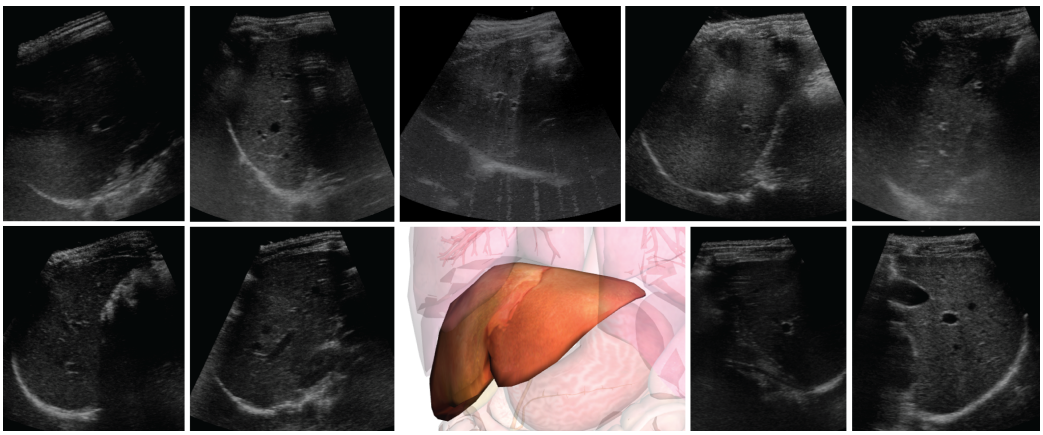
Proceedings of the MICCAI 2015 workshop

Challenge on Liver Ultrasound Tracking

CLUST 2015

Held in conjunction with MICCAI 2015
Munich, Germany, October 9, 2015



Preface

Non-invasiveness and high temporal resolution makes ultrasound (US) imaging an attractive choice for tracking and tissue motion analysis. Clinical applications include cardiac, respiratory and muscular motion estimation. This challenge is concerned with tracking the motion of the liver during free-breathing, with the ultimate aim of enabling motion compensation in image-guided interventions and therapies.

A large number of relevant tracking strategies for US liver images have been proposed, but it is hard to compare their performances. The Challenge on Liver Ultrasound Tracking (CLUST) 2014, held in conjunction with MICCAI 2014, allowed for the first time the quantitative, direct comparison of different tracking methods using the same tracking objective and validation strategy. CLUST14 successfully established the first reference tracking performance and is still available as an open challenge [1].

This challenge expands and renews CLUST14. The dataset was increased from 38 to 64 2D sequences and from 16 to 22 3D sequences. Annotations were improved by manually labeling landmarks on 10% of all frames by three observers (instead of one) and taking the mean position. Additionally, the quality of the annotations was ensured by visual inspection and corrections if required. We had planned to also include annotations of the diaphragm or liver boundary in the challenge. Yet we experienced high observer disagreements and hence excluded these annotations to not compromise the ground truth quality. Furthermore, in contrast to CLUST14, an on-site challenge will be held during the conference satellite event to confirm the performance of the methods on unseen images.

In details, we collected a dataset of 86 sequences of volunteers under free breathing and released 64 sequences, provided by six groups (see pages 51-53). The length of the sequences ranges from 4 seconds to 10 minutes and acquisitions were done with different US scanners and settings. The dataset is divided into two parts, according to the image dimension. The first part is composed of 64 2D sequences, and the second part consists of 22 3D sequences, both with point-landmark annotations and from healthy volunteers. The data were anonymized and distributed in the format of sequences of 2D images (.png) or 3D images (.mha). The data were split into a training and a test set. Training data (40% of all sequences) and part of the test data

[1] CLUST14 `http://clust.ethz.ch/clust2014.html`

(40%) were available prior to the challenge (see pages 52-53). Annotations were provided for the training set, to allow for some tuning of the tracking algorithm. For the test set, the annotations of the first images were provided. These needed to be tracked over time. The remaining 20% of the test data, is distributed at the MICCAI CLUST event for on-site application of the methods. This helped the organizers and participants to comment on algorithm run-time, parametrization and tuning, flexibility, and feasibility for a real application scenario. The details and results for the on-site dataset were not included in this proceedings book, as they were available after the paper submission deadline.

In response to the call for papers, we had 50 requests for access to the data. A total of 6 papers were accepted to the workshop. These papers underwent a peer-review process, with each paper being reviewed by 2 members of the Organizing Committee. The revised papers, incorporating the reviewers' comments, are included in this proceedings book. During the workshop, challenge participants were able to apply their method on new images and present their research. Attendees benefited from discussions, learning about different tracking techniques and gaining a perspective of the challenges and potentials of US tracking.

We would like to express our sincere appreciation to the authors whose contributions to this proceedings book have required considerable commitment of time and effort. We also thank (in alphabetical order of surnames) Jyotirmoy Banerjee from the Biomedical Imaging Group, Erasmus MC, Rotterdam, The Netherlands; Herve Liebgott from CREATIS INSA Lyon, France; Frank Lindseth and Sinara Vijayan from SINTEF Medical Technology, Trondheim, Norway; Julia Schwaab from mediri GmbH, Heidelberg, Germany; and their colleagues for providing data and annotations. Without their help this workshop would not have been possible.

August 2015

Valeria De Luca
Emma Harris
Muyinatu A. Lediju Bell
Christine Tanner

Workshop Organization

Organizing Committee

Valeria De Luca	ETH Zurich, Switzerland
Emma Harris	Institute of Cancer Research, London, UK
Muyinatu A. Lediju Bell	Johns Hopkins University, Baltimore, USA
Christine Tanner	ETH Zurich, Switzerland

Website

`http://clust.ethz.ch/`

Contents

Liver Ultrasound Tracking using a Learned Distance Metric

Daniel Nouri and Alex Rothberg

4Catalyzer, Inc.,
Guilford CT 06437, USA,
{dnouri,arothberg}@4catalyzer.com

Abstract. We present a method for landmark tracking in long liver ultrasound sequences. We employ metric learning, and train a convolutional neural network to map from pixel intensities of grayscale ultrasound image patches into a low-dimensional embedding space such that patches showing the same landmark at their center have a small L2 distance in the embedding. We then locate landmarks throughout a sequence of ultrasound frames by extracting patches from a search window inside the target frame and finding the patch in the target frame that in the embedding space minimizes the distance to a number of template patches containing the landmark and extracted from previous frames. Our approach had a mean tracking error of 2.83mm, with 38 of 62 tracked points having an error of less than 1.5mm.

Keywords: ultrasound, tracking, medical imaging, learned distance metric, CLUST15

1 Introduction

Ultrasound (US) imaging is a widely used medical imaging technique due to its relatively low-cost components, fast acquisition speed, and safe, non-ionizing radiation. In addition, because it also offers high temporal resolution images in real-time, US is often used for tissue tracking during image-guided intervention and therapy.

Tracking the motion of tissue in an ultrasound sequence is complicated by respiratory motion, image noise, and the relatively long (often more one minute) acquisitions. Tracking is further complicated by large changes in shape of the tracking target, particularly when anatomical targets are not captured in plane. Long acquisitions are particularly difficult due to high likelihood of both patient and operator motion. In many cases, the US capture probe is handheld.

