

Bi-Directional Semi-Supervised Training of Convolutional Neural Networks for Ultrasound Elastography Displacement Estimation

Ali K. Z. Tehrani¹, Mostafa Sharifzadeh, Emad Boctor, and Hassan Rivaz², *Senior Member, IEEE*

Abstract—The performance of ultrasound elastography (USE) heavily depends on the accuracy of displacement estimation. Recently, convolutional neural networks (CNNs) have shown promising performance in optical flow estimation and have been adopted for USE displacement estimation. Networks trained on computer vision images are not optimized for USE displacement estimation since there is a large gap between the computer vision images and the high-frequency radio frequency (RF) ultrasound data. Many researchers tried to adopt the optical flow CNNs to USE by applying transfer learning to improve the performance of CNNs for USE. However, the ground-truth displacement in real ultrasound data is unknown, and simulated data exhibit a domain shift compared to the real data and are also computationally expensive to generate. To resolve this issue, semisupervised methods have been proposed in which the networks pretrained on computer vision images are fine-tuned using real ultrasound data. In this article, we employ a semisupervised method by exploiting the first- and second-order derivatives of the displacement field for regularization. We also modify the network structure to estimate both forward and backward displacements and propose to use consistency between the forward and backward strains as an additional regularizer to further enhance the performance. We validate our method using several experimental phantom and *in vivo* data. We also show that the network fine-tuned by our proposed method using experimental phantom data performs well on *in vivo* data similar to the network fine-tuned on *in vivo* data. Our results also show that the proposed method outperforms current deep learning methods and is comparable to computationally expensive optimization-based algorithms.

Index Terms—Convolutional neural network (CNN), displacement estimation, optical flow, semisupervised training, ultrasound elastography (USE).

I. INTRODUCTION

ULTRASOUND (US) imaging has been increasingly used by researchers and clinicians in diagnosis and image-guided intervention since it is less expensive and more portable compared to other imaging modalities. US elastography (USE) is an imaging technique that detects viscoelastic properties of the tissue and has been found useful in different applications, including ablation monitoring [1], [2] and breast lesion characterization [3]. USE methods track the motion of the tissue and can be broadly categorized into two groups: dynamic and quasi-static elastography. In dynamic elastography, acoustic radiation force or an internal force is used to generate fast motions in the tissue [4]. In contrast, the motions in the tissue are slow in quasi-static elastography and can be induced by simply pressing the probe by the operator (free-hand palpation) or using a robotic arm [5]–[8].

In free-hand palpation USE, the operator compresses the tissue by the probe usually in the axial direction. The US radio frequency (RF) data before and after the compression are compared to obtain the displacement map, which indicates the movement of each individual sample. The displacement map is utilized to obtain the strain map, which contains relative elasticity information. In this article, we aim to estimate axial displacement for quasi-static free-hand palpation USE.

Existing techniques for performing the displacement estimation step can be categorized into conventional and deep learning-based methods. Window-based [6], [9]–[12] and optimization-based [8], [13]–[15] are two main groups of conventional methods that have been used widely for USE. Recent optimization-based methods have surpassed window-based ones [14], [15]. The main idea of the optimization-based methods is to determine an initial coarse estimation, usually by dynamic programming (DP) [8], followed by minimizing a regularized cost function to obtain the displacement map. OVERWIND is a recent algorithm that combines the window- and optimization-based methods [15]. OVERWIND employs the L_1 norm as the regularizer to preserve the sharpness of the displacements on the boundaries.

Manuscript received November 11, 2021; accepted January 24, 2022. Date of publication January 27, 2022; date of current version March 28, 2022. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2020-04612 and in part by NSF CAREER under Grant 1653322. (Corresponding author: Ali K. Z. Tehrani.)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board (IRB) at Johns Hopkins.

Ali K. Z. Tehrani, Mostafa Sharifzadeh, and Hassan Rivaz are with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: a_kafaei@encs.concordia.ca; mostafa.sharifzadeh@concordia.ca; hrivaz@ece.concordia.ca).

Emad Boctor is with the Department of Radiology and Radiological Science, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: eboctor@jhmi.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TUFFC.2022.3147097>, provided by the authors. Digital Object Identifier 10.1109/TUFFC.2022.3147097

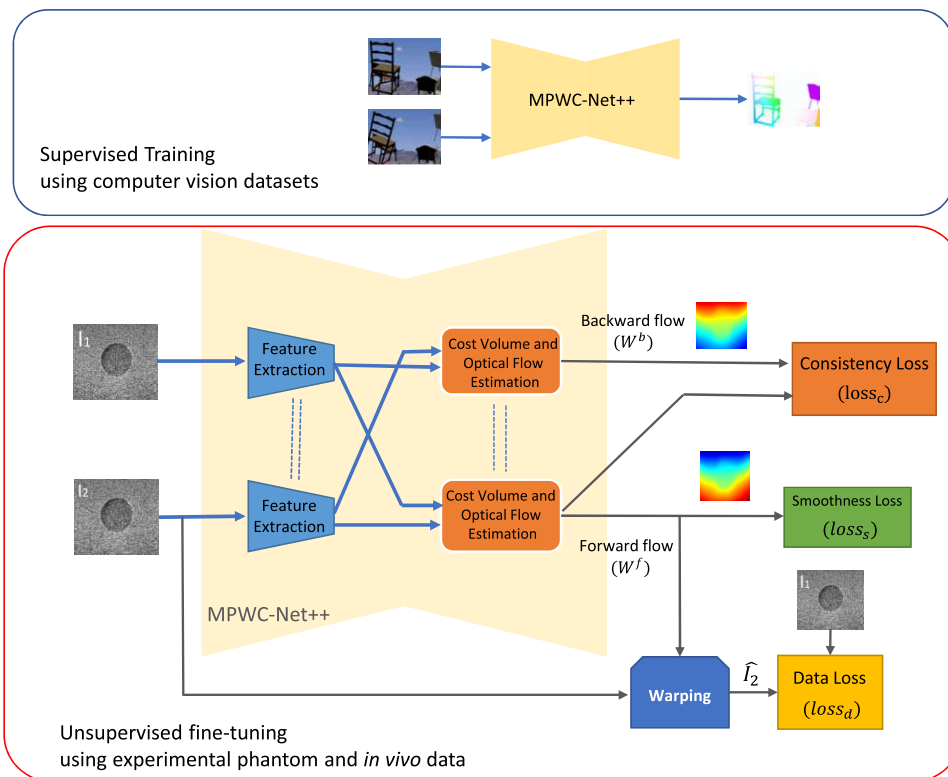


Fig. 1. Overview of the proposed semisupervised training method. The network is first trained using computer vision datasets by supervised methods (the block on top). The network is fine-tuned by real US data using the proposed unsupervised training method (bottom block). The network structure is altered to be able to estimate both forward and backward flows. The layers connected by the dashed lines share weights. The cost volume and optical flow estimation blocks with shared weights are used to estimate both forward and backward flows.

