Program Digest, 2019 IEEE International Ultrasonics Symposium (IUS)
Glasgow, Scotland, October 6-9, 2019

Basal Strain Estimation in Transesophageal Echocardiography (TEE) using Deep Learning based Unsupervised Deformable Image Registration

1st Torjus Haukom
Department of Electronic Systems
Norwegian Univ. of Science and Technology
Trondheim, Norway
torjuh@stud.ntnu.no

2nd Erik Andreas Rye Berg
Center for Innovative Ultrasound Solutions
Norwegian Univ. of Science and Technology
Clinic of Cardiology
St. Olavs hospital
Trondheim, Norway
erik.a.berg@ntnu.no

3rd Svend Aakhus
Department of Circulation and Medical Imaging
Norwegian Univ. of Science and Technology
Trondheim, Norway
s-aakhus@online.no

4th Gabriel Hanssen Kiss
Operating Room of the Future
St. Olavs hospital
Center for Innovative Ultrasound Solutions
Norwegian Univ. of Science and Technology
Trondheim, Norway
gabriel.kiss@stolav.no

***Abstract*—As of today, perioperative monitoring in the operating room is based on vital signs and clinical observations by the anesthesiologist. This, however, does not offer a complete monitoring of left ventricular function throughout the intervention. We hypothesize that functional monitoring of the heart can be performed automatically based on transesophageal echocardiographic (TEE) images. The main goal of this work is therefore to compute the non-linear deformation between subsequent images in a TEE sequence of the left ventricle and estimate basal longitudinal strain to assess regional myocardial function using a deep learning approach. An unsupervised approach based on a convolutional neural network (CNN) was implemented. The output of the CNN network is a dense vector field that describes the non-linear deformation required to maximize the similarity between two images.**

Recordings from 42 consecutive complete TEE exams from the Echocardiography Unit were anonymized and used for training. Recordings from 5 consecutive TEE exams performed during heart surgery, also anonymized, were used as test set. For the test set patients, the basal strain was manually annotated in EchoPac by an expert echocardiographer. For this work, 19 heart cycles were randomly selected from the test set and checked to ensure visibility of the points to be tracked. Overall when estimating strain there was a mean difference of $7.25\%(\pm 4.56\%)$ in the strain values when compared to expert annotations.

We have proven that point tracking is working as expected in most images, where the myocardium is well depicted. However, dropouts, noise generated by implants or air bubbles after surgery, confuse the tracker and drifting occurs.

***Index Terms*—patient monitoring, deep learning, deformable image registration**

I. Introduction

Cardiac surgery is a complex and comprehensive intervention and is not without risks. Procedures such as bypass surgery and valve replacements may negatively impact cardiac function, often causing decreased myocardial contractility and in some cases, atrial fibrillation and myocardial infarction [1]. Consequently, patients undergoing such procedures have their cardiac function and hemodynamics monitored through the perioperative phase. Currently, this monitoring is done by evaluating vital signs, such as blood pressure, heart rate, blood oxygen level and respiratory rate, as well as through manual TEE assessment of the cardiac function.

Traditionally, TEE assessment of cardiac function in the perioperative period has been performed by simple visual inspection of the ultrasound recordings. In the last decade, however, efforts have been made to standardize quantitative indicators of cardiac function, as they rely less on the individual echocardiographer's experience and preferences [2]. Myocardial strain is one such quantitative indicator, measuring global or regional myocardial deformation, with prognostic value in patients undergoing cardiac surgery [3]. Various methods for deriving strain values have been commercialized and are widely used today. They include tissue Doppler imaging (TDI) and subsequent integration of strain rate values along the myocardium or 2D/3D speckle tracking approaches. However, as of now most of the strain estimators require trained personnel to manually annotate landmarks or ventricle contours in the images for the tracking to work. This makes strain estimation a time and resource consuming task, less suitable for repeated monitoring in the operating theater. Efforts have been made to provide fully automatic strain measurements. Knackstedt et al. demonstrated the reliability of the AutoLV algorithm (TomTec-Arena 1.2, TomTec Imaging