In this paper, we present a new tracking scheme based on a distance metric for US image patches that is learned from data. We use the learned distance metric to compare candidate square image patches with patches extracted from a target reference frame. The algorithm requires a training phase in which the distance metric is learned from raw pixel intensity values, for all device types

simultaneously. No further parameterization is needed when applying the algorithm to previously unseen data. We evaluated this new scheme participating in the MICCAI CLUST15 [2] liver ultrasound tracking challenge.

1.1 Related Work

Several systems that use deep neural networks to learn distance metrics have been proposed for applications such as face verification and signature verification. Training for verification instead of classification has the advantage that fewer labeled examples are needed, and that systems can naturally generalize to categories previously unseen [7,8,20].

The very similar CLUST14 challenge saw a wide range of proposed methods ranging from non-linear image registration, and long-term and short-term template matching, to Bayesian methods. None of the proposed methods incorporated deep neural networks as part of their solution [17]. However, the winner of the *MICCAI 2013 Grand Challenge on Mitosis Detection* was a system using a deep neural network at its core [10].

2 Materials and Methods

2.1 Ultrasound Data

2D B-mode ultrasound data was provided as part of the MICCAI 2015 Challenge on Liver Ultrasound Tracking (CLUST) [2]. The data were cine images of human livers. The data came from a number of patients and institutions (CIL, ETH, ICR, and MED datasets) and were acquired by one of five ultrasound systems (Ultrasonix MDP, Siemens Antares, Elekta Clarity - Ultrasonix, DiPhAs Fraunhofer and Zonare z.one). The data had varying spatial (0.28 – 0.55mm) and temporal resolution (11 – 23Hz) and sequences lasted from 59.4 – 328.6s. The number of annotations per image sequence ranged from one to five liver features.

24 of the 48 datasets, totaling 53 target annotations, were provided with ground truth annotations (of liver blood vessels) throughout the acquisition sequence. Approximately 10% of the frames had the locations of the tracking points annotated. A total of 62 points had to be tracked in the test-set where only the initial position of liver features (blood vessels centers) was given.

Annotations were provided in the following form: frame number, x-pixel (lateral position) and y-pixel (axial position).

2.2 Distance Metric Learning

Given a sequence of ultrasound images $I_{0...N}$, along with an annotated landmark L given by its position $c_0 \in \mathbb{R}^2$ in the first frame I_0, the problem is to locate the center positions $c_{1...N}$ of the given landmark in all subsequent frames $I_{1...n}$. We solve the problem by training a convolutional neural network (ConvNet)

that learns a function $G_W(p)$ to map ultrasound image patches p into to a low-dimensional space such that the distance metric

$$D_W(p_i, p_j) = ||G_W(p_i), G_W(p_j)||_2 \tag{1}$$

is small if p_i and p_j show the same landmark at their center and large otherwise.

The weights W of mapping function G_W are learned using stochastic gradient descent and the following loss function originally proposed in [14],

$$\mathcal{L}(p_i, p_j, s_{ij}, W) = \begin{cases} \frac{1}{2}\, D_W(p_i, p_j)^2 & \text{if } s_{ij} = 1, \\ \frac{1}{2}\, max(0, m - D_W(p_i, p_j))^2 & \text{if } s_{ij} = 0 \end{cases} \tag{2}$$

where $s_{ij} = 1$ for a pair of patches (p_i, p_j) that show the same landmark at their center and $s_{ij} = 0$ otherwise. m is a margin constant used to limit the penalty for dissimilar pairs; it was set to 0.1.

2.3 Training Data

For training, we form pairs of square patches (p_1, p_2) of the same landmark extracted from different frames using the ground-truth annotations. These are the pairs for which the distance $D_W(p_1, p_2)$ is learned to be small.

In addition, we form twice as many pairs of patches for training where p_1 contains the same landmark as p_2 but shifted away from the center by at least 4 and by at most 46 pixels in both dimensions uniformly at random. We also train with some pairs where both patches show different landmarks taken from the same sequence. These are the pairs for which the distance is learned to be big.

Because our mapping function G_W has many learnable parameters (1,865,278 in our best configuration), and thus tends to show high variance, we augment our training data by randomly flipping patches in both vertical and horizontal directions.

We pre-process all ultrasound image frames with a small-size median filter. All extracted patches are of size 46 x 46, which we determined empirically to be optimal. We use all available training data to learn the parameters of G_W.

2.4 Convolutional Neural Network Architecture

A ConvNet is a feed-forward neural network that uses successive pairs of convolutional and max-pooling layers, followed by fully connected layers. The input to our ConvNet is raw pixel intensities, the output is an embedding in low-dimensional space. All weights W of the network are learned from scratch using the contrastive loss function in (2). The weights are randomly initialized using Glorot initialization as described in [12]. Weights are updated during training using Nesterov's Accelerated Gradient [18]. We train a single network that learns the weights of G_W for all sequences and device types simultaneously. Table 1 lists the architecture of our ConvNet.

Our decision to use a ConvNet to implement G_W is motivated by the recent successes of using ConvNets in mitosis detection, and in computer vision tasks in general [10]. Through the use of learning curves in our experiments we've determined that our ConvNet is still well in the regime where using more training data (and possibly more aggressive data augmentation) leads to a linear increase in performance. Another intriguing property of ConvNets is that they can learn from raw pixels directly, and thus eliminate the often tedious task of engineering features and choosing dataset-specific parameters by hand. The max pooling layer calculates the max value of a particular feature over a region of the image. This ensures that the same result is obtained even when images features undergo small translations.

Table 1. 8-layer architecture of our ConvNet with a total of 1,865,278 learnable parameters. Layer type: I - input, C - convolutional, MP - max-pooling, MO - maxout [13], FC - fully-connected.

Layer	Type	Maps and Neurons	Filter size
0	I	1Mx46x46	—
1	C	32Mx46x46	5x5
2	MP	32Mx25x25	2x2
3	C	64Mx23x23	3x3
4	MP	64Mx12x12	2x2
5	FC	200	1x1
6	MO	50	4x1
7	FC	50	1x1

2.5 Template Patches

In a given ultrasound frame I_i, we predict the center $c_i \in \mathbb{R}^2$ position of the tracked landmark L by finding a target patch p that minimizes

$$D_W(p, t_0) + \frac{\sum_{k=i-K}^{i-1} D_W(p, t_k)}{K} \tag{3}$$

for $K + 1$ template patches t extracted from previous frames and showing landmark L. See Figure 1 for examples of the distance map created by the window search.

Template patch t_0 is extracted from the initial frame I_0 with its center position c_0 provided by the human annotation. Patches $t_{i-K} \dots t_{i-1}$ are extracted each from K previous frames $I_{i-K} \dots I_{i-1}$ with their center at the position of the tracking algorithm's previously predicted landmark position $y_i \in \mathbb{R}^2$. Thus, to be able to extract template patches for use in frame I_i, we must first predict the position y of L in frames $I_{i-K}, \dots, I_{i-1}$.

Through our experiments, we determined $K = 10$ to be the optimal number of template patches to use from previous predictions.

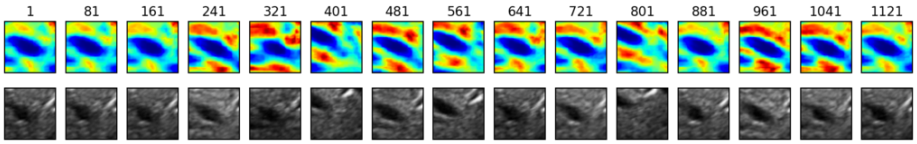


Fig. 1. Distance map created by the window search; the dark blue region represents the most similar match. Performed on the second point in `ICR-01` on frames $1, 81 \ldots 1121$.