Deep learning-based methods, which have been recently proposed for USE, employ optical flow convolutional neural networks (CNNs) to obtain the displacement map. The first few works used the optical flow CNNs as black boxes for USE [16], [17] or as the initial estimator for optimization-based methods instead of DP [18], [19]. However, the computer vision images and US data are vastly different and the CNN architectures used for the former are not optimized for high-frequency RF data. Motivated to address this issue, we modified the well-known PWC-Net architecture [20] to be adapted to USE, considering the physics of RF data [21]. We called the network modified PWC-Net (MPWC-Net) and obtained substantially more accurate displacement compared to PWC-Net. In another work, we proposed MPWC-Net++ that was an improved version of MPWC-Net with a higher search range and more accurate output displacement [22]. These methods require a GPU to run efficiently and can perform high-frame-rate USE given the rapidly increasing computational power of GPUs. However, their main drawback is that they have a larger variance compared to conventional methods since they are not regularized in contrast to conventional methods. Consequently, their strain images have lower overall quality compared to conventional methods [21]–[23].

Unsupervised training was another avenue that has been followed by the researchers. Delaunay *et al.* [24], [25] trained a U-Net using real US data and developed a recurrent network to deal with a sequence of frames. In [26], we used a light

network, referred to as LiteFlowNet [27], and trained it in a semisupervised fashion. We first used computer vision datasets with known ground truths to train the network using supervised techniques. In the next step, real US data were used to fine-tune the network using an unsupervised method. We substantially improved the strain image quality by using this technique without requiring a large amount of training data.

In this article, we follow the semisupervised training approach. The overview of the method is shown in Fig. 1. We first employ computer vision datasets to train the network in a supervised fashion. We use MPWC-Net++, which has shown high performance in USE. We then fine-tune the network by real US data and extend our idea of semisupervised method by proposing bi-directional unsupervised fine-tuning. We change the structure of MPWC-Net++ to estimate both forward and backward displacements, which is more efficient than running the network two times to estimate them. Consistency loss is also proposed, which is obtained by comparing forward and backward strains. Furthermore, we shed light on the choice of weights for regularization by relating some of them to others. We demonstrate the high-performance of the proposed methods using experimental phantom and *in vivo* data.

II. MATERIAL AND METHOD

A. Deep Supervised CNNs

In this section, several CNNs used in USE displacement estimation are explained.

1) *PWC-Net*: Sun *et al.* [20] proposed using cost volume and warping of the features for optical flow estimation. They achieved the state-of-the-art performance in different computer vision datasets. PWC-Net used a pyramidal structure, in which the optical flow was estimated in different resolution levels. At each pyramid level, the features of the second image were warped by the estimated flow of the previous pyramid to reduce the flow in the next pyramid level. In the next step, cost volume was employed to compare the features of the fixed and warped moved images. In the last step, the optical flow was estimated and used in the next pyramid. Using pyramid structure resulted in the reduction of the number of learnable weights and improved the performance of optical flow estimation [20]. Recently, a variant of PWC-Net called iterative residual refinement PWC-Net (IRR-PWC-Net) was proposed [28]. This network reduced the number of learnable weights even further by using the optical flow estimation block iteratively.

2) *MPWC-Net*: PWC-Net was proposed for computer vision images originally and was not well suited to the high-frequency RF data. We modified the structure of PWC-Net and proposed MPWC-Net for USE [21]. We removed the strides of the first feature extraction layer to preserve high-frequency information in RF data. In order to avoid failure of the network in low pyramid levels, where RF data do not have enough information due to downsampling, we used envelope and B-mode images as additional input channels. We obtained competitive performance with conventional optimization-based methods. The code and the simulation dataset for fine-tuning are available online at code.sonography.ai.

3) *MPWC-Net++*: MPWC-Net had a low displacement range since strides of the first feature extraction layer were removed. In addition, we showed that the real displacement range is much lower than the theoretical one since only a small quantity of training data has high displacement ranges. Therefore, the network is not trained enough to deal well with large displacements, and the predicted flows are noisy in this condition. To address these problems, instead of removing both strides, we only removed one of them and kept the other one. Also, the search range of the cost volume was increased from 4 to 5 in each pyramid level. By doing these modifications, the network had a higher search range and could work better for large displacements. Furthermore, we used IRR-PWC-Net [28] since it has a more efficient structure. Although we applied these modifications to improve the performance in the USE application, they also led to an improvement even for the computer vision dataset [22].

However, the improvements came with costs. Due to modifying the network structure, we had to train the network from scratch (MPWC-Net did not require training from scratch). The training itself was slower compared to the original PWC-Net for two reasons: first, having larger feature maps after removing strides; and second, increasing the cost volume search range. The MPWC-Net++ was trained from scratch using computer vision images, while LiteFlowNet used in [26] was pretrained, and we only fine-tuned it using real US data.

Training from scratch takes 840 h using an NVIDIA Tesla P6 GPU, substantially more time compared to the fine-tuning since the network is randomly initialized and a large dataset (for MPWC-Net++ 22 000 pairs) is employed to train the network. The network weights are publicly available online at code.sonography.ai.

B. Semisupervised Method

Simulation data do not model nonlinear or multiple scattering effects present in real US data [24], [26]. Therefore, we proposed to use real US data for fine-tuning. This method was semisupervised since we first used a pretrained network trained on computer vision images by supervised methods. In the next step, real US data were used to fine-tune the network.

The moved image was warped with the forward flow and compared with the first image, which is called photometric loss. This loss alone resulted in noisy displacements; therefore, inspired by the physics of RF data, we proposed using the first second-order derivatives of displacement in the axial and lateral directions as the regularization. The detailed description of the loss function can be found in [26].

In order to preserve the information of high-frequency RF data, we were not allowed to downsample images and had to use large image sizes during unsupervised fine-tuning, which is challenging due to GPU memory limitations. We used a light network (LiteFlowNet [27]) and gradient checkpointing [29] to be able to train the network on our GPU (Nvidia TITAN V with 12 GB of RAM). Furthermore, we limited the training to only forward flow and the backward flow was used to detect occluded regions.

