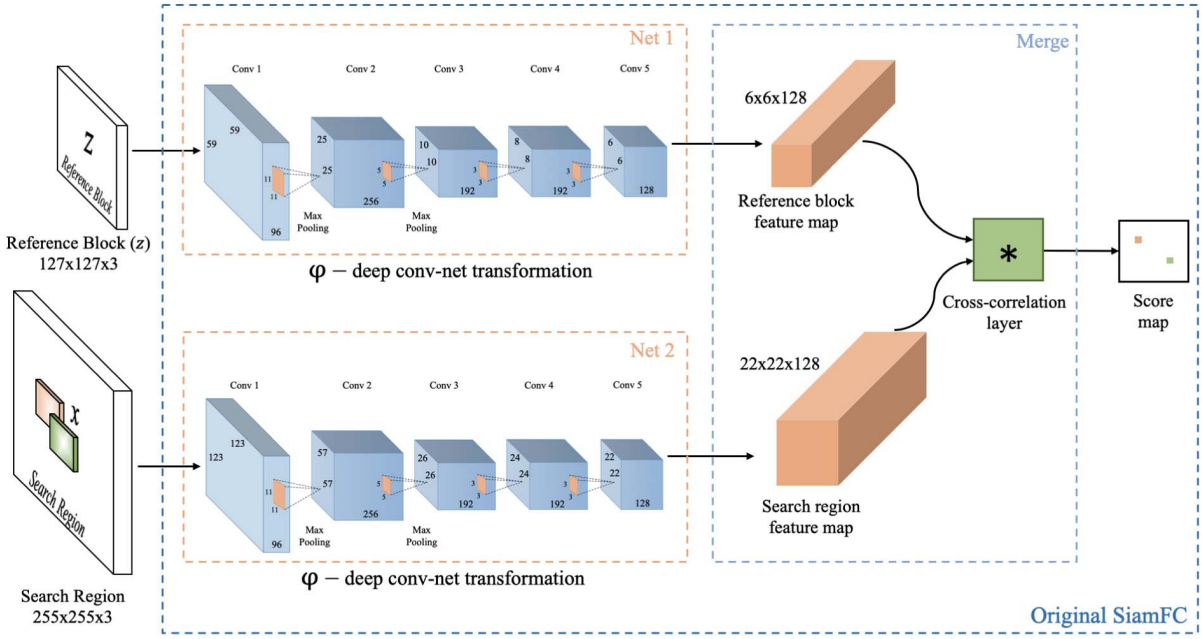


Fig. 3. SiamFC proposed in [40]. The reference block z and the search region centered around z with candidate blocks x are *Input 1* and *Input 2*, respectively. The convolutional stage resembling the architecture proposed in [39] forms *Net 1* and *Net 2*. The fully convolutional property of the architecture allows inputting images of different sizes. Finally, the cross correlation layer is used to merge the two representations in SiamFC. The orange and green pixels in the score map represent the correlation score of the two translated subwindows in the search region. The candidate block with the highest correlation score is selected as the tracked output.

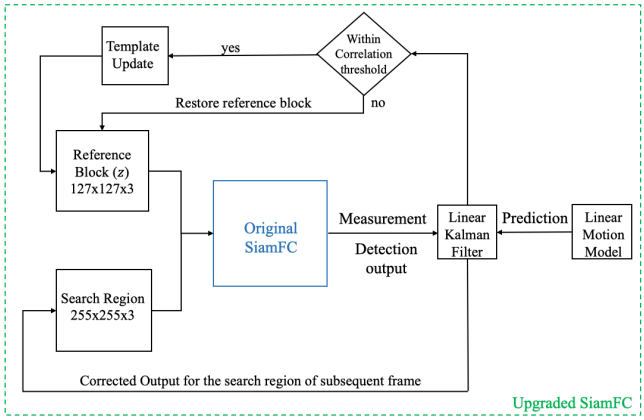


Fig. 4. Schematic of the upgraded SiamFC. Considering detection from SiamFC as the measurement and the output of the linear motion model as the prediction, LKF generates the corrected outputs. If the corrected output is within the permissible correlation threshold, the reference block is updated. Otherwise, the reference block is restored from the most recent frame.

In this article, we emphasize that addressing these issues is of vital importance. Two major contributions of this article are given as follows: 1) we upgrade the underlying tracker by incorporating the template update module and 2) we introduce a motion model to predict the motion of the ROIs. These two modules are explained in Sections II-B and II-C.

B. Template Update

SiamFC is a detector-based tracker and, hence, sensitive to deformations. In SiamFC, the features extracted from the reference block are compared with the features of candidate blocks of subsequent frames. It then returns a high score if the two images are matched or a low score otherwise. As mentioned earlier, φ is used to extract the features of the reference block. This becomes problematic when there is a change in the appearance of the reference block due to artifacts, such as speckle decorrelation, out-of-plane motion, or human errors in handling US probes. To overcome this limitation, a go-to solution is to update the template in every frame. Then, the template update module adapts the tracker to changes in the structure of the reference object every frame. However, updating the template every frame becomes a problem in cases such as out-of-plane motion. For instance, if we update the template when the reference object is out of the plane, tracking fails miserably in the subsequent frames. In order to avoid this, we propose to perform anchoring of the reference object. Anchoring is the process of updating the reference object only within a permissible threshold of change. It is analogous to anchoring an object (e.g., a ship) with an anchor, where the object is allowed to move within a permissible radius equal to the length of the anchor chain. Similarly, we allow updation of the template by thresholding based on a correlation score between reference block from the previous $[(n-1)\text{th}]$ frame and the best-matching candidate block from the current (nth) frame. Let z and x^k represent the reference block and the best-matched candidate block within the search region x, respectively. Then, the correlation coefficient G is calculated using the following equation:

$$G = \frac{\sum_i \sum_j \left(z_{ij} - \bar{z}\right)\left(x_{ij}^k - \bar{x}^k\right)}{\sqrt{\sum_i \sum_j \left(z_{ij} - \bar{z}\right)^2}\sqrt{\sum_i \sum_j \left(x_{ij}^k - \bar{x}^k\right)^2}}. \quad (2)$$

C. Linear Kalman Filter

In order to address the missing motion model, we propose to introduce a simple LKF. In the proposed design, we use (3) and (4) to model the pixel coordinates. In (3), **s** represents the displacement of the pixel in the Cartesian grid. Their location is represented by its coordinates (x, y) for lateral and axial positions, respectively. Similarly, in (4), **v** represents the velocity of the pixel represented by $(\dot{x}, \dot{y})$ in lateral and axial directions, respectively. The position and velocities of the pixel in both directions are considered to be the states of the system represented as $\mathbf{x}_k = [x, y, \dot{x}, \dot{y}]^T$. Also, **a** and dt represent acceleration and time step, respectively, in both equations

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \mathbf{v}_{k-1}\,\mathrm{dt} + \frac{1}{2}\mathbf{a}\,\mathrm{dt}^2 \tag{3}$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{a}\,\mathrm{dt}. \tag{4}$$

The above motion model is implemented using an LKF. We model the state and measurement equations of the LKF as follows:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{n}_k \tag{5}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k. \tag{6}$$

In (5), $\mathbf{x}_k$ represents the current state, **A** and **B** represent the state transition matrix and the control matrix, respectively, and $\mathbf{u}_k$ and $\mathbf{n}_k$ represent the control vector and the process noise, respectively. In (6), $\mathbf{z}_k$ represents the measurement of the true state $\mathbf{x}_k$. **H** represents the measurement model, and $\mathbf{w}_k$ represents the measurement noise. The subscript k represents the time instance.

Kalman filtering is carried out in two phases: prediction phase and update phase. The two phases are briefly described as follows.