2.6 Search Window

When looking for a patch p that has minimum distance to template patches t for landmark L as defined in Section 2.5, we only consider patches in the target frame I_i that have their center pixel within a defined square search window. This search window is searched through exhaustive search and itself centered at the predicted position y_{i-1} of landmark L in the previous frame I_{i-1}, or at the initial annotation c_0 for $i = 1$.

The predicted position y_i is defined by the center pixel of the patch p that's found to have minimum distance. We chose the width of the search window to be 24 pixels, which allows tracking to compensate for errors in previous frames.

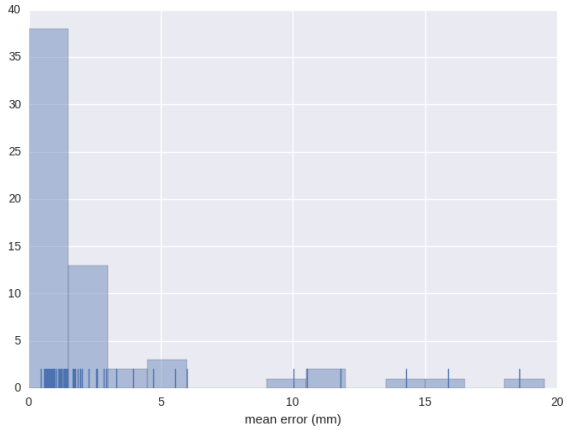


Fig. 2. Histogram of mean tracking errors. 38 of 62 landmarks in the test set have a mean tracking error of $1.5mm$ or less.

3 Experimental Results

We implemented the proposed approaches and methods in the previous section using Nolearn [3] and Lasagne [1] in Python. Lasagne uses Theano [6] for execu-

tion, which allows us to use GPUs for computations. We ran all execution using the Amazon Web Services (AWS) `g2.2xlarge` instances[1].

Each version of the network architecture was trained using a single `g2.2xlarge` instance though various network architectures and hyperparameters settings were often trained in parallel using multiple machines[2]. The sliding window search was performed using a single `g2.2xlarge` instance though each sequence could be made to run in parallel.

The tracking results for each sequence group are shown in Table 2. The results in `CIL`, `MED-1` and `MED-2` were relatively consistent with small standard deviations, 95^{th} percentiles and maximum values. For the `ETH` and `ICR` examples, there were examples where the search "got lost" and the algorithm returned a window very off from the desired target. For example the mean error on `ICR-07_2` was $18.55mm$. Figure 2 visualizes a number of outliers in the mean test error distribution.

The computational time to learn the distance metric was approximately 1.5 hours for the best performing models. Training differs slightly depending on the exact network architectures used and the number of training epochs needed for sufficient convergence. Once the distance metric was learned, the same metric was applied to each sequence group. The processing time for the search was 100msec / annotation / frame. This time was per learned distance metric. Given that we were performing an ensembling where two motion vectors were averaged in order to produce the final result, the actual time was double that: 200msec / annotation / frame though the two estimations can be performed entirely in parallel. Real-time performance of our system is well within reach considering that the GPUs we used in our experiments are about four times slower than the most modern GPUs available.

Table 2. Tracking errors for the 2D point-tracking test data.

	Mean [mm]	Std [mm]	95% [mm]	Min [mm]	Max [mm]
Sequence set					
CIL	1.65	0.97	3.49	0.01	5.13
ETH	2.61	4.33	13.35	0.01	27.70
ICR	5.80	8.86	29.01	0.03	39.39
MED1	2.13	2.25	7.10	0.01	16.83
MED2	1.53	1.03	3.83	0.02	6.41
All sequences	2.83	4.86	13.13	0.01	39.39

[1] `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using_cluster_computing.html`

[2] The cluster of machines was managed using StarCluster.

4 Conclusions

In this paper, we proposed a method of tracking target tissues in long (over one minute) 2D ultrasound sequences of liver. The proposed method uses a ConvNet to learn a distance metric which can then be used in a sliding window fashion to determine the motion vector of the tissue from the previous to the current frame. The experimental results were obtained using 24 sequences of ultrasound with 62 annotated landmarks. The results showed the proposed method has good average accuracy, though there were circumstances where the technique "got lost" and produced results far from the target.

The current implementation is not computationally optimal. The embedding of each window is computed independently, resulting in many redundant convolution operations. Using "fully convolutional networks" instead, we should be able to reach real-time performance easily [16].

Because our ConvNet-based embedding function has many degrees of freedom, it exhibits high variance. In future work, we aim to reduce variance by averaging the outputs of multiple networks trained on the same data but with different random initialization. We're also confident that running the embedding function on patches flipped vertically and horizontally and averaging results would lead to better generalization. These two techniques would both come at the expense of slower runtime performance.

When calculating a distance map inside a given search window, we observe that the map tends to be quite noisy. In future work, we want to look at smoothing functions to be able to more robustly find the correct center pixel.

References

1. Lasagne, `https://github.com/Lasagne/Lasagne`
2. Miccai challenge on liver ultrasound tracking, `http://clust.ethz.ch/`
3. nolearn, `https://github.com/dnouri/nolearn`
4. Banerjee, J., Klink, C., Peters, E., Niessen, W., Moelker, A., van Walsum, T.: 4d liver ultrasound registration. In: Ourselin, S., Modat, M. (eds.) Biomedical Image Registration, Lecture Notes in Computer Science, vol. 8545, pp. 194–202. Springer International Publishing (2014), `http://dx.doi.org/10.1007/978-3-319-08554-8_20`
5. Bell, M.A.L., Byram, B.C., Harris, E.J., Evans, P.M., Bamber, J.C.: In vivo liver tracking with a high volume rate 4d ultrasound scanner and a 2d matrix array probe. Physics in Medicine and Biology 57(5), 1359 (2012), `http://stacks.iop.org/0031-9155/57/i=5/a=1359`
6. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: A cpu and gpu math compiler in python. In: van der Walt, S., Millman, J. (eds.) Proceedings of the 9th Python in Science Conference. pp. 3 – 10 (2010)
7. Bromley, J., Guyon, I., Lecun, Y., Sckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. In: In NIPS Proc (1994)
8. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Computer Vision and Pattern Recognition,