C. Proposed Method

Let $I_1, I_2 \in \mathbb{R}^{3 \times W \times H}$ denote the fixed and moved images having three channels with width W and height H , and $W^f \in \mathbb{R}^{2 \times W \times H}$ and $W^b \in \mathbb{R}^{2 \times W \times H}$ denote the forward flow ($I_1 \rightarrow I_2$) and backward flow ($I_2 \rightarrow I_1$), respectively. The data loss function for unsupervised fine-tuning can be defined as [26]

$$\text{loss}_d = \Phi(I_1 - \hat{I}_2)_{w \times w} \quad (1)$$

where \hat{I}_2 is the second image warped by the W^f , and unlike [26], a window of size $(w \times w)$ around the sample of interest is selected to compute the loss to reduce the noise caused by interpolation step of warping operation (here, we use a 3×3 window). Φ is the Charbonnier loss that has been widely used in unsupervised optical flow training [30] and defined in the following [31]:

$$\Phi(x) = |(x^2 + \varepsilon)^\alpha|_1 \quad (2)$$

where $|\cdot|_1$ denotes the L_1 norm and α can be altered to give different importance to x . We used $\alpha = 0.5$ for the data loss (would be L_1 norm) and $\alpha = 0.2$ for smoothness and consistency loss to emphasize small values of x . It should be mentioned that inspired by [23], RF data, the envelope and

imaginary part of Hilbert transform of RF data, are utilized as three separate channels of input images.

In order to have a smooth displacement field, the first-order derivatives of the displacements in the axial and lateral directions are used for regularization

$$\text{loss}_s^1 = \lambda_{11} \Phi \left(\frac{\partial W_a^f}{\partial a} - \left\langle \frac{\partial}{\partial a} W_a^f \right\rangle \right) + \lambda_{12} \Phi \left(\frac{\partial W_a^f}{\partial l} \right) + \lambda_{21} \Phi \left(\frac{\partial W_l^f}{\partial a} \right) + \lambda_{22} \Phi \left(\frac{\partial W_l^f}{\partial l} \right) \quad (3)$$

where W_a^f , W_l^f , $(1/\partial a)$, $(1/\partial l)$, and λ denote axial, lateral displacements, the derivative in axial and lateral directions, and their corresponding weights, respectively. The axial derivative of the axial displacement is subtracted by its mean ($\langle \cdot \rangle$ denotes the mean value) to reduce the bias of the regularization similar to [15].

The second-order derivatives of the displacements have been found useful for USE [24], [26], [32]. Hence, they can be used to regularize the displacements

$$\text{loss}_s^2 = \lambda_{31} \Phi \left(\frac{\partial^2 W_a^f}{\partial^2 a} \right) + \lambda_{32} \Phi \left(\frac{\partial^2 W_a^f}{\partial a \partial l} \right) + \lambda_{41} \Phi \left(\frac{\partial^2 W_l^f}{\partial l \partial a} \right) + \lambda_{42} \Phi \left(\frac{\partial^2 W_l^f}{\partial^2 l} \right). \quad (4)$$

Unlike [25] that used the first- and second-order derivatives of only axial displacement, we used the first- and second-order derivatives of both axial and lateral displacement in both directions. The second-order derivatives do not introduce bias, but they require higher weights to be as effective as the first-order derivatives.

1) Hyperparameter Tuning: It can be seen that there are eight hyperparameters that we need to set before training the network. In our recent work [26], we set them empirically, while in this article, we tried to reduce the number of hyperparameters by relating some of them to others using US principles.

The distance between two adjacent samples in the axial and lateral directions is also vastly different since the sampling frequencies and the number of samples are widely dissimilar. The distance between two adjacent samples in the axial direction can be obtained by $c/(2f_s)$, where c denotes the sound speed and f_s is the sampling frequency. A rough approximation of the lateral distance between two samples would be the distance between two adjacent A-lines, which is much larger than the axial distance in a typical US image. Therefore, λ_{12} and λ_{22} must be several times smaller than λ_{11} and λ_{21} .

As noted in [32], the second-order derivatives are much smaller than the first-order ones. Therefore, to be as effective as the first-order derivatives, their weight should be several times larger than the first-order derivatives. We set this weight to be $\lambda_{31} = 5\lambda_{11}$. We also set the lateral derivative weights similar to the first-order derivatives ($\lambda_{41} = \beta\lambda_{31}$). Finally, the

smoothness regularizer can be written as

$$\begin{aligned} \text{loss}_s = & \Phi \left(\frac{\partial W_a^f}{\partial a} - \left\langle \frac{\partial}{\partial a} W_a^f \right\rangle \right) + \beta \Phi \left(\frac{\partial W_a^f}{\partial l} \right) \\ & + 0.5 \Phi \left(\frac{\partial W_l^f}{\partial a} \right) + 0.5 \beta \Phi \left(\frac{\partial W_l^f}{\partial l} \right) \\ & + 5 \left\{ \Phi \left(\frac{\partial^2 W_a^f}{\partial^2 a} \right) + \beta \Phi \left(\frac{\partial^2 W_a^f}{\partial a \partial l} \right) \right. \\ & \left. + 0.5 \Phi \left(\frac{\partial^2 W_l^f}{\partial l \partial a} \right) + 0.5 \beta \Phi \left(\frac{\partial^2 W_l^f}{\partial^2 l} \right) \right\} \quad (5) \end{aligned}$$

where β depends on the ratio of the sampling frequency in the axial and lateral directions. Setting the weights does not require an exact calculation of the sampling frequencies and should be less than 1; we set this hyperparameter to 0.1. Finally, for incompressible materials, Poisson's ratio is approximately 0.5 [33], which means that the strain in the lateral direction is half of the axial one. Therefore, λ_{21} can be substituted by $0.5\lambda_{11}$. It should be noted that the explained method to tune the weights of the regularizers is only a rough estimate of the optimal values; therefore, the training is not too sensitive to the variations of these weights and even changing the weights by as much as 100% yields similar results.

2) Bi-directional Strain Consistency: In this article, inspired by recent unsupervised methods in optical flow estimation [34], we proposed to utilize forward and backward consistency in addition to the data and smoothness losses. In unsupervised optical flow methods, the difference between forward and backward displacements was used for consistency loss [34].

Strain images are often showed in USE as a surrogate of the elastic modulus. Therefore, it would be useful to utilize the derivatives of the displacements for the consistency loss. Assuming a uniform tissue, the estimated forward and backward axial strains can be written as

$$\begin{aligned} \varepsilon^f &= \varepsilon_{gt} + \mathcal{N}(\mu, \sigma^2) \\ \varepsilon^b &= -\varepsilon_{gt} + \mathcal{N}(-\mu, \sigma^2) \end{aligned} \quad (6)$$

where we assumed that the strain true value is ε_{gt} and the error is modeled by a normal distribution with the bias and variance of μ and σ^2 , respectively. The forward and backward consistency losses can be defined as

$$\text{loss}_c = \Phi(\varepsilon^f + \varepsilon^b). \quad (7)$$

Substituting (6) into (7) yields

$$\text{loss}_c = \Phi(\mathcal{N}(0, \sigma^2/2)). \quad (8)$$

This equation indicates that minimizing loss_c results in reducing the variance in estimation of forward and backward strains. Similar to the smoothness loss (loss_s), the strain in both axial and lateral directions were employed, which can be written as

$$\text{loss}_c = \Phi \left(\frac{\partial W_a^f}{\partial a} + \frac{\partial W_a^b}{\partial a} \right) + 0.5 \beta \Phi \left(\frac{\partial W_l^f}{\partial l} + \frac{\partial W_l^b}{\partial l} \right) \quad (9)$$

where we used the same weights of the smoothness loss.