1) Prediction Phase: In this phase, the priori state estimate and the priori state covariance estimate are calculated represented by (7) and (8). In (7), $\hat{\mathbf{x}}_k^-$ represents the *a priori* state estimate, $\mathbf{P}^-{}_k$ represents the *a priori* state covariance at time k, and **Q** represents the process covariance matrix

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \tag{7}$$

$$\mathbf{P}^-{}_k = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^{\mathbf{T}} + \mathbf{Q}. \tag{8}$$

2) Update Phase: In this phase, the error between measurement and prediction is appropriately weighted by calculating the Kalman gain. With the available Kalman gain and the new measurement, the posterior estimate of the state $\hat{\mathbf{x}}_k$ is calculated. Finally, we update the posterior estimate of the state covariance $\mathbf{P}_k$

$$\mathbf{K} = \mathbf{P}_k^-\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{C}\right)^{-1} \tag{9}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}\left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right) \tag{10}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_k^-. \tag{11}$$

In (9), **K** represents the Kalman gain, and **C** represents the measurement covariance matrix. $\hat{\mathbf{x}}_k$ in (10) represents the posterior state estimate, and $\mathbf{P}_k$ in (13) represents the posterior state covariance matrix.

From the aforementioned design, the state transition matrix **A** and the control matrix **B** are modeled as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \mathrm{dt} & 0 \\ 0 & 1 & 0 & \mathrm{dt} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{B} = \begin{bmatrix} \mathrm{dt}^2/2 & 0 \\ 0 & \mathrm{dt}^2/2 \\ \mathrm{dt} & 0 \\ 0 & \mathrm{dt} \end{bmatrix}. \tag{12}$$

Finally, the process covariance matrix (**Q**) is obtained by calculating $\mathbf{Q} = \mathbf{B}\mathbf{B}^{\mathbf{T}}$ resulting in the model shown as follows:

$$\mathbf{Q} = \begin{bmatrix} \frac{\mathrm{dt}^4}{4} & 0 & \frac{\mathrm{dt}^3}{2} & 0 \\ 0 & \frac{\mathrm{dt}^4}{4} & 0 & \frac{\mathrm{dt}^3}{2} \\ \frac{\mathrm{dt}^3}{2} & 0 & \mathrm{dt}^2 & 0 \\ 0 & \frac{\mathrm{dt}^3}{2} & 0 & \mathrm{dt}^2 \end{bmatrix}. \tag{13}$$

3) Upgraded SiamFC: Putting it all together, Fig. 4 represents the upgraded SiamFC with template update and LKF modules introduced into the underlying architecture. In summary, a reference block is defined in the first frame of the video sequence and is input to SiamFC. It then selects the best-matching candidate block in the subsequent frame, which is input to the LKF as measurement. Meanwhile, coordinates of the centroid of the reference block are predicted based on the linear prediction model presented in the previous subsection. LKF, with prediction and measurements, outputs the corrected coordinates. If this output is within the correlation threshold compared to the reference block of the previous frame, the reference block is updated. Otherwise, the reference block is restored from the most recent frame.

III. Experiments

A. Data: CLUST 2D Dataset

The proposed model was evaluated on a publicly available open dataset CLUST 2D [70]–[75]. This dataset contains 63 2-D US image sequences of liver acquired from healthy volunteers under free-breathing. The dataset contains data provided by three groups, the Biomedical Imaging Research Laboratory of CREATIS INSA, Lyon, France (CIL); the Computer Vision Laboratory, ETH Zürich, Zürich, Switzerland (ETH); and mediri GmbH, Heidelberg, Germany (MED). A wide range of US equipment, including five US scanners and six types of transducers, was used to collect the data. Each image sequence ranges from 4 s to about 10 min. The temporal resolution ranges from 6 to 31 Hz. The spatial resolution of the images ranges from 0.27 mm × 0.27 mm to 0.77 mm × 0.77 mm. About 38% (24/63) of the image sequences were made available with annotations for multiple frames as the training set, and the remaining 62% (39/63) of the image sequences were released as the testing set, in which annotations were only available for the first frame. At most four landmarks were annotated per image sequence. Although multiple landmarks are provided for a single image sequence, the challenge requires only single-object tracking at a time instead of multiobject tracking. A total of 53 landmarks were annotated in the training set, and a total of 85 landmarks were annotated in

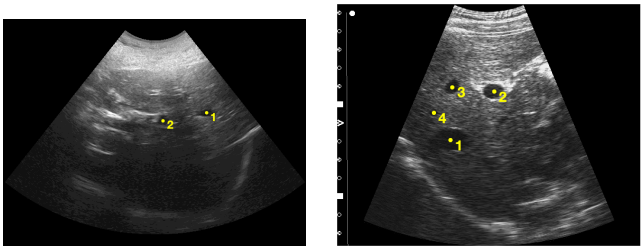


Fig. 5. Sample frames with annotations provided by the CLUST organizers. Left: first frame from the image sequence *CIL-01* that belongs to the training set that contains two landmarks. Right: first frame from the image sequence *MED-07-4* that belongs to the testing set, which contains four landmarks. The annotations are numbered and represented as yellow dots.

the testing set. In addition, about 10% of the images from the testing set were manually annotated by three different observers with a review from an additional observer. Fig. 5 represents sample frames with different numbers of landmark annotations considered from the training set and testing set from the CLUST 2D dataset provided by the organizers. The results for the challenge are evaluated by the MICCAI organizers, and the results are displayed on the leaderboard on their website.

A qualitative analysis was also performed on the training set. In this set of experiments, we tested and analyzed the contribution of each of the four architectures described: 1) original SiamFC (orgSiamFC); 2) original SiamFC with template update (TU_SiamFC); 3) original SiamFC with LKF (LKF_SiamFC); and the 4) original SiamFC with template update and LKF (upgdSiamFC) were tested using the training set. Synthetically generated occlusions were induced to mimic possible artifacts, such as speckle decorrelation, out-of-plane motion, or human errors in handling US probes. In each of the image sequences, a rectangular region with pixel values in the range $(0, 255)$ from a uniform discrete distribution was induced as occlusion in two consecutive frames around the reference object. The area of the induced occlusion patch was nine times the area of the landmark ROI chosen in the first frame. The occlusions were induced such that the occlusion patch was centered about the landmark. Such 2-frame occlusions were induced three times at frame numbers 30, 60, and 90. The performance of the four architectures, in this case, was evaluated by calculating the TE against the ground truth.

B. Performance Evaluation

Given the ground-truth annotations p_i and tracked outputs $\hat{p}_i$, the tracking error (TE) for a given target i is calculated as

$$\mathrm{TE}_i(t) = \|p_i - \hat{p}_i\| \tag{14}$$

where $\|\cdot\|$ represents the Euclidean distance between the estimated landmark position p_i and its ground-truth annotation $\hat{p}_i$. The organizers of CLUST challenge evaluate the results by considering the mean, standard deviation (std), and 95th percentile (TE95th) of the TE considered over all frames [57].

C. Parameter Initialization

The two modules introduced into the original architecture require parameters to be initialized. For the template update module, the correlation coefficient is set based on trial and error experiments performed on the training set with a visual inspection. Having set the correlation coefficient for the template update module, we proceed to initialize the parameters for the LKF.

The LKF needs two parameters to be initialized beforehand: the coefficient of state uncertainty (q) and the coefficient of measurement uncertainty (r). In order to find the appropriate values for the two coefficients, we ran the proposed model with varying parameter values in the range $(0, 10)$ and calculated the TE against the available ground truth. The parameters corresponding to the lowest TE were chosen.

The landmarks in the CLUST 2D dataset are points represented by their coordinates (x, y). In order to track these landmarks, an ROI has to be selected centered on the landmark. To this end, we choose a rectangular region of predefined height and width around the landmark available from the first frame. SiamFC has the capability to adjust the size of the ROI. Since only centroid locations are provided with the ground

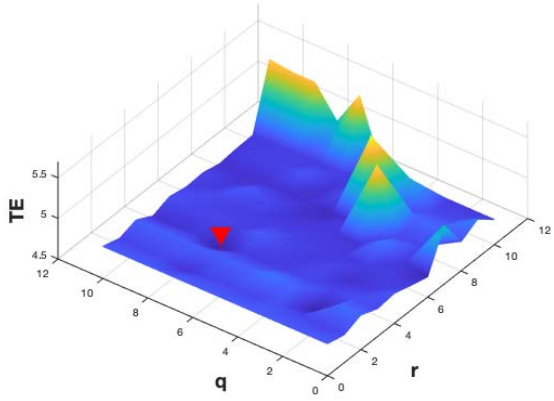


Fig. 6. Performance of the proposed model with varying parameter values in the range ▷0, 10◁ for all the image sequences in the training set. Based on the performance, the coefficient of state uncertainty was chosen to be $q = 7$, and the coefficient of measurement uncertainty was chosen to be $r = 3$.