978-1-7281-4595-2/19/$31.00 ©2019 IEEE

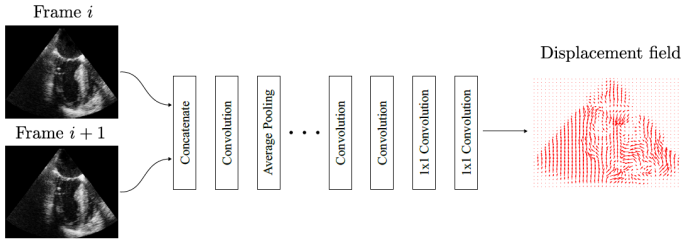

Fig. 1. Architecture of the convolutional neural network used. As input, it expects two consecutive frames from an ultrasound sample. The output is a low-resolution displacement field to be interpolated. Several alternating convolutional and average pooling layers may be added before the final convolutions to achieve the desired spacing between the B-spline control points.

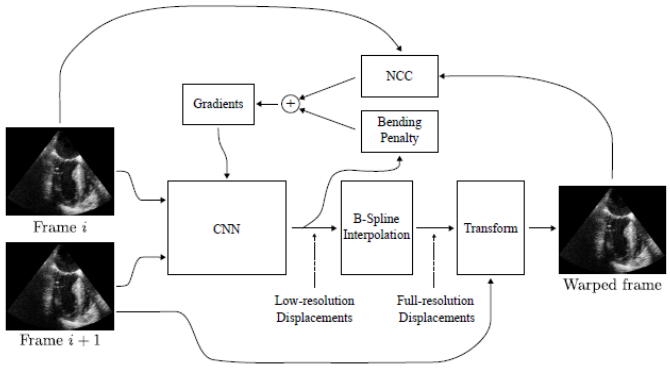

Fig. 2. Training procedure for the motion estimation network. The CNN produces a low resolution displacement field, which is then interpolated and used to warp frame i + 1 into frame i. Frame i and the warped frame is then compared using normalized cross-correlation. A differential based bending penalty is calculated from the low-resolution displacements and added to the cross-correlation to form the loss function. The loss is differentiated, and the gradients are used to update the CNN parameters.

Systems, Unterschleissheim, Germany), which detects the contour of the myocardium, to measure global longitudinal strain (GLS) in transthoracic images. Their experiments showed a high correlation with manual methods. A more recent approach aimed at on-site analysis, proposed by Østvik et al., uses supervised deep learning to classify the view, crop the samples to the myocardium, and track its motion through the cardiac cycle and estimate GLS. The motion estimation was performed using a neural network trained on a synthetic dataset, while the view classification and cropping networks were trained on manually annotated data. Their results are promising but still preliminary.

The main aim of the present work is to derive the non-linear deformation between subsequent images in a TEE sequence of the left ventricle and estimate basal longitudinal strain in order to assess regional myocardial function.

II. Material and Methods

A. Patient data

For training and evaluation of the strain estimation pipeline, TEE B-mode images were obtained by cardiologists with echocardiographic expertise from 47 patients using GE Vivid E95 and E9 systems with a 6VT-D probe (GE Vingmed Ultrasound, Horten, Norway). 42 of these patients were examined in the clinic for diagnostic purposes, and five patients were examined before and after undergoing cardiac surgery (coronary artery bypass grafting in four cases, mitral valve clipping in one case). These were consecutive cases, no selection based on image quality was performed. At least three complete cardiac cycles were captured in three views: 4-chamber, 2-chamber, and long-axis. The frame rate of the recordings was in the range of 30 to 60 frames per second, and the resolution ranged from 255x180 to 537x380 depending on the width and depth of the scan. All samples were anonymized before analysis, and to facilitate processing, the images were converted from the proprietary DICOM format to 2D images by applying a polar-Cartesian transform on the raw B-mode lines.