By using the loss functions defined in (1), (5), and (9), the total loss function can be written as

$$\text{loss} = \text{loss}_d + \lambda \text{loss}_s + \gamma \text{loss}_c. \quad (10)$$

Because of reducing the number of hyperparameters, only λ and γ should be tuned for the training, which can be done based on the training data. Too large values of λ and γ lead to a blurry strain image, while too small values result in noisy strain images. We set these hyperparameters ($\lambda = 0.03$ and $\gamma = 0.05$) by visually inspecting the strain images of the validation set after training with different values of λ and γ . It is worth mentioning that similar data and smoothness loss can be used for the backward flow. However, the consistency loss that we added has a similar behavior since it tries to make the backward strain close to the inverse of the forward one.

Adding the consistency loss demands high memory since both forward and backward flows are used for backpropagation. We utilized an NVIDIA A100 GPU with 40 GB of RAM to be able to train using the proposed loss function and simultaneously avoid downsampling to preserve RF data high-frequency information. Since training this network might be infeasible for some researchers, we have made the networks' weights available online at code.sonography.ai.

D. Data Collection

1) *Experimental Phantom*: We used a tissue-mimicking breast phantom made from Zerdine (Model 059, CIRS: Tissue Simulation and Phantom Technology, Norfolk, VA, USA) for data collection. The phantom contained a number of hard inclusions. The background has an elastic modulus of 20 kPa and the inclusions have at least twice elastic modulus of the background. This phantom was utilized to obtain training and test data. We made sure that different parts of the phantom were imaged for training and testing to avoid data leakage.

We employed Alpinion E-Cube R12 research US machine (Bothell, WA, USA) for training and test. The L3-12H linear array probe with the center frequency of 10 MHz and the sampling frequency of 40 MHz was utilized for image acquisition.

2) *In Vivo Data*: *In vivo* data were collected at Johns Hopkins Hospital using a research Antares Siemens system by a VF 10-5 linear array. The sampling frequency was 40 MHz and the center frequency was 6.67 MHz. Data were collected from patients with liver cancer during open-surgical RF thermal ablation. For more information, please see [8]. The institutional review board approved the study with the consent of the patients.

E. Training Schedule

We first collected data using the Alpinion machine from the breast phantom. We then selected image pairs having maximum axial displacement larger than three pixels. In total, 2200 image pairs were used for training. The networks were trained for 40 epochs and the learning rate was set to $30e^{-6}$ that is reduced by 1/2 every ten epochs. For *in vivo* data results, we also fine-tuned the network trained by the bi-directional method using *in vivo* data. This network was fine-tuned using 500 *in vivo* image pairs for 20 epochs and the learning rate

was $20e^{-6}$ and reduced by a factor of 2 every five epochs. In our experiments, we named this network as bi-directional unsupervised+ft.

F. Compared Methods

The compared methods are listed as follows.

- 1) OVERWIND is an optimization-based method that estimates subpixel displacement. This method requires the initial displacement, which is obtained by DP [8]. OVERWIND considers a window around each sample and uses total variation for the regularization. This method obtains high-quality strain images [15].
- 2) The recently proposed network MPWC-Net++ that is the modified version of MPWC-Net. This network is only trained on computer vision images and no training on US data is done [22].
- 3) We fine-tune MPWC-Net++ using the unsupervised technique without the consistency loss. Also, similar to [26], it has the second-order derivatives only in the direction of the displacement ($\lambda_{32}, \lambda_{41} = 0$). In this case, the unsupervised fine-tuning would be similar to the semisupervised method [26] with some minor improvements. The difference is that a better network (MPWC-Net++) with more suitable regularization weights is employed.
- 4) Our proposed bi-directional unsupervised fine-tuning method. In this method, the consistency loss is added to the unsupervised loss function and all the second-order derivatives are employed in the smoothness loss.
- 5) For *in vivo* data section, we also fine-tune the bi-directional network using *in vivo* data.

We compare our bi-directional semisupervised method with recent methods in USE: OVERWIND is a high-performance and nondeep learning method. MPWC-Net++ is one of the best networks used for USE without training on US data. The unsupervised variant of MPWC-Net++ combines this high-performance network with the unsupervised fine-tuning [26].

III. RESULTS

A. Quantitative Metrics

Contrast-to-noise ratio (CNR) and strain ratio (SR) are two popular metrics that have been used to assess the performance of the elastography methods. These metrics are suitable for experimental phantom and *in vivo* data where the ground-truth strain is unknown. Windows around the target and background are chosen to compute these metrics. CNR and SR can be obtained by [5]

$$\text{SR} = \frac{\bar{s}_t}{\bar{s}_b}, \quad \text{CNR} = \sqrt{\frac{2(\bar{s}_b - \bar{s}_t)^2}{\sigma_b^2 + \sigma_t^2}} \quad (11)$$

where \bar{s}_t and \bar{s}_b are the mean of the strain in the selected target and background regions, respectively, and σ_t and σ_b are the standard deviations of the target and the background regions, respectively. Assuming that the target strain is to be lower than the background region, lower SR represents a higher difference

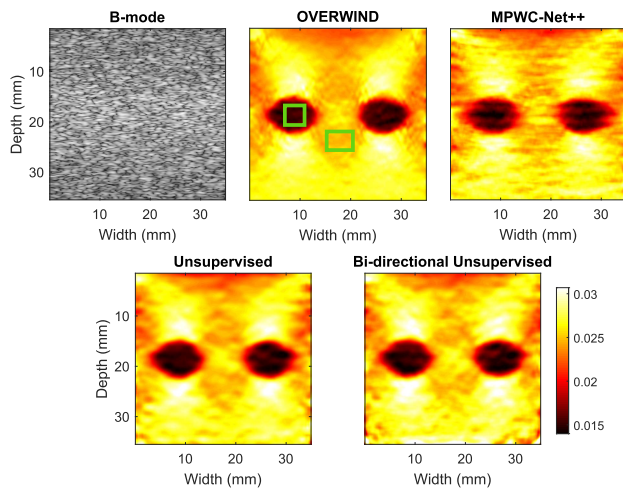


Fig. 2. Phantom result 1 with the maximum strain value of 3%. Green boxes indicate windows for computing CNR and SR.

between the mean strain value of the target and the background region and is therefore desired. CNR provides some notion of detectability, which combines the difference in mean between two samples is detectable and their background noise.