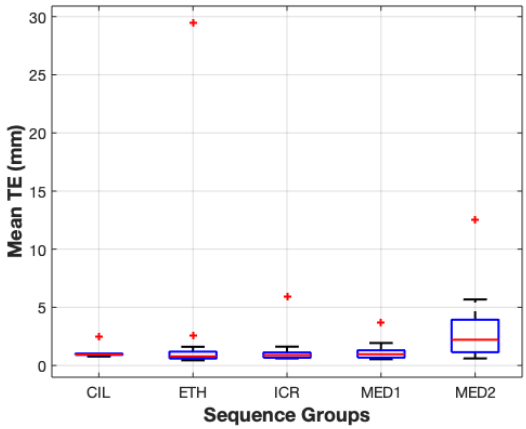


Fig. 7. Boxplot of $\mathrm{TE}_{\mathrm{mean}}$ obtained using upgdSiamFC for all the sequence groups in the testing set. The figure indicates that two major outliers ($\mathrm{TE}_{\mathrm{mean}} > 10$ mm) from the *ETH* and the *Med2* groups are responsible for a high TE.

truth, we exploit this feature to choose a rectangular region around the landmark based on visual inspection and then allow SiamFC to adjust the size of the rectangle in subsequent frames.

IV. Results and Discussion

As mentioned earlier, the correlation coefficient G in the template update module was chosen based on trial and error experiments with a visual inspection. The value of the correlation coefficient was chosen to be $G = 0.7$. With the G set, the linear motion model parameters were chosen based on the performance of the upgdSiamFC on the training set. Fig. 6 represents the performance of our approach when using the two coefficients q and r. The parameters were varied in the range $(0, 10)$, and the proposed model was run through all the image sequences provided with the training set. From Fig. 6, it can be seen that the coefficient pair that produced the lowest TE were $q = 7$ and $r = 3$. The size of the rectangular region of the first frame varied from 10 mm $\times$ 10 mm to 22 mm $\times$ 22 mm, depending on the size of the landmark.

A. Quantitative Analysis

The evaluation of the proposed model was performed on the testing set as specified by the organizers. The performance of our model on each image sequence group is listed in Table I. It can be seen that the proposed model achieves an overall accuracy of 1.59 ± 3.68. Fig. 7 represents the boxplot of TE_{mean} for each image sequence group. It can be seen that there are two major outliers ($\text{TE}_{\text{mean}} > 10$ mm) that have extremely high TE. We emphasize that the results were obtained without performing any transfer learning (retraining model with US images) on the SiamFC.

Table II presents the comparison of our method with respect to the other state-of-the-art methods and human observers on the testing set. In the given table, the *No Tracking* row indicates that no tracking method was used, and the landmark location available on the initial frame was used for the prediction of the landmark in the subsequent frames. This row indicates the necessity of an object tracking method for image-guided radiation therapy. The groups in II have presented both CNN-based methods [52], [53], [66] and traditional methods, such as block matching [59], optical flow [60], [63], correlation filters-based matching [61], [65], and SIFT-based feature matching [62]. The CNN-based methods are advantageous in which they are capable of learning hierarchical features. In addition, the nonlinearity allows for learning intricate features resulting in accurate matching.

Fig. 8 represents an example of tracking a landmark in a randomly chosen image sequence from the training set. The top two graphs represent the displacement of the landmark along with lateral (x) and axial directions (y), respectively, for a set of consecutive frames. Landmark locations obtained by upgadSiamFC, ground truth, and the *no tracking* methods are plotted. Ground-truth locations for landmarks are not available for every frame. For better visualization, we also display a set of images with annotations for the landmark location obtained by the ground truth, upgdSiamFC, and *no tracking* methods at the bottom of Fig. 8. The locations corresponding to the annotations in these images are also plotted in the two graphs (marked with red "+").

TABLE I
Tracking Performance for Each Sequence Group on the Testing Set

Sequence Group	Num. of videos	Mean_TE (mm)	Std_TE (mm)	TE95th (mm)
CIL	06	1.17	0.89	2.95
ETH	30	1.65	4.48	2.65
ICR	13	1.29	1.83	5.16
MED1	27	1.74	2.93	5.80
MED2	09	1.57	1.93	6.72
Overall	**85**	**1.59**	**3.68**	**4.21**

TABLE II
Tracking Performance of upgdSiamFC Against Other State-of-the-Art Methods and Human Observers. Our Method is Highlighted in Bold

Methods	Mean_TE (mm)	Std_TE (mm)	TE95th (mm)
No Tracking	6.25	5.11	16.48
Liu., et al.	0.69	0.67	1.57
Shepard A., et al.	0.72	1.25	1.71
Williamson T., et al.	0.74	1.03	1.85
Jeungyoon L., et al.	0.85	0.8	2.32
Shen C., et al.	1.11	0.91	2.68
Hallack A., et al.	1.21	3.17	2.82
Gomariz A., et al.	1.34	2.57	2.95
Makhinya M. and Goksel O.	1.44	2.8	3.62
upgdSiamFC	**1.59**	**3.69**	**4.21**
Ihle F. A.	2.48	5.09	15.13
Kondo S	2.91	10.52	5.18
Nouri D. and Rothberg A	3.35	5.21	14.19
observer1	0.46	0.36	1.13
observer2	0.47	0.34	1.08
observer3	0.44	0.32	1.03

With a detailed analysis, it was found that the image sequences *ETH-11-1* and *Med-11* were responsible for the high TE. Fig. 9 represents a set of images from the two image sequences. The first row represents images from *ETH-11-1*, where the reference object is completely lost during tracking. In this particular image sequence, a visual inspection reveals that there is hardly any difference between background and foreground. Since our model was not trained using the training set for this particular dataset unlike [52], [53], it eventually lost the object resulting in a very high TE ($\text{TE}_{\text{mean}} = 29.45$ mm). The second row represents images from *Med-11*, where the reference object was found to be very close to the border of the US image. The very close proximity of the reference object to the border prevents the landmark from being centered within the rectangle. Consequently, the upgdSiamFC failed to center the landmark within the rectangular region in a considerably large number of frames resulting in a very high TE ($\text{TE}_{\text{mean}} = 12.55$ mm).