B. Data pre-processing

The data was divided into three separate datasets. A training set consisting of samples from 32 patients chosen randomly was used for training the CNN. For hyperparameter tuning and to monitor the model performance during training a validation set consisting of the samples from 10 patients was used. In both of these datasets, the frames were zero padded to match the resolution of the sample with the highest resolution within each set to enable training on batches. All frames of the samples in the training and validation sets were organized into pairs of consecutive frames. During training, these pairs were drawn randomly to construct batches. The samples from the remaining 5 patients were used for testing. As the goal of the method was to track points through the cardiac cycle, the frames of these samples were kept in order. The test set samples were divided into smaller samples showing a single cardiac cycle. Since, the location of two mitral attachment points was assumed to be known, they were manually annotated in the initial frame (end-diastole) of each of the divided samples. All datasets were preprocessed by applying a proprietary contrast enhancement algorithm, and all pixels were scaled to a range of [0, 1].

C. CNN based motion estimation and landmark tracking

To track the landmarks on the basal segment, we implemented a CNN network similar to the one proposed by de Vos et al. [4]. At the heart of the method is a CNN that takes two consecutive frames I_i and I_{i+1} from an ultrasound sequence, performs a deformable image registration between them and outputs a low-resolution displacement field $\vec{D}$ describing the motion between the two frames in x and y directions. This displacement field is then upsampled using cubic B-splines to make a dense displacement field $\vec{D}^d$ with one motion vector per pixel such that:

$$I_i(x,y) \approx I_{i+1}(x + D_x^d(x,y), y + D_y^d(x,y)) \qquad (1)$$

Thus, to perform point tracking, one can follow the displacement vector from frame to frame. The CNN architecture, shown in Figure 1, consists of a concatenation layer, merging

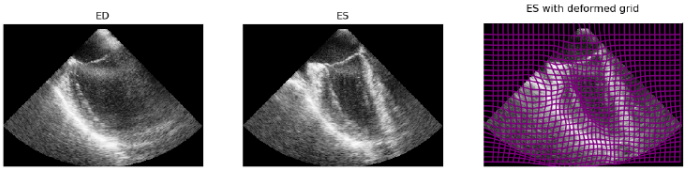

Fig. 3. Frames from a sample recording at ED and ES, including a deformed grid illustrating the deformation between these frames.

the two consecutive frames into one tensor, followed by alternating convolutional and average pooling layers. The number of pooling layers determines the resolution of the displacement field and thus, the number of B-spline control points. These low-resolution feature maps are passed through two more convolutional layers before finally two 1 x 1 convolutions are performed yielding the estimated displacements.

In Figure 2, a flow chart visualization of the training procedure for the motion estimator is shown. Following this procedure, consecutive frame pairs are fed to the CNN, which produce low-resolution displacement fields. These fields are interpolated using B-splines and used to warp the second frame. Then the warped frame is compared to the first frame using normalized cross-correlation. This approach has two major advantages. The neural network consists of only convolutional layers, meaning that the trained network can make estimates on frames of any resolution, and the training is performed unsupervised, eliminating the need for costly ground truth annotation. In essence, this means that the method can be repurposed for any type of images, medical or other, with minimal adjustments.

To ensure spatial smoothness of the displacements, a differential based bending penalty can be added to the negated cross-correlation for regularization. This sum forms the loss function used to optimize the parameters of the network. The bending penalty P, given in equation 2, minimizes the second order spatial derivatives of the displacements. This ensures that the transformation is locally affine, meaning that the transformation is globally smooth. The scaling factor λ controls the amount of regularization.

$$P = \lambda \sum_{x,y \in I} \left(\frac{\partial^2 \vec{D}^d}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \vec{D}^d}{\partial y^2} \right)^2 + 2 \left(\frac{\partial^2 \vec{D}^d}{\partial x \partial y} \right)^2 \quad (2)$$

B-splines were chosen as the interpolation method because they are controlled locally. That means that a change in one control point only affects the neighborhood around it in the resulting image. As strain is a measure of local deformation, this is a significant advantage compared to other popular interpolation methods such as thin plate splines. Another advantage is that the k-th derivative of a B-spline of degree n is simply a B-spline of degree $n - k$. This means that the differentials needed for the bending penalty can be calculated by interpolating $\vec{D}$ using linear and quadratic B-splines [5].