2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 539–546 vol. 1 (June 2005)
9. Cifor, A., Risser, L., Chung, D., Anderson, E., Schnabel, J.: Hybrid feature-based diffeomorphic registration for tumor tracking in 2-d liver ultrasound images. Medical Imaging, IEEE Transactions on 32(9), 1647–1656 (Sept 2013)
10. Cireşan, D., Giusti, A., Gambardella, L., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) Medical Image Computing and Computer-Assisted Intervention MICCAI 2013, Lecture Notes in Computer Science, vol. 8150, pp. 411–418. Springer Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-40763-5_51
11. De Luca, V., Tschannen, M., Szkely, G., Tanner, C.: A learning-based approach for fast and robust vessel tracking in long ultrasound sequences. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) Medical Image Computing and Computer-Assisted Intervention MICCAI 2013, Lecture Notes in Computer Science, vol. 8149, pp. 518–525. Springer Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-40811-3_65
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International conference on artificial intelligence and statistics. pp. 249–256 (2010)
13. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. arXiv preprint arXiv:1302.4389 (2013)
14. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Computer vision and pattern recognition, 2006 IEEE computer society conference on. vol. 2, pp. 1735–1742. IEEE (2006)
15. Lediju, M., Byram, B., Harris, E., Evans, P., Bamber, J.: 3d liver tracking using a matrix array: Implications for ultrasonic guidance of imrt. In: Ultrasonics Symposium (IUS), 2010 IEEE. pp. 1628–1631 (Oct 2010)
16. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. ArXiv e-prints (Nov 2014)
17. Luca, V.D., Benz, T., Kondo, S., König, L., Lübke, D., Rothlübbers, S., Somphone, O., Allaire, S., Bell, M.A.L., Chung, D.Y.F., Cifor, A., Grozea, C., Günther, M., Jenne, J., Kipshagen, T., Kowarschik, M., Navab, N., Rühaak, J., Schwaab, J., Tanner, C.: The 2014 liver ultrasound tracking benchmark. Physics in Medicine and Biology 60(14), 5571 (2015), http://stacks.iop.org/0031-9155/60/i=14/a=5571
18. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k2)$. Soviet Mathematics Doklady 27(2), 372–376 (1983)
19. Preiswerk, F., De Luca, V., Arnold, P., Celicanin, Z., Petrusca, L., Tanner, C., Bieri, O., Salomir, R., Cattin, P.C.: Model-guided respiratory organ motion prediction of the liver from 2D ultrasound. Medical Image Analysis 18(5), 740–751 (Jul 2015), http://dx.doi.org/10.1016/j.media.2014.03.006
20. Sun, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. CoRR abs/1406.4773 (2014), http://arxiv.org/abs/1406.4773
21. Vijayan, S., Klein, S., Hofstad, E., Lindseth, F., Ystgaard, B., Lango, T.: Validation of a non-rigid registration method for motion compensation in 4d ultrasound of the liver. In: Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on. pp. 792–795 (April 2013)

Liver Ultrasound Tracking Using Kernelized Correlation Filter With Adaptive Window Size Selection

Satoshi Kondo

Konica Minolta Inc., Osaka, Japan
satoshi.kondo@konicaminolta.com

Abstract. We propose a method to track tissues in long ultrasound sequences of liver. The proposed method is based on kernelized correlation filter (KCF) and we introduce two extensions to KCF; adaptive window size selection and motion vector refinement with template matching. We compare KCF and the proposed method by using some training sequences of 2D ultrasound and the mean tracking error can be improved with the proposed method by up to nearly 3 pixels. The tracking performance is also assessed on 19 test sequences of 2D ultrasound with 62 regions of interests. Mean tracking error is 1.09 mm.

Keywords: Ultrasound, Liver, Tracking, Kernelized correlation filter, Template matching

1 Introduction

It is important to track a region of interest (ROI) to compensate motion to ensure accuracy of robot-assisted diagnosis [1], focused ultrasound surgery [2] and dose delivery in radiation therapies [3]. Ultrasound is one of potential imaging modalities for image guidance and has some advantages such as real-time imaging, noninvasive and cheap comparing to other imaging modalities such as CT and MRI.

Various methods have been proposed for tracking a moving object in a video sequence. In recent years, tracking methods using discriminative approach have been proposed and reported to exhibit high performance [4–6]. Especially, Kernelized Correlation Filter (KCF) is known to show high performance despite its high speed processing [7, 8].

KCF shows high tracking performance, but it has some problems such as, (1) the user has to specify a region enclosing the target object to track, (2) KCF emphasizes robustness than accuracy. For example, the criteria of true positive is that the tracked position is within 20 pixels relative to the ground truth in [7].

On the other hand, in medical applications, both robustness and accuracy are required for tissue tracking. In this paper, we propose a tracking method of tissues in long ultrasound sequences of liver. The proposed method is based on

KCF and we introduce two extensions to KCF. The first one is adaptive window size selection and the second one is motion vector refinement.

2 Overview of Kernelized Correlation Filter

In this section, we briefly explain KCF, which is a basis of our proposed method. In KCF, when a target object to track is specified, a discriminative function is calculated by kernel ridge regression using the image inside the region of the target to track as a positive sample and the images in the surrounding region of the target as negative samples. Since the positive sample and the negative samples are expressed with a circulant matrix in KCF, the regression coefficient vector in the kernel space is obtained by using Discrete Fourier Transform (DFT),

$$\hat{\boldsymbol{\alpha}}^* = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda}, \tag{1}$$

where $\mathbf{y}$ is a regression target vector, 1 for an element corresponding to a positive sample and 0 for an element corresponding to a negative sample, $\mathbf{x}$ is an image patch in a tracked region, λ is a regularization weight in ridge regression, a hat $\hat{}$ and a star * denote the DFT of a vector and complex-conjugate, respectively. In the case of two dimensional data and the dimension of $\mathbf{x}$ is $M \times N$, the dimension of $\mathbf{y}$ is also $M \times N$. Note that the tracking window, which is the size of $\mathbf{x}$, has 2.5 times the size of the target to track in the implementation of KCF.

In case of Gaussian kernel, $\hat{\mathbf{k}}^{\mathbf{xx}'}$ is

$$\hat{\mathbf{k}}^{\mathbf{xx}'} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}(\hat{\mathbf{x}} \odot \hat{\mathbf{x}}'^*)\right)\right), \tag{2}$$

where $\mathcal{F}^{-1}$ and $\odot$ denote inverse Fourier transform and element-wise multiplication, respectively.

In the detection phase, a regression function in Eq. (3) is calculated and the position where the regression value is maximum is the tracked position.

$$\hat{\mathbf{f}}(\mathbf{z}) = \left(\hat{\mathbf{k}}^{\mathbf{xz}}\right)^* \odot \hat{\boldsymbol{\alpha}}, \tag{3}$$

where $\mathbf{z}$ is a image patch in a frame to track which has the same size with $\mathbf{x}$.

Then, $\hat{\boldsymbol{\alpha}}$ is re-calculated at the new tracked position in the next frame using Eq. (1). In the implementation, however, $\hat{\boldsymbol{\alpha}}$ is gradually updated as in Eq. (4).

$$\hat{\boldsymbol{\alpha}}'_{t+1} = \beta\hat{\boldsymbol{\alpha}}_{t+1} + (1-\beta)\hat{\boldsymbol{\alpha}}_t, \tag{4}$$

where β is a weight for the interpolation.