Table III represents the TE_{mean} for each of the image sequence group after the two outliers were removed (standard deviation is not shown since tracked outputs for every frame is not provided by the CLUST organizers). It can be seen that TE_{mean} reduces considerably for the groups *ETH* (1.65–0.99 mm) and *Med2* (1.57–0.40 mm). As a result,

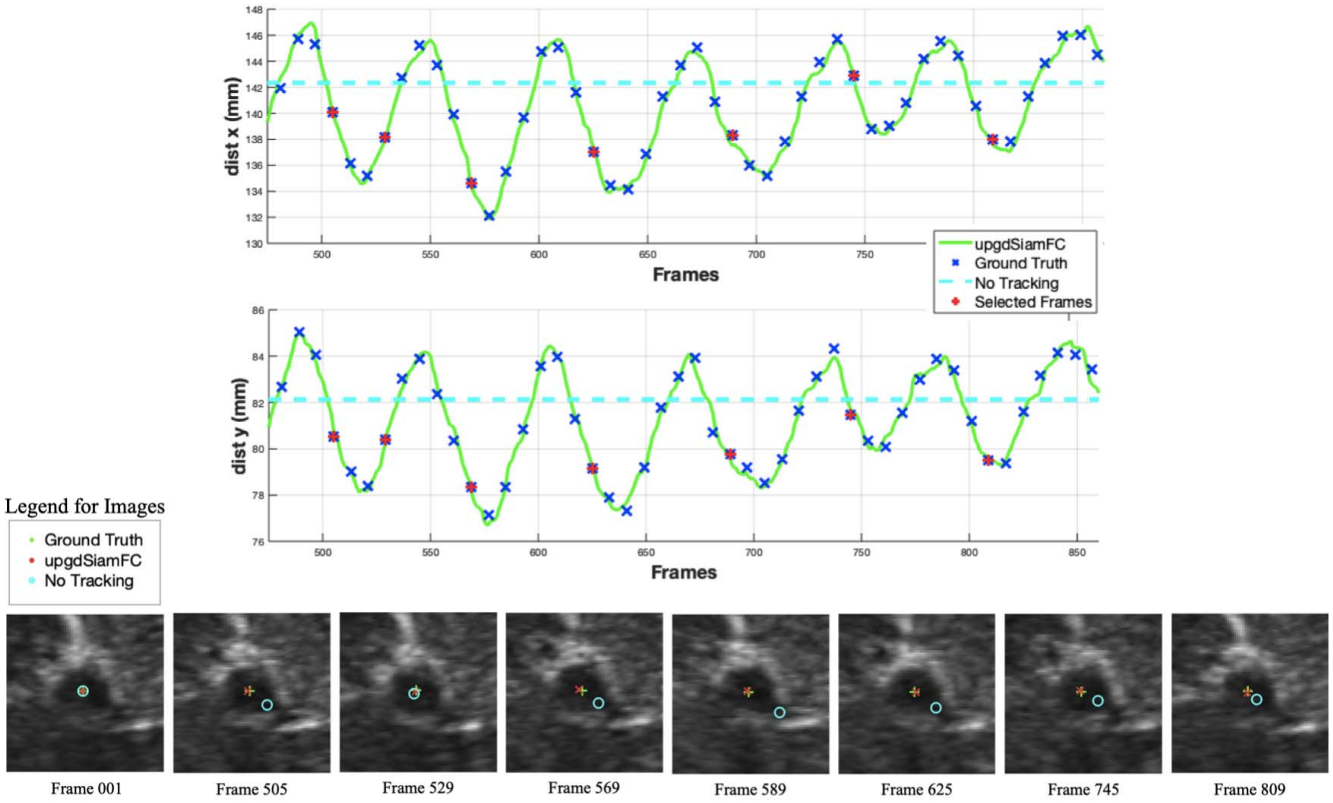


Fig. 8. Tracking of a given landmark by upgdSiamFC. The two top graphs represent the displacement of the landmark along lateral (*x*) and axial (*y*) directions, respectively, for a set of consecutive frames. The ground-truth locations are superimposed on the tracked locations obtained using upgdSiamFC. Locations of the landmark predicted by *No tracking* method are represented in dashed lines. A set of frames (eight frames including the first frame) are also selected to be displayed with annotations from ground truth, upgdSiamFC, and *no tracking*. Annotations for the selected frames (except for the first frame) are also plotted on the graphs (marked with red "+").

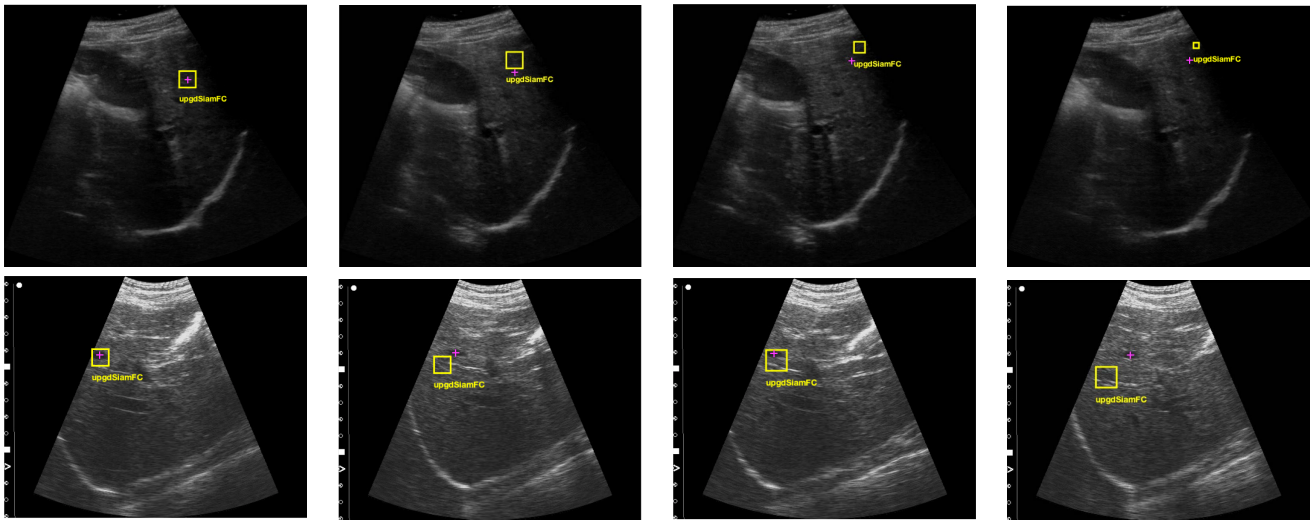
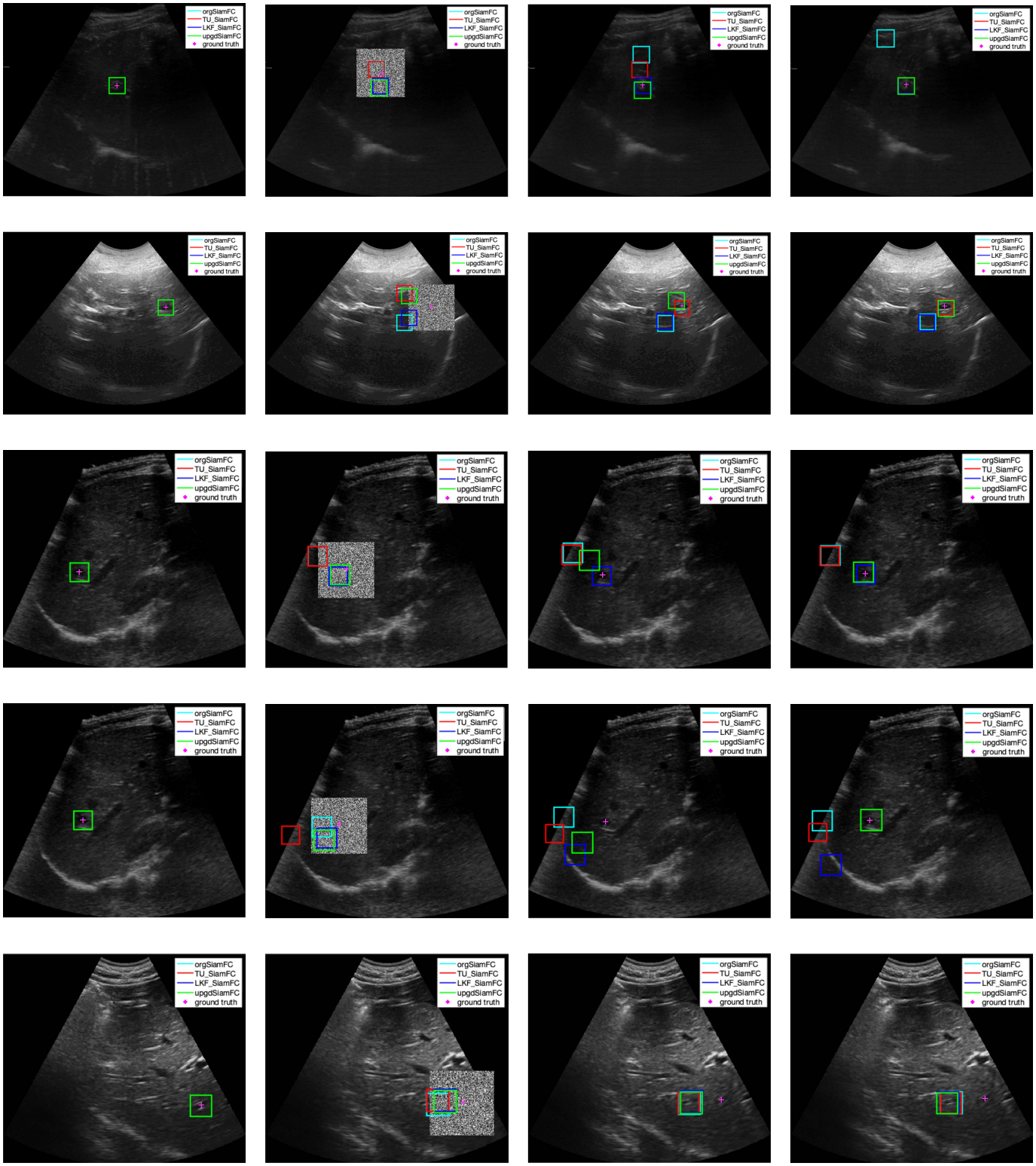