D. Implementation details

The motion estimation was implemented in the Python programming language, using Tensorflow version 1.10 with eager execution enabled. Readers interested in the source code are referred to the Github repository:
$https://github.com/torjush/Strain\_estimation$.

All convolutional layers except for the output layer consisted of 32 filters, and batch normalization and ReLU activations were applied. The last 1 x 1 convolutional layer consisted of two filters and was left unconstrained with no activation to freely estimate the displacements. Average pooling was done over 2 x 2 neighborhoods with a stride of 2, thus downsampling the features by 2 in each pooling layer. Zero-padding was performed for all convolution and pooling layers to ensure the right dimensions of the dense displacement field.

To increase performance, the B-spline interpolation was implemented using fractionally strided convolutions, in which zeros are placed between each pixel in the image to achieve the desired dimensions. A B-spline kernel may then be constructed in order to weight the original samples appropriately and as a consequence to produce an interpolated image through a convolution operation.

Because the downsampling factor is determined by the number of pooling layers in the network, the B-spline kernels for both the interpolation and the computation of the bending penalty may be precomputed, leaving only the convolution to be performed at run-time. The regularization parameter was set to $\lambda = 5 \cdot 10^6$, and was chosen by trial and error. All weights were initialized using the Glorot Uniform initializer and optimized using the Adam optimizer with a learning rate $\alpha = 10^{-4}$. The batch size used was 16 pairs of consecutive frames. The training was performed on a Tesla K80 GPU (Nvidia, Santa Clara, California).

III. Results

Figure 3 shows the result of the motion estimation task. A coordinate grid is warped by the estimated motion vectors between the end-diastolic (ED) and end-systolic (ES) frames of a recording, resulting in a deformed grid illustrating the movement between these points in time.

To assess the results of the strain estimation pipeline as a whole, reference values for basal longitudinal strain were provided by a trained physician. The reference values were acquired by manually annotating the images and tracking the myocardium using the EchoPAC (GE Vingmed Ultrasound, Horten, Norway) speckle tracking software. With both methods, basal end-systolic strain was measured for all cardiac cycles available in the recordings and then averaged. For the anteroseptal segment in the long-axis view, the strain estimates produced by the CNNs are made by tracking two points on either side of the left ventricular outflow tract, whereas the reference values are made tracking points further down on the myocardium. Overall when estimating basal strain, there was a mean difference of $7.25\% \pm 4.56\%$. Figure 4 presents the ground truth vs CNN estimated values in 19 heart cycles from 5 patients that underwent cardiac surgery.

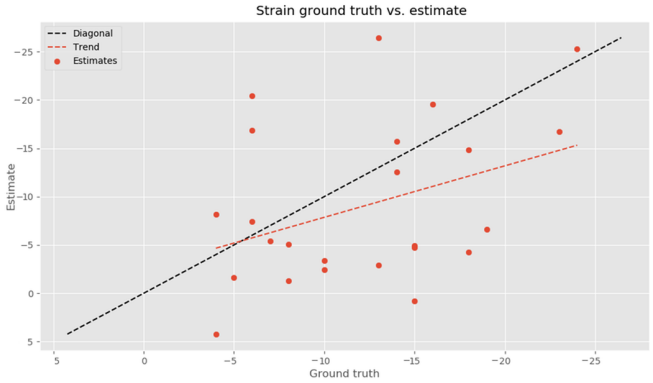

Fig. 4. Strain ground truth vs. deep learning estimates for 19 heart cycles in the test set.

IV. Discussion

During the training of the CNNs, no evidence of overfitting was observed. The validation loss follows a moving average of the training loss closely on a decreasing trend, indicating successful learning. The inference times estimated are quite high (approximately 230 ms between frame pairs), significantly higher then the ones reported by Østvik et al. [6]. However, a direct comparison of run-times is unfair since we run interference in Python on the CPU, while Østvik et al. used a laptop with a powerful GPU. Inference times can be significantly reduced if the experiments are ran using a GPU, due to their excellent performance on convolutions. If our method would be implemented in a compiled language, such as C++, further performance gains will be achieved. Another way of reducing inference time would be to crop the frames before inference. As the landmarks are located in the top half of a frame, it is wasteful to estimate the frame-to-frame displacements for the entire image.