Instead of selecting the target and background windows and calculating a single value for SR and CNR, thousands of SR and CNR values are calculated as follows. First, two large background and target windows are selected. We then select small patches within these windows and calculate SR and CNR for different combinations of target and background windows. The number of these small patches should be large enough to produce statistically reliable estimates of SR and CNR. It should be mentioned that we selected large windows in regions with uniform strains (as marked in the figures) and small overlapping patches inside these windows to compute the CNR and SR. We then reported the mean and standard deviation of these values similar to our previous works [21], [22], [26].

B. Experimental Phantom Results

The results of different parts and compression levels of the experimental phantom are shown in Figs. 2–5. In Fig. 2, the inclusions are not visible in the B-mode images, while they can be detected by USE methods. Comparing deep learning methods, unsupervised training substantially improves the strain quality of MPWC-Net++. Our proposed bi-directional method obtains similar or higher quality strain images compared to the unsupervised method and substantially better results in all experiments compared to MPWC-Net++, especially when the compression is low (e.g., Figs. 4 and 5). It can be seen that for those images, MPWC-Net++ does not provide a clear image of the inclusion, while the proposed method obtains the highest quality strain images among the compared deep learning methods. OVERWIND obtains high-quality strain images and the proposed method performs comparably to OVERWIND in terms of strain quality.

The quantitative results are listed in Tables I (for CNR) and II (for SR). In terms of CNR,

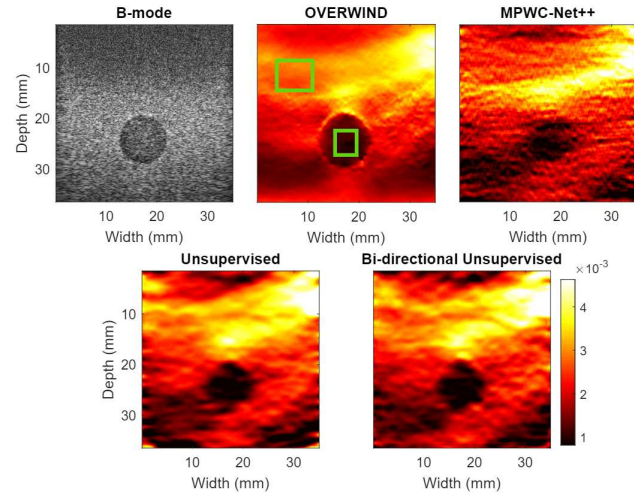


Fig. 3. Phantom result 2 with the maximum strain value of 0.4%. Green boxes indicate windows for computing CNR and SR.

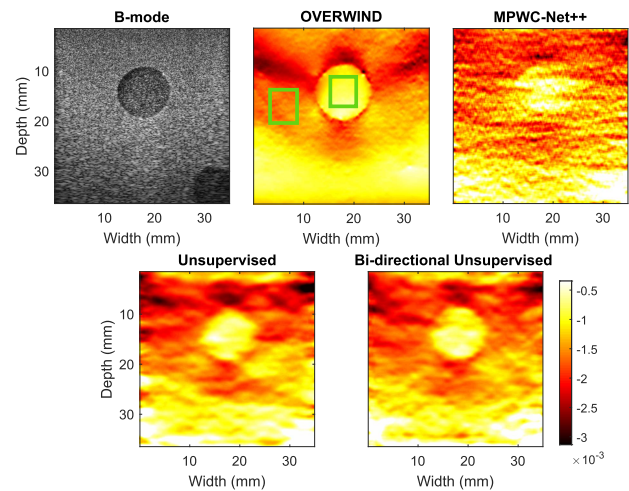


Fig. 4. Phantom result 3 with the maximum strain value of 0.3%. Green boxes indicate windows for computing CNR and SR.

our proposed method substantially increases the CNR of MPWC-Net++ and outperforms the unsupervised method in most cases. To be more specific, the bi-directional unsupervised method increases the CNR of MPWC-Net++ from 12.02, 8.74, 5.12, 3.73, and 10.25 to 27.71, 17.19, 13.84, 12.82, and 21.40. It also obtains CNR values close to OVERWIND or even better (in Fig. 4).

The SR results agree with [21] where we showed that MPWC-Net has better SR compared to another optimization-based method. In most cases (except Fig. 3), a bi-directional unsupervised method has the better SR and is close to the lowest ones in that case.

The visual and quantitative results of the experimental phantoms confirm that our proposed method improves the overall quality of the strain images.

1) *Smoothing Window Effect on Strain Image*: After displacement estimation, a smoothing window along with the derivative kernel is used to reduce the error and compute the derivative of the displacement. Larger windows smooth the displacement more but sacrifice the resolution of the

TABLE I

CNR RESULTS (HIGHER IS BETTER). THE BOLD FONT HIGHLIGHTS THE BEST, AND THE UNDERLINE INDICATES THE BEST DEEP LEARNING RESULTS. NUMBERS MARKED WITH ASTERISKS INDICATE RESULTS THAT ARE NOT STATISTICALLY SIGNIFICANT (p -VALUE > 0.01), E.G., OVERWIND AND THE PROPOSED METHOD IN FIG. 2

	Fig. 2	Fig. 3	Fig. 4	Fig. 5 (1)	Fig. 5 (2)
OVERWIND	27.26±4.27*	25.28±6.62	12.40±2.38	12.83±5.35*	27.84±8.82
MPWC-Net++	12.02±1.59	8.74±1.85	5.12±1.11	4.83±2.09	7.82±3.66
Unsupervised	31.78±7.47	15.12±4.12	9.94±1.64	8.67±4.09	14.37±4.27
Bi-directional Unsupervised	27.71±5.20*	17.19±4.45	13.84±4.49	12.82±4.83*	21.40±3.69

TABLE II

SR(%) RESULTS (LOWER IS BETTER). THE BOLD FONT HIGHLIGHTS THE BEST, AND THE UNDERLINE INDICATES THE BEST DEEP LEARNING RESULTS. NUMBERS MARKED WITH ASTERISKS AND STAR INDICATE RESULTS THAT ARE NOT STATISTICALLY SIGNIFICANT (p -VALUE > 0.01)

	Fig. 2	Fig. 3	Fig. 4	Fig. 5 (1)	Fig. 5 (2)
OVERWIND	62.26±0.71	40.21±3.2	50.04±3.08*	61.78±5.74*	36.56±2.00
MPWC-Net++	61.56±1.70*	35.08±6.26	49.29±4.41*	69.43±8.46	40.20±10.3
Unsupervised	61.27±0.67*	25.89±3.97	48.15±6.16**	60.96±9.67*	32.06±7.47
Bi-directional Unsupervised	59.24±0.50	28.12±4.12	45.82±2.53	58.79±5.72	30.32±2.90