Fig. 9. Two image sequences with extremely high TE_{mean} ($>$ 10 mm). First row: images from the image sequence *ETH-11-1*, where the reference object is lost due to the lack of difference between background and foreground. TE_{mean} for this sequence was 29.17 mm. Second row: images from the image sequence *Med-11*, where the reference object was found to be very close to the border, and consequently, tracker fails to keep up with the landmark. TE_{mean} for this sequence was 12.55 mm. Ground-truth locations are represented using "+."

overall TE_{mean} also reduced from 1.59–1.15 mm. With the two outliers removed, it is worth noting that the performance of our proposed model is better than the Siamese architecture-based method proposed by Gomariz *et al.* [53] (see Table II: $TE_{mean} = 1.34$ mm) despite the fact that transfer learning using CLUST 2D dataset was not performed on our model.

The other four outliers (3 mm $< TE_{mean} <$ 6 mm) with relatively lower TE_{mean} were a result of drastically deforming landmarks, in which the centroid of the landmarks does not coincide with the centroid of the rectangle although the reference object is within the rectangle. Tracking of these landmarks could be improved by incorporating transfer learning using CLUST 2D dataset [52], [53].



(a) Frame1 (b) Frame31 (c) Frame32 (d) Frame36

Fig. 10. Different scenarios, in which the behavior of the four architectures are show when occlusion was encountered. Rows: image sequences. Columns: frames considered at different instances of time for the given image sequence. First row: images from *ETH-01-2*, where only orgSiamFC fails to retrieve the landmark. Second row: images from *CIL-01*, in which orgSiamFC and LKF_SiamFC fail to retrieve the landmark. Third row: images from *ETH-04-1*, in which orgSiamFC and TU_SiamFC fail to retrieve the landmark. Fourth row: images from *ETH-04-2*, in which the only upgdSiamFC was able to retrieve the landmark. Fifth row: images from *ICR-01* with a boundary case scenario, where all the architectures fail to retrieve the landmark. (a) Frame 1. (b) Frame 31. (c) Frame 32. (d) Frame 36.

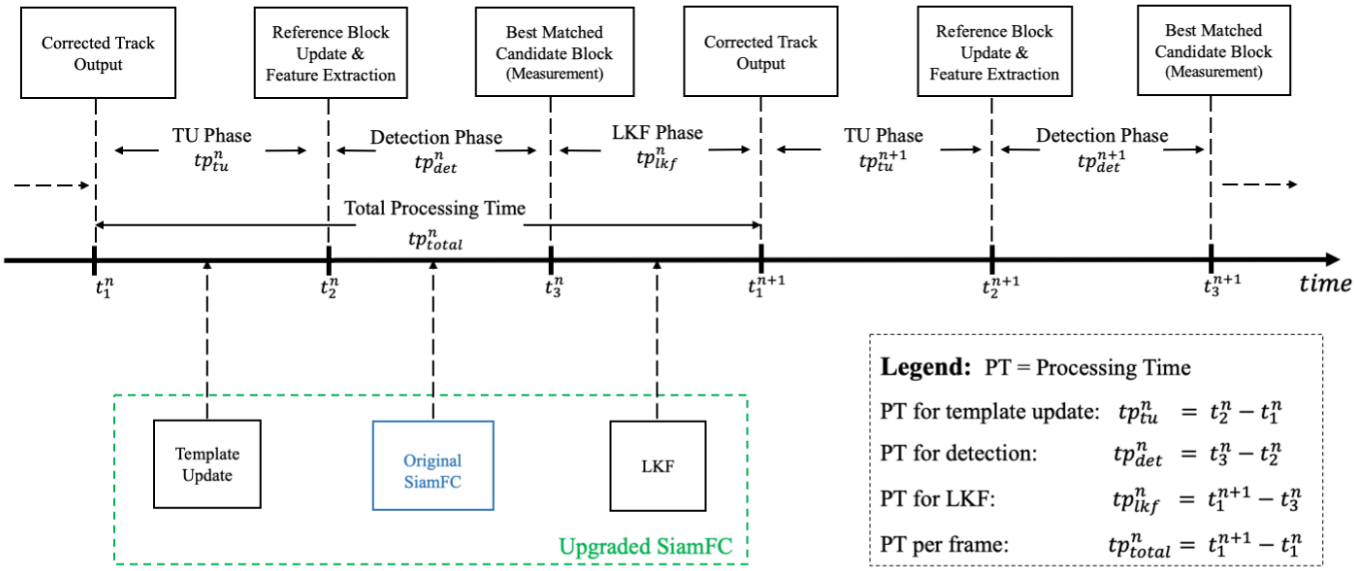


Fig. 11. Temporal relationship of different modules of the proposed model with respect to the reference block and the search region. Processing of a given frame is divided into three parts (in the order): the TU phase $\triangleright \mathrm{tp}^n_{\mathrm{tu}} \triangleleft$, the detection phase $\triangleright \mathrm{tp}^n_{\mathrm{det}} \triangleleft$, and the LKF phase $\triangleright \mathrm{tp}^n_{lkf} \triangleleft$. The start of the phases is marked as t^n_i along the time axis, where the superscript n represents the frame number and the subscript i is an integer representing the beginning of the ith phase. The total processing time per frame $\triangleright \mathrm{tp}^n_{\mathrm{total}} \triangleleft$ is given by $\mathrm{tp}^n_{\mathrm{total}} = \mathrm{tp}^n_{\mathrm{tu}} + \mathrm{tp}^n_{\mathrm{det}} + \mathrm{tp}^n_{lkf}$.

TABLE III
TRACKING PERFORMANCE FOR EACH SEQUENCE GROUP IN THE TESTING SET AFTER REMOVING THE OUTLIERS

Sequence Group	Num. of videos	Mean_TE (mm)
CIL	06	1.17
ETH	29	**0.99**
ICR	13	1.29
MED1	27	1.74
MED2	08	**0.40**
Overall	83	**1.15**

TABLE IV
TRACKING PERFORMANCE OF DIFFERENT ARCHITECTURES ON THE TRAINING SET WHEN OCCLUSIONS WERE INDUCED

Architecture	Mean_TE (mm)	Std_TE (mm)
orgSiamFC	10.40	19.20
TU_SiamFC	4.65	11.14
LKF_SiamFC	7.33	15.77
upgdSiamFC	**2.56**	**5.65**

B. Qualitative Analysis

In order to emphasize the importance of the upgdSiamFC, synthetic occlusions were induced into each of the image sequences. As mentioned earlier, four different architectures were tested for robustness with synthetic occlusions induced around the landmarks. Table IV represents the performance of the four architectures on the training set. It is evident that the best results were obtained when upgdSiamFC is used. Table IV indicates that template update and LKF do contribute, individually, toward improving the robustness of SiamFC to some extent. LKF impedes the detection from drastically changing the location when the landmark is lost due to an occlusion. This helps the tracker to locate back the reference object when the occlusion is removed. However, detections can be very strong in the presence of false positives (landmarks with similar appearance) around the reference block resulting in LKF failing to retrieve the original landmark. On the other hand, template update ensures that the latest appearance of the landmark is updated as the reference block. Therefore, template update helps the detection module to retrieve the landmark despite the change in its appearance compared to the first frame. Nevertheless, there is no mechanism for the template update module to hold back detections from going astray when occlusions are encountered. Therefore, combining the two modules will ensure that the latest appearance of the landmark is updated, and the displacement of the detections is controlled by the motion model. Thus, the upgdSiamFC outperforms the other three architectures by a significant margin.