3 Proposed Method

3.1 Overview

It is desirable that a target area in the object tracking is set to enclose the target object in the first frame by a user. However, in some cases, the user may specify

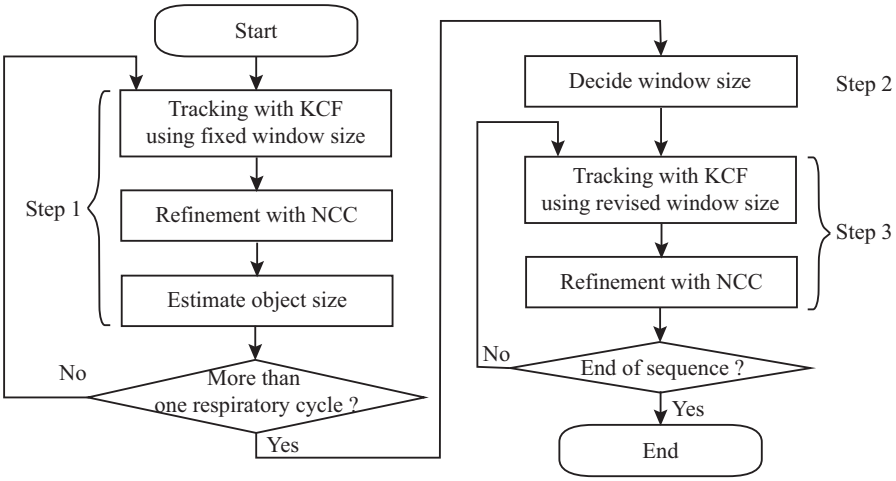


Fig. 1. A flowchar of the proposed method.

only the center position of the target object. In such cases, the tracking system has to decide the region of the target object to successfully track the object. Our proposed method is based on KCF and the size of the tracking window should be set in consideration of the following two aspects. The first one is the size of the tracking target object and the second one is the amount of motion of the target object.

In KCF, a discriminative function is determined using the image inside the region of the target object as a positive sample and the image in the surrounding region of the target object as negative samples. Therefore, it is desirable that the image of the target object and the image in the surrounding region have different texture. KCF calculates the correlation in the Fourier domain in the tracking process as described in Section 2. That means the amount of the motion should be within the area of the Fourier transform which is the same as the size of the tracked window. Also, the ultrasound images of the liver have the characteristic that the motion is approximately periodic which is induced by respiration.

In the proposed method, the size of the target object and the maximum amount of motion are obtained by using initial frames during about one breathing cycle (Step 1). This is a kind of calibration phase. The size of the tracking window is decided by using the size of the target object and the maximum value of the motion vectors (Step 2). In the subsequent frames, tracking is performed with the tracking window (Step 3).

Fig. 1 is a flowchart of the proposed method. We will describe the details of each process in the following sections.

3.2 Tracking using fixed window size for initial frames (Step 1)

In Step 1, tracking is performed by KCF with a predetermined size of the tracking window at first for each frame. The predetemined window has a rectangular area

centered at the point designated by the user to the target object. Then, in order to improve the tracking precision, the template matching is performed for a narrow search region around the position detected by KCF. We use normalized cross correlation to evaluate the template matching.

Note that the learned discriminative function is gradually updated in KCF as mentioned in Section 2. Also, we do not update the template and use the same template for each frame in the template matching. The template is surrounding area of the target object in the first frame. The reason is to use the template obtained in the first frame for each frame is to avoid drift. Template matching is performed only in the region of around $\pm$ 2 pixels of the positions obtained by KCF. When the maximum normalized correlation value is greater than a threshold C_{th}, the result of the template matching is adopted. Otherwise, the tracking result of KCF is adopted without refinement. The area of performing the template matching is decided empirically.

Also, we estimate the size of the target object in every predetermined frame. Since the target object is a tissue such as blood vessels or tumors, the size of the target object is determined by the ellipse fitting in the proposed method.

We repeat the above tracking process for each frame and analyze the temporal history of the tracked positions. When the target object reaches the right (or left) end two times, we determine that one respiratory cycle has been passed. When it is determined that one respiratory cycle has been passed, the process proceeds to Step 2.

3.3 Refinement of region size (Step 2)

In Step 2, the size of the tracking window is determined using the amount of the motion obtained from the tracking results and the size of the target object. The size of the tracking window is a larger value of γ_1 times of the maximum value of the amount of motion between adjacent two frames and γ_2 times of the median value of the object size (major axis). The values are decided for width and height of the tracking window separately. The width and the height of the tracking window are multiples of 8, the minimum value of the width and the height is 16 pixels, and the maximum value is the initial tracking window size. When there are multiple tracking targets in a sequence, the sizes of the tracking windows are decided for each target.

3.4 Tracking using refined window size (Step 3)

In the subsequent frames, tracking with KCF and template matching is performed using the tracking window size determined in Step 2. When the target window size is changed in Step 2, $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{x}}$ in KCF and the size of the template used in the template matching should be changed. This is performed using the template used in Step 1, which is an image patch around the target object in the first frame. Specifically, for the template matching, the template is revised by extracting the center area of the initial template. The updated template is

Table 1. Tracking results for the 2D point-tracking training data. The numbers show the tracking errors in pixels.

SequenceName	KCF		Proposed	
	Mean	Maximum	Mean	Maximum
CIL-02 #1	3.00	6.50	2.52	5.81
ETH-05-2 #2	6.51	28.98	3.67	27.06
ICR-04 #2	2.38	8.50	1.77	6.50
MED-05-1 #1	5.93	14.64	5.30	12.71

converted to $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{x}}$ for KCF. Thus, $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{x}}$ is reset at the beginning of Step 3.

The subsequent process, tracking with KCF and the template matching, is the same as in Step 1.

4 Experimental Results

We evaluated the performance of the proposed method using the 2D point-tracking training data of liver ultrasound. The training data was provided by organizers of CLUST 2015, MICCAI Challenge on Liver Ultrasound Tracking.

Our implementation is based on the open source MATLAB code (version 2) at http://home.isr.uc.pt/∼henriques/circulant/. In the experiment, we used the following values for parameters. The feature is gray scale pixel value. Note that we compared the tracking performance with gray scale feature and Histogram Oriented Gradient (HOG) feature [9] as a preliminary experiment and the gray scale feature showed better performance for liver ultrasound sequences, while HOG feature shows much better performance than gray scale feature for surveillance and sport videos in [7]. The kernel type in KCF is Gaussian kernel with $\sigma = 0.2$. We selected a Gaussian kernel based on preliminary experimental results. λ in Eq. (4) is 10^{-4}. β in Eq. (4) is 0.0075, which is one tenth of the default parameter in case the feature vector is the gray scale feature in KCF. The initial tracking window and template size and the threshold C_{th} are 96×96 pixels and 0.8, respectively. The object size is estimated every 5 frames in Step 1. In Step 2, γ_1 and γ_2 in step 2 are 8 and 4, respectively. These values were decided empirically. We used the same parameters for all the sequences.

We compared the tracking performance of the proposed method with KCF in which the tracking window size is fixed at 96×96 pixels. Table 1 shows the results for some the 2D point-tracking training sequences. In Table 1, mean and maximum tracking errors are shown for KCF and the proposed method. Note that the ground truth is not given for all frames for the training sequences and the errors are calculated only for the frames the ground truth is given.

As can be seen in Table 1, the proposed method shows better tracking performance comparing to the original KCF. The proposed method can improve the mean errors by up to nearly 3 pixels and the maximum errors by nearly up to 2 pixels.

Table 2. Tracking results for the 2D point-tracking test data. The numbers show the tracking errors in millimeters.

SequenceName	Mean	Standard deviation	95th percentile	Minimum	Maximum
CIL	2.21	1.82	6.00	0.10	8.05
ETH	0.83	0.77	2.55	0.01	10.94
ICR	0.90	0.64	1.99	0.01	7.80
MED1	1.64	2.27	5.47	0.02	17.29
MED2	1.37	1.19	4.26	0.01	7.64
All	1.09	1.35	3.07	0.01	17.29

The tracking results for each sequence group (CIL, ETH, ICR, MED1 and MED2) in the test dataset are shown in Table 2.