Overall on the testing data, the CNN model tends to underestimate the motion of the landmarks. As the bending penalty, in its effort to ensure spatial smoothness, also penalizes the magnitude of the displacements, it is also possible that a decrease in the regularization parameter could alleviate the underestimation. Alternatively, an interesting experiment could be to daisy-chain several networks with few downsampling layers. This would estimate the larger motions incrementally while keeping track of the smaller motions as well.

Out-of-plane movement and constant noise in the samples were issues that are difficult to overcome. Especially, the test set was quite noisy, as air bubbles and the insertion of artificial valves deteriorated the quality of the ultrasound recordings. Efforts were made to keep the tracking from failing in these situations, including smoothing the motion vectors using exponential moving averages and Kalman filtering without success. Using such smoothing methods also induce latency in the tracking, which may lead to failed tracking in later frames and introduce bias in the strain estimates.

Drifting in uniform regions was another common issue. This can be attributed to the way the motion estimators are trained. As the normalized cross-correlation was optimized, the trained models do not differentiate which pixels go where in the warped frames, as long as they have similar brightness. That means that the pixels within a uniform region may be shuffled, causing folding in the warped frame. To reduce folding more networks could be daisy-chained, or the regularization parameter could be increased. Both of these approaches come at a cost, as more networks would increase the inference time, and increased regularization would be more biased towards small displacements.

V. Conclusion

The motion estimation task was performed by a fully convolutional neural network, which was trained in an unsupervised manner and achieved results that are comparable to the ones measured by the human expert, even though a consistent under-estimation was observed. Since the network contains only convolutional layers, it can predict motion on samples of any resolution. This is particularly useful in ultrasound, where the view and sample rate impacts the spatial resolution, so that samples acquired using the same equipment on the same patient may have different dimensions. A further advantage of the proposed method is that it does not require manual annotation of the training set, which is a time-consuming process.

Acknowledgment

The authors would like to acknowledge Espen Holte, Bjørnar Grenne and Håvard Dalen who acquired the clinical data, as well as the support from: ”Operating Room of the Future at St. Olavs hospital” and ”CIUS - Centre for Innovative Ultrasound Solutions, NTNU”.

References

[1] S. Zaunseder, M. Riedl, J. Kurths, H. Malberg, R. Bauernschmitt, and N. Wessel, “Impact of cardiac surgery on the autonomic cardiovascular function,” *Journal of Computational Surgery*, vol. 1, no. 1, p. 9, 2014.

[2] R. M. Lang, L. P. Badano, V. Mor-Avi, J. Afilalo, A. Armstrong, L. Ernande, F. A. Flachskampf, E. Foster, S. A. Goldstein, T. Kuznetsova, P. Lancellotti, D. Muraru, M. H. Picard, E. R. Rietzschel, L. Rudski, K. T. Spencer, W. Tsang, and J.-U. Voigt, “Recommendations for cardiac chamber quantification by echocardiography in adults: an update from the american society of echocardiography and the european association of cardiovascular imaging.” *European heart journal cardiovascular Imaging*, vol. 16, pp. 233–270, Mar. 2015.

[3] C. L. Reichert, C. A. Visser, R. B. van den Brink, J. J. Koolen, H. B. van Wezel, A. C. Moulijn, and A. J. Dunning, “Prognostic value of biventricular function in hypotensive patients after cardiac surgery as assessed by transesophageal echocardiography.” *Journal of cardiothoracic and vascular anesthesia*, vol. 6, pp. 429–432, Aug. 1992.

[4] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, “A deep learning framework for unsupervised affine and deformable image registration,” *Medical image analysis*, vol. 52, pp. 128–143, 2019.

[5] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, “Voxelmorph, a learning framework for deformable medical image registration,” *IEEE transactions on medical imaging*, 2019.

[6] A. Østvik, E. Smistad, T. Espeland, E. A. R. Berg, and L. Lovstakken, “Automatic myocardial strain imaging in echocardiography using deep learning,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 309–316.