Fig. 10 presents different scenarios substantiating the aforementioned claims. In Fig. 10, five different scenarios are shown corresponding to five image sequences. Each row represents an image sequence. Column (*a*) represents the first frame of the image sequence. It is obvious that detections from all four architectures overlap each other. Column (*b*) represents the 31st frame of the image sequence with occlusion induced. It is expected that the behavior of the detection module in this frame is unpredictable. Column (*c*) represents the 32nd frame (i.e., frame immediately succeeding the occluded-frame). Finally, column (*d*) represents the 36th frame (the fifth succeeding frame after the encounter of the occlusion).

The first row represents an image sequence (*ETH-01-2*) where only orgSiamFC fails to retrieve the landmark. In this given situation, template update and LKF are individually good enough to assist orgSiamFC in reviving the landmark. The second row represents an image sequence (*CIL-01*), in which orgSiamFC and LKF_SiamFC fail to retrieve the

landmark. We attribute this to the presence of a false positive in close proximity to the original landmark. Consequently, both orgSiamFC and LKF_SiamFC are pulled toward the false positive as seen in column (*c*) of the second row. Although LKF hampers drastic movement of the detection, strong detection from false positive dominates the motion model. On the other hand, TU_SiamFC retrieves the landmark since the reference model is updated. The third row represents an image sequence (*ETH-04-1*), in which orgSiamFC and TU_SiamFC fail to retrieve the landmark. In this scenario, while the detection modules in orgSiamFC and TU_SiamFC lost the landmark to a point of no return due to the unpredictable behavior, LKF was successful in holding back the detection and assisting in retrieving the landmark. It should be noted that upgdSiamFC was successful in all the above scenarios since at least one of the modules contributed positively in assisting the detection module to retrieve the landmark. The fourth row represents an image sequence (*ETH-04-2*), in which the only upgdSiamFC was able to retrieve the landmark and all the other architectures failed to do so. The presence of a false positive in the *ETH-04-2* image sequence pulls LKF_SiamFC away from the landmark as seen from the image in column (*c*) of the sequence. Nevertheless, the most recent appearance of the landmark together with the LKF, upgdSiamFC, was able to bounce back to the original landmark. Finally, the fifth row represents an image sequence (*ICR-01*) with a boundary case scenario, where all the architectures fail to retrieve the landmark. The presence of a false positive, which has a very similar appearance to the original landmark, renders both the modules ineffective causing the detection module to shift to the false positive.

C. Processing Time

Our experiments were run on MATLAB 2019b (MathWorks, Natick, MA, USA), on a Windows OS operating on a 3.4-GHz Intel i7 processor with 16-GB RAM. With this configuration, the average processing time (tp) per frame of the proposed model was $tp^n_{\text{total}} = 250$ ms, where the superscript n represents the nth frame. In other words, without the use of GPU and any code optimization techniques, our proposed model operates at 4 frames per second (fps) on the aforementioned machine. Average tp was calculated by taking the mean of tp over all frames in the CLUST dataset. Fig. 11 illustrates the temporal relationship of different modules of the proposed model with respect to the reference block and the search region. It can be seen that, for any given frame, a template update is carried out first based on the track output of the previous frame. Detection is performed after the template update, and finally, the LKF module generates the corrected output. The detection phase claims the maximum share of the tp with $tp^n_{\text{det}} = 230$ ms. Processing time of the template update phase (tp^n_{tu}) and the LKF phase (tp^n_{lkf}) were found to be 19.67 ms and 86μs, respectively.

In [40], the authors claim that the original SiamFC operates at 86 fps on a machine equipped with a single NVIDIA GeForce GTX Titan X and an Intel Core i7−4790K at 4.0GHz. From the aforementioned processing time of different phases, we know that, while the detection phase demands about 91.9% of the tp^n_{total}, the two newly introduced modules claim only a negligible fraction with tp^n_{tu} claiming 7.87% and tp^n_{lkf} claiming 0.035% of tp^n_{total}. Therefore, it is easy to see that our proposed model, with necessary optimization and a GPU, can operate in real-time.

In summary, it is important for the detection module to adapt to deforming objects and to have a motion model to estimate the states of the system over time. Detections tend to deviate and exhibit unpredictable behavior when artifacts such as speckle decorrelation, out-of-plane motion, or human errors in handling US probes are encountered. Best results are obtained by incorporating the two modules—template update and LKF—into the original architecture. From the above results, it is evident that, with a robust model, accurate tracking can be performed even without performing transfer learning. Incorporating this architecture to techniques proposed in [52] and [53], where they trained the Siamese networks for this particular dataset, can further improve the tracking accuracy.

V. Conclusion and Future Work

In this article, we addressed two major limitations of the Siamese architecture-based object tracker. By introducing the template update module, we resolved the constant position model issue and improved the robustness of SiamFC against deforming landmarks. We implemented an LKF to incorporate the missing motion model in the original architecture. The upgraded SiamFC achieved an overall accuracy of 1.59 ± 3.68 that is comparable to other state-of-the-art methods. The upgraded SiamFC also provided promising results against synthetically induced occlusions demonstrating the potential for accurate and robust landmark tracking. For our future work, we intend to improve the detection module of the Siamese network. Using region proposals along with Siamese architecture and combining it with the two modules introduced in this article could significantly improve tracking accuracy. In addition, we also intend to develop nonlinear motion models tailored to the needs of specific tissue motion.

References

[1] J. Jiang and T. Hall, "A robust real-time speckle tracking algorithm for ultrasonic elasticity imaging," in *Proc. IEEE Int. Ultrason. Symp.*, Sep. 2009, pp. 451–454.

[2] D. Boukerroui, J. A. Noble, and M. Brady, "Velocity estimation in ultrasound images: A block matching approach," in *Proc. Biennial Int. Conf. Inf. Process. Med. Imag.* Berlin, Germany: Springer, 2003, pp. 586–598.

[3] G. E. Trahey, J. W. Allison, and O. T. von Ramm, "Angle independent ultrasonic detection of blood flow," *IEEE Trans. Biomed. Eng.*, vol. BME-34, no. 12, pp. 965–967, Dec. 1987.

[4] P. M. Embree and A. B. V. Phantom, "Volumetric blood flow via time-domain correlation: Experimental verification," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 37, pp. 176–189, May 1990.

[5] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yazdi, and X. Li, "Elastography: A quantitative method for imaging the elasticity of biological tissues," *Ultrason. Imag.*, vol. 13, no. 2, pp. 111–134, Apr. 1991.

[6] H. de Hoop, H. Yoon, and K. Kubelick, "Photoacoustic speckle tracking for motion estimation and flow analysis," *J. Biomed. Opt.*, vol. 23, no. 9, Sep. 2018, Art. no. 096001.

[7] G. C. Ng, S. S. Worrell, P. D. Freiburger, and G. E. Trahey, "A comparative evaluation of several algorithms for phase aberration correction," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 41, no. 5, pp. 631–643, Sep. 1994.

[8] S. W. Flax and M. O'Donnell, "Phase-aberration correction using signals from point reflectors and diffuse scatterers: Basic principles," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 35, no. 6, pp. 758–767, Nov. 1988.