As for computational time, we measured the processing time using a computer with an Intel Core i7 3.3 GHz CPU (6 cores) and 64 GB memory. We implemented the proposed method with MATLAB. The average processing time per target object and per frame in Step 1 and Step 3 for each sequence in the proposed method was from 106 to 155 msec and from 75 to 120 msec, respectively. For comparision, The average processing time per target object and per frame for each sequence in the original KCF was from 23 to 58 msec. The average additional time in the proposed method comparing to the original KCF was 59 msec per target object and per frame. We think the processing time can be improved if we implement the template matching in the proposed method by C++, and it's a future item.

5 Conclusion

In this paper, we proposed a tracking method of target tissues in long ultrasound sequences of liver. The proposed method is based on kernelized correlation filter (KCF) and we introduce two extensions to KCF for improving the tracking accuracy. The experimental results showed the proposed method had better accuracy comparing to the original KCF. Mean tracking error with the proposed method for test sequences of 2D ultrasound was 1.09 mm.

Items for future research are to improve the accuracy of tracking tissues near the boarder and expand the proposed method to 3D ultrasound.

References

1. Abolmaesumi, P., Salcudean, S. E., Zhu, W. H., Sirouspour, M. R., DiMaio, S. P.: Image-Guided Control of a Robot for Medical Ultrasound. IEEE Trans. Robotics and Automation, 18(1), 11–23 (2002)
2. Kopelmana, D., Inbarb, Y., Hanannelc, A., Freundlichc, D., Castelb, D., Pereld, A., Greenfeldd, A., Salamona, T., Sarelie, M., Valeanue, A., Papae, M.: Magnetic resonance-guided focused ultrasound surgery (MRgFUS): Ablation of liver tissue in a porcine model. European Journal of Radiology, 59(2), 157-162 (2006)

3. Bouchet, L. G., Meeks, S. L., Goodchild, G., Bova, F. J., Buatti, J. M., Friedman, W. A.: Calibration of three-dimensional ultrasound images for image-guided radiation therapy. Physics in Medicine and Biology, 46(2), 559 (2001)
4. Avidan, S.: Support vector tracking. IEEE Trans. Pattern Analysis and Machine Intelligence, 26(8), 1064–1072 (2004)
5. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 34(7), 1409–1422 (2012)
6. Hare, S., Saffari, A., Torr, P. H.: Struck: Structured output tracking with kernels. In IEEE International Conference on Computer Vision (ICCV), 263–270 (2011)
7. Henriques, J. F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Analysis and Machine Intelligence, 37(3), 583–596 (2015)
8. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Cehovin, L.: A novel performance evaluation methodology for single-target trackers. arXiv preprint arXiv:1503.01313 (2015)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 886–893 (2005)

Motion Tracking in 2D Ultrasound Using Vessel Models and Robust Optic-Flow

Maxim Makhinya and Orcun Goksel

ETH Zürich, Computer Vision Laboratory, Switzerland,
{makhinya,ogoksel}@vision.ee.ethz.ch

Abstract. Planned delivery of focused therapy is adversely affected by internal body motion, such as from breathing, which could be mitigated, if tracked accurately in real-time. By extending an algorithm for superficial vein tracking, we hereby present a robust real-time motion tracking method for 2D ultrasound image sequences of the liver. The method leverages elliptic and template-based models of vessels in the liver, coupled with a robust optic-flow framework. Potential drifts in this iterative tracking are corrected when the breathing phase is close to that of the initial reference frame, detected by comparing the appearance of tracked feature regions. Results are evaluated on the CLUST-2015 dataset, with 1.09 mm mean and 2.42 mm 95th percentile errors in 24 2D test sequences collected from four different centers.

1 Introduction

During radiation therapy and focused ultrasound treatment, patient motion adversely affects the planned irradiation of the target anatomy. Ultrasound tracking can provide a real-time solution to observe and mitigate such motion; thereby requiring smaller treatment margins, minimizing exposure to healthy tissue.

Tracking in ultrasound (US) is challenging due to low US signal-to-noise ratio and changes in landmark appearances in time. Vessels are robust landmarks, easier to identify and track in US, since they have high US contrast and well-defined shapes. We presented earlier an algorithm to identify and track superficial veins in the forearm, for the measurement of peripheral venous pressure [2,3]. In that work, large motions caused by hand-held manipulation of the probe, as well as veins collapsing meanwhile, were to be tracked, for which skin pressure measurements provided a surrogate to identify vein collapses and assist in their tracking. This method, when applied as given in [3] (with modifications to fit the liver images), fails entirely in 33% of the CLUST-2015 training sequences, while achieving a mean error of 1.32 mm for the rest.

Since the vessels do not compress in the liver case and motion is known to be repetitive, we have hereby extended the method of [3] by (*i*) reinitializing tracking with the reference frame when iterative tracking is poor; (*ii*) detecting and taking into account the shadowing from ribs and poor skin contact; (*iii*) allowing for features to go temporarily out of the US view or disappear in the shadow; (*iv*) removing reliance on additional pressure readings and the interactive user

input/correction in the venous-pressure case; and (v) adapting for curvilinear image acquisition. In particular, a more sophisticated motion tracking and a template-based resetting mechanism are introduced to recover from drift and erroneous tracking, while considering the repetitive motion. The Star-based edge detection [5,6] and template-based vessel tracking are employed similarly to [3]. The proposed algorithm was evaluated on a set of 2D image sequences provided by the CLUST-2015 challenge (http://clust.ethz.ch).

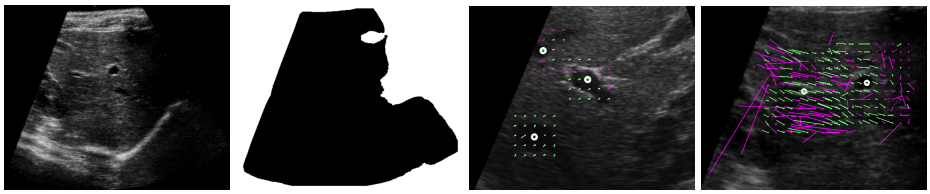
2 Methods

2.1 Motion Tracking

We use Lucas-Kanade method [7] for motion estimation between frames. The method takes a set of points $\{p\}$ in one frame and finds their corresponding positions $\{p'\}$ in the next frame. To limit motion estimation to the US field-of-view and to exclude shadowed areas, a motion mask is employed. The mask is built by binarizing the current frame f_i with a small threshold (5 in our setup) and median filtering the output with a $10 \times 10\,mm^2$ kernel to remove islands, c.f. Figs. 1(a) and 1(b). Median filter was implemented by a box-filter for speed considerations, exploiting the binary nature of the image.

We combine two tracking information: Iterative Tracking (IT) finds motion between consecutive frames f_{i-1} and f_i for individual points-of-interest (POI), whereas Reference Tracking (RT) finds motion from the (initial) reference frame f_0 to f_i for all POIs at once. RT is able to recover POI positions when motion cycle, induced by breathing, is in the same phase as the reference, while IT helps tracking points during the rest of the motion cycle. The points $\{p\}$ are selected on a square regular grid centered around each POI (with a grid separation set to $5\,mm$ herein). Starting from 3×3, the grid size is increased to $\{5 \times 5,\ 7 \times 7,\ \ldots\}$ to ensure a required number of tracking points (100 for RT and 10 for IT) fall inside the motion mask. IT uses current (previous-frame) positions of the POIs $\{mp\}$, while RT uses the reference POI positions $\{mp_0\}$.