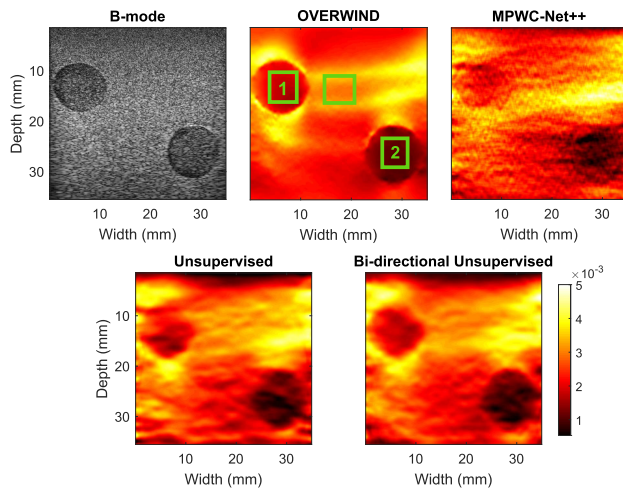


Fig. 5. Phantom result 4 with two inclusions having different elasticities with the maximum strain value of 0.5%. Green boxes indicate windows for computing CNR and SR.

strain image. Therefore, displacement estimation methods that require smaller windows are preferred. When a USE displacement estimation method does not require a large smoothing window, it shows that the method produces a displacement map with a low variance error. To compare the methods, we compute the strain image of two image pairs with smoothing windows of sizes 5, 15, 30, and 40. The CNR values of different smoothing window lengths are shown in Fig. 6. It can be seen that OVERWIND has high CNR values even when the smallest smoothing window is employed. Unsupervised and bi-directional unsupervised methods have good CNR values close to that of OVERWIND. It should be mentioned that the difference between the unsupervised and bi-directional methods in Fig. 6 was not statistically significant (p -value = 0.112). MPWC-Net++ has very low CNR when

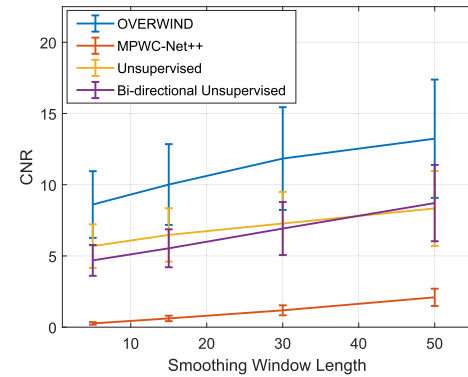


Fig. 6. CNR values of the compared method using different smoothing window lengths. The strain images are shown in Fig. 7.

the smoothing window is small. It indicates that this method is highly sensitive to the length of smoothing window and requires larger ones to produce acceptable strain images, whereas OVERWIND and the two unsupervised methods do not need a large smoothing window to produce reliable strain images and have low variance errors. The strain images are shown in Fig. 7 for smoothing windows of 5 (top), 15 (middle), and 30 (bottom), and the target and background windows for computation of CNR are highlighted. We can see that MPWC-Net++ generates noisy strain images for small smoothing windows, where the inclusion is not visually detectable. However, both unsupervised fine-tuning methods provide a better performance, close to OVERWIND, and generate less variations compared to MPWC-Net++.

C. Lateral Strain

The lateral strain ($\frac{\partial W_l}{\partial l}$) has much lower quality than the axial strain since the main movement is in the axial direction, and the lateral sampling frequency and resolution are low. The

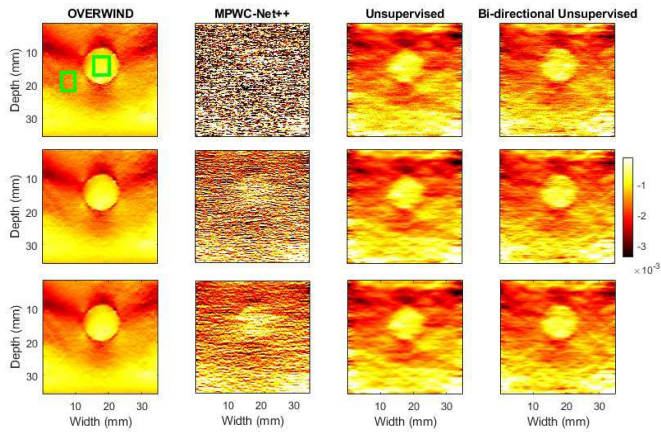


Fig. 7. Strain images of the compared methods. Smoothing window size is 5 (top), 15 (middle), and 30 (bottom).

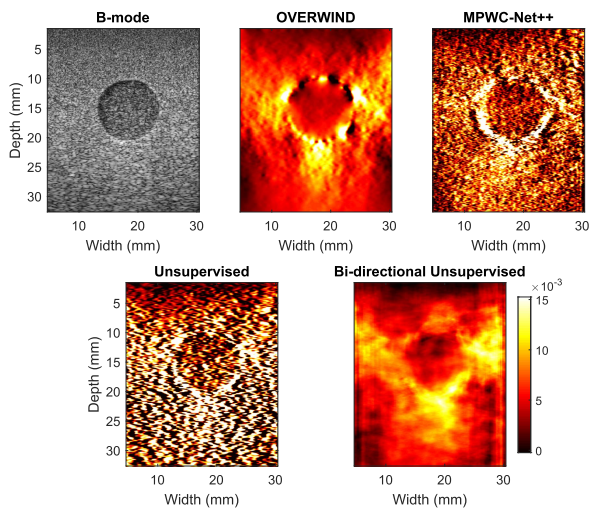


Fig. 8. Lateral strain results. The proposed bi-directional method provides a smooth strain image close to OVERWIND, while MPWC-Net++ and the unsupervised method generate noisy strain images and the inclusion is barely visible.

lateral strain can be utilized in inverse problem methods to find the elastic modulus [35]. Fig. 8 shows the lateral strain obtained by the compared methods (refer to the Supplementary Materials for the axial strain). It can be seen that MPWC-Net++ and the unsupervised method obtain very noisy strain images and the inclusion is hardly visible. However, the proposed bi-directional method and OVERWIND obtain acceptable strain images and the inclusion can be detected.

D. In Vivo Results

Compared methods are evaluated with two *in vivo* data belonging to two patients. We also fine-tuned the bi-directional network using *in vivo* data to find out whether further improvements can be achieved. The strain images of the compared methods are given in Fig. 9 and 10. OVERWIND produces high-quality strain images with low noise, while the strain images obtained by MPWC-Net++ have some oversmoothing, especially in the lateral direction. Both unsupervised methods substantially improve the strain image qualities of MPWC-Net++. The bi-directional+ft also

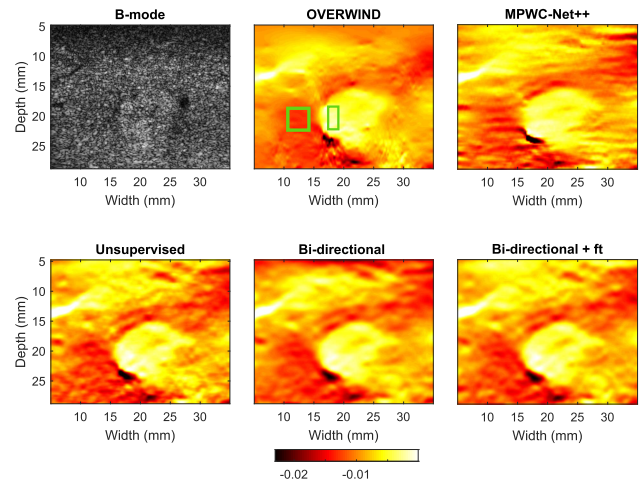


Fig. 9. Strain images of *in vivo* data 1. The tumor has a lower absolute strain value but looks brighter since the strain is negative.