[9] M. Ersboll *et al.*, "Early diastolic strain rate in relation to systolic and diastolic function and prognosis in acute myocardial infarction: A two-dimensional speckle-tracking study," *Eur. Heart J.*, vol. 35, no. 10, pp. 648–656, Mar. 2014.

[10] I. Z. Nenadic, M. W. Urban, J. F. Greenleaf, J.-L. Gennisson, M. Bernal, and M. Tanter, *Ultrasound Elastography for Biomedical Applications and Medicine*. Hoboken, NJ, USA: Wiley, 2019.

[11] A. Pesavento, C. Perrey, M. Krueger, and H. Ermert, "A time-efficient and accurate strain estimation concept for ultrasonic elastography using iterative phase zero estimation," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 46, no. 5, pp. 1057–1067, Sep. 1999.

[12] L. Chen, G. M. Treece, J. E. Lindop, A. H. Gee, and R. W. Prager, "A quality-guided displacement tracking algorithm for ultrasonic elasticity imaging," *Med. Image Anal.*, vol. 13, no. 2, pp. 286–296, 2009.

[13] M. McCormick, N. Rubert, and T. Varghese, "Bayesian regularization applied to ultrasound strain imaging," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 6, pp. 1612–1620, Jun. 2011.

[14] B. Byram, G. E. Trahey, and M. Palmeri, "Bayesian speckle tracking. Part II: Biased ultrasound displacement estimation," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 60, no. 1, pp. 144–157, Jan. 2013.

[15] M. Suhling, M. Arigovindan, C. Jansen, P. Hunziker, and M. Unser, "Myocardial motion analysis from B-mode echocardiograms," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 525–536, Apr. 2005.

[16] E. S. Ebbini, "Phase-coupled two-dimensional speckle tracking algorithm," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 53, no. 5, pp. 972–990, May 2006.

[17] M. Almekkawy and E. Ebbini, "Two-dimensional speckle tracking using parabolic polynomial expansion with riesz transform," in *Proc. IEEE 14th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2017, pp. 201–205.

[18] B. Rebholz, F. Zheng, and M. Almekkawy, "Two-dimensional iterative projection method for subsample speckle tracking of ultrasound images," *Med. Biol. Eng. Comput.*, vol. 58, no. 12, pp. 2937–2951, Dec. 2020.

[19] X. Chen, M. J. Zohdy, S. Y. Emelianov, and M. O'Donnell, "Lateral speckle tracking using synthetic lateral phase," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 51, no. 5, pp. 540–550, May 2004.

[20] J. Jiang and T. J. Hall, "A generalized speckle tracking algorithm for ultrasonic strain imaging using dynamic programming," *Ultrasound Med., Biol.*, vol. 35, no. 11, pp. 1863–1879, 2009.

[21] H. Rivaz, E. Boctor, P. Foroughi, R. Zellars, G. Fichtinger, and G. Hager, "Ultrasound elastography: A dynamic programming approach," *IEEE Trans. Med. Imag.*, vol. 27, no. 10, pp. 1373–1377, Oct. 2008.

[22] T. Benz, M. Kowarschik, and N. Navab, "Kernel-based tracking in ultrasound sequences of liver," in *Proc. Challenge Liver Ultrasound Tracking, MICCAI Workshop*, 2014, pp. 21–28.

[23] B. Rebholz and M. Almekkawy, "Analysis of speckle tracking methods: Correlation and RF interpolation," in *Proc. IEEE 4th Int. Conf. Image Process., Appl. Syst. (IPAS)*, Dec. 2020, pp. 120–124.

[24] B. Rebholz and M. Almekkawy, "Constrained RF level interpolation for normalized cross correlation based speckle tracking," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Sep. 2020, pp. 1–4.

[25] Y. Notomi *et al.*, "Measurement of ventricular torsion by two-dimensional ultrasound speckle tracking imaging," *J. Amer. College Cardiol.*, vol. 45, no. 12, pp. 2034–2041, Jun. 2005.

[26] J.-W. H. Korstanje, R. W. Selles, H. Stam, S. E. R. Hovius, and J. G. Bosch, "Development and validation of ultrasound speckle tracking to quantify tendon displacement," *J. Biomech.*, vol. 43, pp. 1373–1379, Feb. 2010.

[27] B. Peng, Y. Xian, and J. Jiang, "A convolution neural network-based speckle tracking method for ultrasound elastography," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Oct. 2018, pp. 206–212.

[28] J. Porée, M. Baudet, F. Tournoux, G. Cloutier, and D. Garcia, "A dual tissue-doppler optical-flow method for speckle tracking echocardiography at high frame rate," *IEEE Trans. Med. Imag.*, vol. 37, no. 9, pp. 2022–2032, Sep. 2018.

[29] P. Joos *et al.*, "High-frame-rate speckle-tracking echocardiography," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 65, no. 5, pp. 720–728, May 2018.

[30] A. Giachetti, "Matching techniques to compute image motion," *Image Vis. Comput.*, vol. 18, no. 3, pp. 247–260, Feb. 2000.

[31] M. Persson, A. Ryden Ahlgren, A. Eriksson, T. Jansson, H. W. Persson, and K. Lindstrom, "Non-invasive measurement of arterial longitudinal movement," in *Proc. IEEE Ultrason. Symp.*, Oct. 2002, pp. 1783–1786.

[32] M. K. Almekkawy, Y. Adibi, F. Zheng, E. Ebbini, and M. Chirala, "Two-dimensional speckle tracking using zero phase crossing with riesz transform," in *Proc. Meetings Acoust. (ASA)*, vol. 22. New York, NY, USA: Acoustical Society of America, 2014, Art. no. 020004.

[33] B. Rebholz and M. Almekkawy, "Efficacy of kriging interpolation in ultrasound imaging; subsample displacement estimation," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 2137–2141.

[34] M. Cinthio, Å. R. Ahlgren, J. Bergkvist, T. Jansson, H. W. Persson, and K. Lindström, "Longitudinal movements and resulting shear strain of the arterial wall," *Amer. J. Physiol.-Heart Circulatory Physiol.*, vol. 291, no. 1, pp. H394–H402, Jul. 2006.

[35] J. Bang, T. Dahl, A. Bruinsma, J. H. Kaspersen, T. A. Nagelhus Hernes, and H. O. Myhre, "A new method for analysis of motion of carotid plaques from RF ultrasound images," *Ultrasound Med. Biol.*, vol. 29, no. 7, pp. 967–976, Jul. 2003.

[36] J. Luo and E. Konofagou, "A fast normalized cross-correlation calculation method for motion estimation," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 57, no. 6, pp. 1347–1357, Jun. 2010.

[37] M. O'Donnell, A. R. Skovoroda, B. M. Shapo, and S. Y. Emelianov, "Internal displacement and strain imaging using ultrasonic speckle tracking," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 41, no. 3, pp. 314–325, May 1994.

[38] S. Bharadwaj and M. Almekkawy, "Faster search algorithm for speckle tracking in ultrasound images," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 2142–2146.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[40] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 850–865.

[41] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[42] R. J. van Sloun, R. Cohen, and Y. C. Eldar, "Deep learning in ultrasound imaging," *Proc. IEEE*, vol. 108, no. 1, pp. 11–29, Jan. 2020.

[43] A. C. Luchies and B. C. Byram, "Deep neural networks for ultrasound beamforming," *IEEE Trans. Med. Imag.*, vol. 37, no. 9, pp. 2010–2021, Sep. 2018.

[44] A. Wiacek, E. Gonzalez, and M. A. L. Bell, "CohereNet: A deep learning architecture for ultrasound spatial correlation estimation and coherence-based beamforming," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 12, pp. 2574–2583, Dec. 2020.

[45] D. Hyun, L. L. Brickson, K. T. Looby, and J. J. Dahl, "Beamforming and speckle reduction using neural networks," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 66, no. 5, pp. 898–910, May 2019.