Motion estimation in US is highly error-prone; hence, predicted point-wise motion vectors are filtered as follows. In IT, the per-point tracking error (PPE)



(a) (b) (c) (d)

Fig. 1: Initial US frame (a), corresponding motion mask (b). Iterative (c) and reference (d) motion tracking grids, where discarded motion vectors are colored in pink and POIs shown as white circles.

returned by the Lucas-Kanade method is used to discard error-prone pairs of points determined by a pixel error threshold of 6. Regardless of error, a minimum of 5 tracking points $\{p\}$ are kept. From remaining point pairs $\{p \rightarrow p'\}$, a 6 DOF affine transform is computed as the local motion estimation for each IT-tracked POI. For filtering RT, the points are checked for bi-directional consensus, i.e. the consistency of motion from first to current and back to first frame, $f_0 \rightarrow f_i \rightarrow f_0$. Consider this motion yields the following locations $\{p \rightarrow p' \rightarrow p''\}$, then the point is kept, iff $|p-p''|<4\,\text{mm}$. Next, the RT points are also checked for consensus with median motion direction, i.e. kept iff $(p'-p) \cdot \text{med}(\{p'-p\}) > 0$. If more than 60% of original RT points are filtered in the above process, then RT is considered invalid. In RT, a 6 DOF affine transform is computed from the combined set of RT point-pairs in the entire frame as a global motion estimation w.r.t. the reference frame. See samples of motion estimation grids in Figs. 1(c) and 1(d).

2.2 Vessel Size and Position Refinement

From the POI positions given in the refference frame, we first use a set of binary templates of different sizes to estimate the initial vessel size e_0 by template-matching, similarly to the initialization/detection step in [3]. If Normalized Cross Correlation (NCC) of the matched template score for a POI is smaller than a threshold (herein, 0.3 within a [0..1] NCC range), then this POI is considered to be a *non-vessel* structure; and, as such, its position is tracked only by RT and IT (mp_i), completely avoiding vessel-based treatments and later-described Template-based Reset. Otherwise, it is concluded to be a vessel and treated similarly to [3]: For relatively larger vessels, the Star edge-detection together with dynamic programming and ellipse fitting is used. For smaller vessels with difficult to detect edges, template-matching is used with binary templates of hypoechoic ellipses overlaid on hyperechoic backgrounds. We use an axis-aligned ellipse representation for vessels as $\mathbf{e} = [c_\text{x}, c_\text{y}, r_\text{x}, r_\text{y}]^T$, where c_x and c_y denote ellipse center coordinates and r_x and r_y are the semi-axes (radii) along corresponding axes. Although the vessels in the liver are not necessarily axis-aligned, this constraint remaining in the method from earlier venous application still allows for satisfactory tracking, meanwhile providing speed gain by reducing the number of templates.

For each frame f_i, the ellipse center $(c_\text{x}, c_\text{y})_i$ is transformed using the affine IT transform of the corresponding POI. Then, the center and radii are refined using $(a.)$ the Star method, when $(r_\text{y})_i > 10\,\text{px}$, or $(b.)$ binary template-matching, otherwise. The center refinement is restricted to $2 \times 2\,mm^2$ around the previous center $(c_\text{x}, c_\text{y})_i$, and the radii are permitted to change up to $2\,mm$ per frame. The vessel size is restricted in each axis to be within [75..120]% of its initial size in the reference frame to increase robustness to false detections.

2.3 Template-based Reset

The initial reference frame f_0 and the current frame f_i are used to re-initialize tracked POI positions, when the breathing/motion phase is the same as in the

reference frame. For this, first an auto-correlation noise level is estimated in the reference frame following initialization: An image patch of size $2(r_{\mathrm{x}}, r_{\mathrm{y}})_0 + (10mm, 10mm)$ centered at $(c_{\mathrm{x}}, c_{\mathrm{y}})_0$ is taken from f_0 and its NCC with shifted versions of itself (to eight surrounding positions with ± 0.5 mm offsets) are computed, with the minimum NCC score being our *reset-threshold* of self-similarity.

For each frame f_i, the above reference POI patch is template-matched within a region of $(c_{\mathrm{x}}, c_{\mathrm{y}})_0 \pm (10mm, 10mm)$, where the position of the best match is reported as a position reset candidate tp_i, iff its NCC score is larger than the reset-threshold determined for that POI as described above.

2.4 Motion Tracking Recovery

Fig. 2 presents an overview of per-frame tracking. For each frame f_i, RT and IT yield affine transforms Ar_i and $\{Ai_i\}$, respectively, which are used to track points without any vessel assumptions (e.g., for non-vessel structures). Additionally, $\{Ai_i\}$ are used to update positions of vessel representations $\{e_{i-1}\}$. Combined with the Template-Based Reset, a best ellipse estimate is then picked and its position is refined further. Alg. 1 gives algorithmic details of tracking recovery stages for improved robustness. Each stage is further explained below.

Updating Motion Tracked Points: Kalman filtering [1] is used for tracking POI locations, when RT is valid. Kalman-filter state is reset, if RT fails.

Picking Best Position: Tracking is switched from vessel tracking in Sec. 2.2 to pure motion-tracking by IT, if the POI moves outside the image or into a shadowed area; defined by a *visibility-mask* constructed from the earlier motion mask by including only shadowed areas, which extend all the way down to the far-side of the image. A vessel is considered not-visible, if more than half of the bounding box of a vessel representation e_i are outside this visibility-mask. To

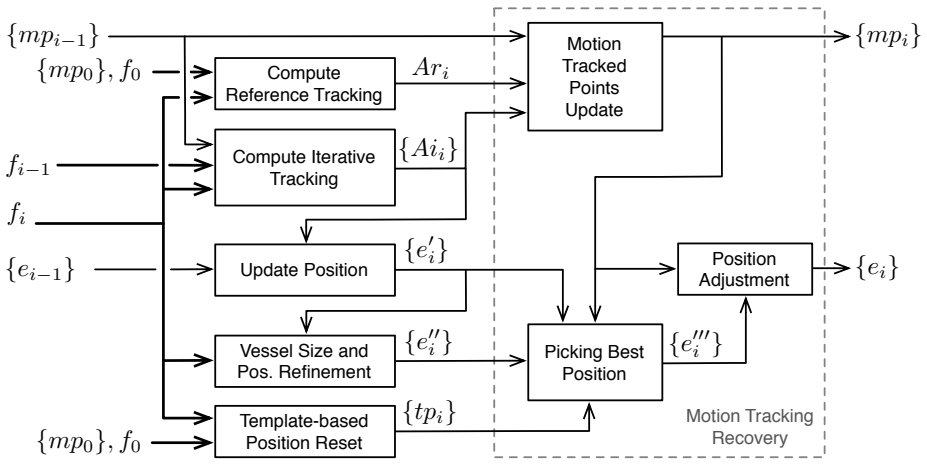


Fig. 2: Algorithm overview.