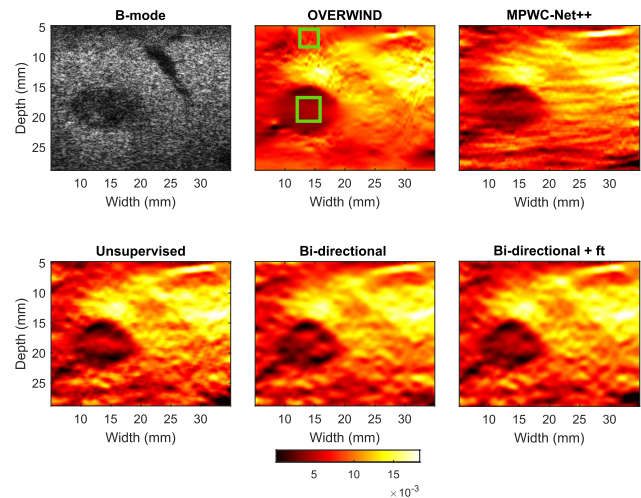


Fig. 10. Strain images of *in vivo* data 2. The tumor has a lower absolute strain value and looks darker since the strain is positive.

obtains high-quality strain images, but the difference with the bi-directional method is not discernible.

The quantitative results are given in Table III. OVERWIND achieves the highest CNR for *in vivo* data 1, while bi-directional unsupervised+ft has the best CNR for *in vivo* data 2. It can also be seen that fine-tuning on *in vivo* data does not result in considerable CNR improvement (it has slightly better CNR than bi-directional for *in vivo* data 2 and worse CNR for *in vivo* data 1). In terms of SR, bi-directional and bi-directional+ft have the best SR values. MPWC-Net++ and OVERWIND have the highest SR among the compared methods.

IV. DISCUSSION

In this article, we employed semisupervised training to improve the performance of an optical flow network for USE. Although we used MPWC-Net++, which outperformed other networks for USE, the training method can be applied to other networks as well. It should be mentioned that the optical flow networks usually have a pyramidal structure, meaning that the displacements are estimated in different resolutions.

TABLE III

QUANTITATIVE RESULTS OF *in vivo* DATA. THE PAIRS MARKED BY ASTERISK OR STAR ARE NOT STATISTICALLY SIGNIFICANT (p -VALUE > 0.01)

	<i>In vivo</i> data 1		<i>In vivo</i> data 2	
	CNR	SR(%)	CNR	SR(%)
OVERWIND	17.28±5.31	21.80±4.16	7.18±1.58*	51.40±6.06*
MPWC-Net++	11.51±3.08	25.50±6.63	7.31±2.47*	48.89±11.61*
Unsupervised	11.91±2.62	19.20±5.30	6.73±3.09	47.83±20.83**
Bi-directional Unsupervised	<u>16.27±5.26</u>	19.35±5.46	7.91±3.18	45.60±11.83
Bi-directional Unsupervised + ft	14.37±4.40	19.19±6.10	8.86±2.64	46.14±12.91

TABLE IV

SSIM (%) OF STRAIN IMAGES OF OVERWIND VERSUS DEEP LEARNING METHODS

	Fig. 2	Fig. 3	Fig. 4	Fig. 5	<i>in vivo</i> 1	<i>in vivo</i> 2
MPWC-Net++	90.12	73.36	60.84	90.77	92.66	93.79
Unsupervised	98.96	95.56	94.86	96.11	90.08	91.13
Bi-directional Unsupervised	98.89	96.27	95.76	96.90	93.57	96.62

Similar to previous works [21], [26] and unsupervised optical flow works [30], we only used the last output resolution for fine-tuning.

The optical flow CNNs trained on computer vision images do not employ regularizations due to abrupt changes in scenes such as a moving car in front of a fixed background. However, the displacement in USE is usually smooth without any sudden changes. The effect of the absence of the regularization can be seen in Fig. 7 (top), where the smoothing window is very small. The strain estimated by MPWC-Net++ has a high variance, whereas the bi-directional unsupervised method provides smooth and high-quality strain images by incorporating smoothness and consistency constraints in the estimated displacement.

The values reported in Tables I–III are the mean and standard deviation of CNR and SR values. We have conducted a statistical analysis to find out whether the difference between obtained values is statistically significant. We employ the Friedman test [36], and the p -values are given in the Supplementary Materials.

Regarding the choice of weights, the explained method of tuning the weights only gives a rough estimate of the optimum weights. However, the output strain image is not considerably sensitive to these weights and similar performance can be obtained by different weights.

To consider texture (which is not captured by SR or CNR), structural similarity index (SSIM) [37] has been used to evaluate the texture similarity between OVERWIND and the deep learning methods. The results are provided in Table IV. It is clear that the proposed bi-directional method achieves a higher SSIM score compared to the other methods, which indicates that it produces strain images closer to OVERWIND than the other compared methods.

Running time is another important aspect that needs to be investigated. The deep learning methods shine in this aspect, and our proposed method can provide high-quality strain images close to OVERWIND for real-time applications. To give a general view about the computation time, OVERWIND takes 26 s for an image pair of

size 1920×384 on CPU (eighth generation, core i7). It should be noted that this is a MATLAB implementation, and an optimized implementation in C will be much faster. MPWC-Net++ and bi-directional MPWC-Net++ take 0.166 and 0.174 s on NVIDIA A100 GPU, respectively. The bi-directional variant of MPWC-Net++ takes slightly more time than MPWC-Net++ since it estimates both forward and backward displacements; however, it still takes much less time than running the network two times and computing forward and backward displacements separately.

V. CONCLUSION

In this article, we proposed a bi-directional semisupervised deep learning method. We used strain consistency along with data and smoothness loss. We also employed the second-order derivatives regularization of axial and lateral displacements in both directions. Furthermore, we reduced the number of hyperparameters by relating some of them to others by considering the underlying principles of the US. We showed that our proposed method substantially improved current optical flow networks used for USE. We validated our method using different experimental phantom and *in vivo* data. Our proposed method obtained strain images close to OVERWIND.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the handling associate editor for constructive comments.