[46] M. Amiri, R. Brooks, and H. Rivaz, "Fine tuning U-Net for ultrasound image segmentation: Which layers," in *Domain Adaptation and Representation Transfer and Medical Image Learning With Less Labels and Imperfect Data*. Berlin, Germany: Springer, 2019, pp. 235–242.

[47] Y. H. Yoon, S. Khan, J. Huh, and J. C. Ye, "Efficient B-mode ultrasound image reconstruction from sub-sampled RF data using deep learning," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 325–336, Feb. 2019.

[48] R. Prevost *et al.*, "3D freehand ultrasound without external tracking using deep learning," *Med. Image Anal.*, vol. 48, pp. 187–202, Aug. 2018.

[49] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[50] M. G. Kibria and H. Rivaz, "GLUENet: Ultrasound elastography using convolutional neural network," in *Proc. Simulation, Image Process., Ultrasound Syst. Assist. Diagnosis Navigat.* Cham, Switzerland: Springer, 2018, pp. 21–28.

[51] A. K. Z. Tehrani and H. Rivaz, "Displacement estimation in ultrasound elastography using pyramidal convolutional neural network," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 12, pp. 2629–2639, Dec. 2020.

[52] F. Liu, D. Liu, J. Tian, X. Xie, X. Yang, and K. Wang, "Cascaded one-shot deformable convolutional neural networks: Developing a deep learning model for respiratory motion estimation in ultrasound sequences," *Med. Image Anal.*, vol. 65, Oct. 2020, Art. no. 101793.
[53] A. Gomariz, W. Li, E. Ozkan, C. Tanner, and O. Goksel, "Siamese networks with location prior for landmark tracking in liver ultrasound sequences," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2019, pp. 1757–1760.
[54] S. Bharadwaj and M. Almekkawy, "Deep learning based motion tracking of ultrasound image sequences," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Sep. 2020, pp. 1–4.
[55] S. Bharadwaj and M. Almekkawy, "Motion estimation for ultrasound image sequences using deep learning," *J. Acoust. Soc. Amer.*, vol. 148, no. 4, p. 2487, 2020.
[56] S. Bharadwaj and M. Almekkawy, "Improved siamese network for motion tracking in ultrasound images," *J. Acoust. Soc. Amer.*, vol. 149, no. 4, p. A114, Apr. 2021.
[57] V. De Luca *et al.*, "Evaluation of 2D and 3D ultrasound tracking algorithms and impact on ultrasound-guided liver radiotherapy margins," *Med. Phys.*, vol. 45, no. 11, pp. 4986–5003, 2018.
[58] V. De Luca *et al.*, "The 2014 liver ultrasound tracking benchmark," *Phys. Med. Biol.*, vol. 60, p. 5571, Jul. 2015.
[59] A. J. Shepard, B. Wang, T. K. F. Foo, and B. P. Bednarz, "A block matching based approach with multiple simultaneous templates for the real-time 2D ultrasound tracking of liver vessels," *Med. Phys.*, vol. 44, no. 11, pp. 5889–5900, Nov. 2017.
[60] T. Williamson, W. Cheung, S. K. Roberts, and S. Chauhan, "Ultrasound-based liver tracking utilizing a hybrid template/optical flow approach," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, no. 10, pp. 1605–1615, Oct. 2018.
[61] C. Shen, J. He, Y. Huang, and J. Wu, "Discriminative correlation filter network for robust landmark tracking in ultrasound guided intervention," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2019, pp. 646–654.
[62] A. Hallack, B. Papiez, A. Cifor, M. Gooding, and J. Schnabel, "Robust liver ultrasound tracking using dense distinctive image features," in *Proc. MICCAI Challenge Liver Ultrasound Tracking*, 2015, pp. 28–35.
[63] M. Makhinya and O. Goksel, "Motion tracking in 2D ultrasound using vessel models and robust optic-flow," in *Proc. MICCAI CLUST*, vol. 20, 2015, pp. 20–27.
[64] G. Zachmann, I. U. Frese, and F. A. Ihle, "Random forests for tracking on ultrasonic images," M.S. thesis, Univ. Bremen, Bremen, Germany, 2017.
[65] S. Kondo, "Liver ultrasound tracking using kernelized correlation filter with adaptive window size selection," in *Proc. MICCAI Workshop, Challenge Liver Ultrasound Tracking*, 2015, pp. 13–19.
[66] D. Nouri and A. Rothberg, "Liver ultrasound tracking using a learned distance metric," in *Proc. MICCAI Workshop, Challenge Liver Ultrasound Tracking*, 2015, pp. 5–12.
[67] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2. Paris, France: Lille, 2015, pp. 1–8.
[68] S. Chanda, A. C. Gv, A. Brun, A. Hast, U. Pal, and D. Doermann, "Face recognition—A one-shot learning perspective," in *Proc. 15th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Nov. 2019, pp. 113–119.
[69] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[70] V. De Luca, M. Tschannen, G. Székely, and C. Tanner, "A learning-based approach for fast and robust vessel tracking in long ultrasound sequences," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Berlin, Germany: Springer, 2013, pp. 518–525.
[71] F. Preiswerk *et al.*, "Model-guided respiratory organ motion prediction of the liver from 2D ultrasound," *Med. Image Anal.*, vol. 18, no. 5, pp. 740–751, Jul. 2014.
[72] M. A. Lediju, B. C. Byram, E. J. Harris, P. M. Evans, and J. C. Bamber, "3D liver tracking using a matrix array: Implications for ultrasonic guidance of IMRT," in *Proc. IEEE Int. Ultrason. Symp.*, Oct. 2010, pp. 1628–1631.
[73] M. A. L. Bell, B. C. Byram, E. J. Harris, P. M. Evans, and J. C. Bamber, "*In vivo* liver tracking with a high vol. rate, 4D ultrasound scanner and a 2D matrix array probe," *Phys. Med. Biol.*, vol. 57, no. 5, p. 1359, 2012.
[74] S. Vijayan, S. Klein, E. F. Hofstad, F. Lindseth, B. Ystgaard, and T. Lango, "Validation of a non-rigid registration method for motion compensation in 4D ultrasound of the liver," in *Proc. IEEE 10th Int. Symp. Biomed. Imag.*, Apr. 2013, pp. 792–795.
[75] J. Banerjee, C. Klink, E. D. Peters, W. J. Niessen, A. Moelker, and T. van Walsum, "4D liver ultrasound registration," in *Proc. Int. Workshop Biomed. Image Registration*. Cham, Switzerland: Springer, 2014, pp. 194–202.

Skanda Bharadwaj (Member, IEEE) received the B.E. degree in telecommunication engineering from the PES Institute of Technology, Bengaluru, India, in 2015, and the M.S. degree in computer science and engineering from The Pennsylvania State University, University Park, PA, USA, in 2021.

He was a Computer Vision Engineer with Continental Automotive Components, Bengaluru, for three years. His research interests include computer vision, motion tracking, and deep learning.

Sumukha Prasad (Graduate Student Member, IEEE) received the B.E. degree in telecommunication engineering from PES University, Bengaluru, India, in 2015. He is currently pursuing the master's degree with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, University Park, PA, USA.

He is currently a graduate assistant with an interest in deep learning and computer vision for biomedical imaging, ultrasound computed tomography, and motion tracking.

Mohamed Almekkawy (Member, IEEE) received the B.S. degree in electrical engineering from Ain Shams University, Cairo, Egypt, in 1998, the M.S. degree in electrical engineering from Cairo University, Giza, Egypt, in 2006, and the University of Minnesota, Twin Cities, MN, USA, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Minnesota in 2014.

In 2015, he joined the School of Electrical Engineering and Computer Science, The Pennsylvania State University, University Park, PA, USA, as an Assistant Professor. His current research interests include deep learning for biomedical imaging applications, medical image reconstruction, ultrasound tomography, speckle tracking, elasticity imaging, focused ultrasound for neuromodulation, and therapeutic applications.

Dr. Almekkawy is also an Associate Editor and a Technical Committee Member of the IEEE Engineering in Medicine and Biology Conference (EMBC).