REFERENCES

- [1] A. Kling and J. Jiang, "Potential of determining thermal dose for ablation therapies using ultrasound elastography: An *ex vivo* feasibility study," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Oct. 2018, pp. 1–4.
- [2] F. Lee *et al.*, "Evaluating HIFU-mediated local drug release using thermal strain imaging: Phantom and preliminary *in-vivo* studies," *Med. Phys.*, vol. 46, no. 9, pp. 3864–3876, Sep. 2019.
- [3] T. J. Hall, Y. Zhu, C. S. Spalding, and L. T. Cook, "In vivo results of real-time freehand elasticity imaging," in *Proc. IEEE Ultrason. Symp.*, vol. 2, Oct. 2001, pp. 1653–1657.
- [4] J. Bercoff, M. Tanter, and M. Fink, "Supersonic shear imaging: A new technique for soft tissue elasticity mapping," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 51, no. 4, pp. 396–409, Apr. 2004.
- [5] J. Ophir *et al.*, "Elastography: Ultrasonic estimation and imaging of the elastic properties of tissues," *Proc. Inst. Mech. Eng., H, J. Eng. Med.*, vol. 213, no. 3, pp. 203–233, 1999.
- [6] T. Varghese, E. E. Konofagou, J. Ophir, S. K. Alam, and M. Bilgen, "Direct strain estimation in elastography using spectral cross-correlation," *Ultrasound Med. Biol.*, vol. 26, pp. 1525–1537, Nov. 2000.
- [7] C. Schneider, A. Baghani, R. Rohling, and S. Salcudean, "Remote ultrasound palpation for robotic interventions using absolute elastography," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Berlin, Germany: Springer, 2012, pp. 42–49.

- [8] H. Rivaz, E. M. Bocror, M. A. Choti, and G. D. Hager, "Real-time regularized ultrasound elastography," *IEEE Trans. Med. Imag.*, vol. 30, no. 4, pp. 928–945, Apr. 2011.
- [9] A. Nahiyan and M. K. Hasan, "Hybrid algorithm for elastography to visualize both solid and fluid-filled lesions," *Ultrasound Med. Biol.*, vol. 41, no. 4, pp. 1058–1078, Apr. 2015.
- [10] 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.
- [11] J. Jiang and T. J. Hall, "A parallelizable real-time motion tracking algorithm with applications to ultrasonic strain imaging," *Phys. Med. Biol.*, vol. 52, no. 13, p. 3773, 2007.
- [12] M. Mirzaei, A. Asif, M. Fortin, and H. Rivaz, "3D normalized cross-correlation for estimation of the displacement field in ultrasound elastography," *Ultrasonics*, vol. 102, Mar. 2020, Art. no. 106053.
- [13] T. J. Hall *et al.*, "Recent results in nonlinear strain and modulus imaging," *Current Med. Imag. Rev.*, vol. 7, no. 4, pp. 313–327, 2011.
- [14] H. S. Hashemi and H. Rivaz, "Global time-delay estimation in ultrasound elastography," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 64, no. 10, pp. 1625–1636, Oct. 2017.
- [15] M. Mirzaei, A. Asif, and H. Rivaz, "Combining total variation regularization with window-based time delay estimation in ultrasound elastography," *IEEE Trans. Med. Imag.*, vol. 38, no. 12, pp. 2744–2754, Dec. 2019.
- [16] B. Peng, Y. Xian, Q. Zhang, and J. Jiang, "Neural-network-based motion tracking for breast ultrasound strain elastography: An initial assessment of performance and feasibility," *Ultrason. Imag.*, vol. 42, no. 2, pp. 74–91, Mar. 2020.
- [17] 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.
- [18] M. G. Kibria and H. Rivaz, "GLUENet: Ultrasound elastography using convolutional neural network," in *Simulation, Image Processing, and Ultrasound Systems for Assisted Diagnosis and Navigation*. Cham, Switzerland: Springer, 2018, pp. 21–28.
- [19] M. G. Kibria and H. Rivaz, "Global ultrasound elastography using convolutional neural network," 2018, *arXiv:1805.07493*.
- [20] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.
- [21] 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.
- [22] A. K. Z. Tehrani and H. Rivaz, "MPWC-Net++: Evolution of optical flow pyramidal convolutional neural network for ultrasound elastography," *Proc. SPIE*, vol. 11602, Feb. 2021, Art. no. 1160206.
- [23] A. K. Z. Tehrani, M. Amiri, and H. Rivaz, "Real-time and high quality ultrasound elastography using convolutional neural network by incorporating analytic signal," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 2075–2078.
- [24] R. Delaunay, Y. Hu, and T. Vercauteren, "An unsupervised approach to ultrasound elastography with end-to-end strain regularisation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2020, pp. 573–582.
- [25] R. Delaunay, Y. Hu, and T. Vercauteren, "An unsupervised learning approach to ultrasound strain elastography with spatio-temporal consistency," *Phys. Med. Biol.*, vol. 66, no. 17, Sep. 2021, Art. no. 175031.
- [26] A. K. Tehrani, M. Mirzaei, and H. Rivaz, "Semi-supervised training of optical flow convolutional neural networks in ultrasound elastography," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2020, pp. 504–513.
- [27] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8981–8989.
- [28] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5754–5763.
- [29] C.-L. Hung, C.-F. Hsin, H.-H. Wang, and C. Y. Tang, "Optimization of GPU memory usage for training deep neural networks," in *Proc. Int. Symp. Pervasive Syst., Algorithms Netw.* Cham, Switzerland: Springer, 2019, pp. 289–293.
- [30] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova, "What matters in unsupervised optical flow," 2020, p. 3, vol. 1, no. 2, *arXiv:2006.04902*.
- [31] J. T. Barron, "A general and adaptive robust loss function," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4331–4339.
- [32] M. Ashikuzzaman, A. Sadeghi-Naini, A. Samani, and H. Rivaz, "Combining first- and second-order continuity constraints in ultrasound elastography," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 68, no. 7, pp. 2407–2418, Jul. 2021.
- [33] E. J. Chen, J. Novakofski, W. K. Jenkins, and W. D. O'Brien, "Young's modulus measurements of soft tissues with application to elasticity imaging," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 43, no. 1, pp. 191–194, Jan. 1996.
- [34] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 7251–7259.
- [35] N. Mohammadi, M. M. Doyley, and M. Cetin, "Ultrasound elasticity imaging using physics-based models and learning-based plug-and-play priors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 1165–1169.
- [36] E. Theodorsson-Norheim, "Friedman and quade tests: BASIC computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples," *Comput. Biol. Med.*, vol. 17, no. 2, pp. 85–99, Jan. 1987.
- [37] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1488–1499, Apr. 2012.