Algorithm 1 Motion Tracking Recovery

for each mp_{i-1} in $\{mp_{i-1}\}$ **do** ▷ Updating Motion Tracked Points
  if Is_Valid(Ag_i) **then**
    $mp_i^* \leftarrow Ag_i \cdot mp_{i-1}$
    $mp_i \leftarrow$ Update_Kalman_Filter(mp_i^*)
  else
    $mp_i \leftarrow Al_i \cdot mp_{i-1}$
    Reset_Kalman_Filter(mp_i)
for each $(e_i', e_i'', tp_i, mp_i)$ in $\{e_i', e_i'', tp_i, mp_i\}$ **do** ▷ Picking Best Position
  if Overlaps_Vessel_Mask(e_i') **or** Overlaps_Vessel_Mask(e_i'') **then**
    $e_i''' \leftarrow e_i'$
  else
    $cp \leftarrow$ Position_Of_Highest_NCC(Center(e_i''), mp_i,Size(e_i''))
    if Is_Valid(tp_i) **then**
      $cp \leftarrow$ Position_Of_Highest_NCC(cp, tp_i,Size(e_i''))
    $e_i''' \leftarrow (cp$,Size(e_i''))
for each (e_i''', mp_i) in $\{e_i''', mp_i\}$ **do** ▷ Position Adjustment
  $(s_x, s_y) \leftarrow$Size(e_i''')
  $s \leftarrow 20 + (s_x + s_y)/2$
  $d \leftarrow |$ Center(e_i''') $- mp_i|$
  $a \leftarrow \frac{1}{1+(d/s)^2}$
  $cp \leftarrow$Center(e_i''') $\cdot\, a + mp_i \cdot (1 - a)$
  $e_i \leftarrow (cp$,Size(e_i'''))

evaluate potential vessel locations and hence pick the best, a *vesselness* score is computed for each potential location $(c_\mathrm{x}, c_\mathrm{y})$ by the NCC of a similar-sized binary template and the image patch around that location.

Position Adjustment: Thanks to the robustness of the combined RT and IT strategies, tracked points $\{mp_i\}$ stay relatively close to actual targeted POI; nevertheless, not always track those with high precision. Conversely, the methods in Sec. 2.2 can locate vessel center relatively precisely, although such tracking may drift to adjacent hypoechoic structures in case of low local contrast or large motion. Accordingly, as seen in Alg. 1, a final position-adjustment for vessel-like POIs ensures that representations $\{e_i\}$ stay in close proximity of tracked points $\{mp_i\}$ – which is a constraint relaxed for larger vessels.

For vessel-like POIs the center positions $\{(c_\mathrm{x}, c_\mathrm{y})_i\}$ and, for others, the positions $\{mp_i\}$ are reported as the output tracked location.

3 Results and Discussion

The algorithm was implemented in C++ using OpenCV libraries. In particular, *calcOpticalFlowPyrLK* function was used for motion estimation (window-size set to $5\,mm$ and the number of pyramid levels to 5), *matchTemplate* for template matching in *CV_TM_CCOEFF_NORMED* mode, and *KalmanFilter* for RT position filtering (with *measurementNoiseCov* set to 200). Additionally, OpenMP

was used to accelerate motion estimation by running RT and all ITs in parallel, as well as for parallelization of template matching in Sec. 2.2.

The algorithm was evaluated on a Windows-based PC, equipped with an Intel Core i7-3770K CPU @ 3.5GHz. The performance depends on (*i*) the acquisition frame rate, since for larger frame-to-frame relative motion the motion estimation takes longer; (*ii*) tracked vessel sizes, affecting Star or template-matching performance; and (*iii*) the number of POIs to track. Table 1 presents per-sequence tracking speed for all challenge sequences. The reason for the processing speed to vary can be attributed to larger motion at lower acquisition frame rates as well as different number of POI in given seeuences. It is, nevertheless, seen that our processing is faster than the acquisition in all cases, with a latency of no more than the acquisition frame rate.

All algorithm's parameters were optimized using a training data set, provided by the CLUST-2015 organizers. Table 2 presents tracking performance results evaluated by CLUST organizers on their test data-set. Each sequence had up to 4 points marked in an initial reference frame, and the algorithm tracked them through the rest of the frames (average image resolution is $\sim$447$\times$552 px and average sequence length is $\sim$3761 frames). The best average individual score in the previous tracking challenge CLUST-2014 [4], on slightly smaller data-set and with different annotators, was 1.33 mm mean error (standard-deviation σ=1.94); and the error for median fusion of six participants was 1.08 mm (σ=1.42). Our method is seen to perform with 1.09 mm mean error (σ=1.75), superior to any earlier results, and at a comparable level with the earlier consensus (median-fused) tracking results.

It was observed in some sequences that frames were *dropped*; sometimes later appearing as an out of context frame (probably injected later due to a buffer overflow in the video capturing device, unless this is a data-preparation artifact). Such frames were detrimental to motion tracking. Therefore, if the average per-point error PPE in IT for a frame is over 3 times higher than the median average PPE of last 5 frames, we simply skipped that frame and returned previous POI positions.

Table 1: Per-sequence performance, where image acquisition rate is given in Hz, and algorithm's performance in Frames Per Second (FPS).

Sequence	**CIL**		**ETH**									
	03	04	06-1	06-2	07-1	07-2	08-1	08-2	09-1	09-2	10-1	10-2
Hz	18	15	16	16	17	17	17	17	15	15	17	17
FPS	28.5	38.5	50.5	43.9	30.8	30.2	31.4	31.4	20.2	25.3	20.5	18.1
Sequence	**ICR**				**MED**							
	05	06	07	08	06-1	06-2	07-1	07-2	07-3	07-4	08-1	08-2
Hz	20	21	23	23	20	20	20	20	20	20	11	11
FPS	48.8	50.7	44.5	36.7	22.6	24.5	29.3	28.2	25.7	21.0	16.2	17.5

Table 2: Mean tracking error, standard deviation, 95th percentile, minimum and maximum errors for each point of interest as well as average scores per sets of points and total scores for all points in the CLUST-2015 2D datasets (all results are in millimeters).

Individual Scores

POI	Mean	σ	95%	Min	Max
CIL					
03_1	0.93	0.49	1.81	0.08	2.89
03_2	5.07	2.84	10.17	0.71	15.06
04_1	0.95	0.44	1.78	0.21	2.44
04_2	0.89	0.45	1.75	0.02	2.08
ETH					
06-1_1	0.80	0.27	1.23	0.10	2.08
06-2_1	0.48	0.26	0.98	0.02	1.35
07-1_1	0.71	0.40	1.50	0.02	2.61
07-1_2	1.22	0.57	2.21	0.02	3.52
07-2_1	0.92	0.75	2.57	0.02	3.90
07-2_2	1.14	0.59	2.20	0.08	3.84
08-1_1	0.99	0.52	1.95	0.14	3.34
08-1_2	0.74	0.31	1.29	0.07	2.20
08-2_1	0.79	1.00	1.43	0.02	7.08
08-2_2	0.63	0.32	1.20	0.07	1.82
09-1_1	0.66	0.56	1.16	0.06	10.93
09-1_2	0.62	0.52	1.11	0.02	10.01
09-1_3	0.83	0.53	1.28	0.04	10.30
09-1_4	1.75	1.69	5.23	0.08	8.35
09-2_1	0.57	0.25	0.97	0.02	1.34
09-2_2	4.27	7.18	22.30	0.05	25.55
09-2_3	4.50	5.99	17.21	0.06	22.03
10-1_1	0.81	0.22	1.22	0.28	1.68
10-1_2	0.59	0.25	1.03	0.10	1.30
10-1_3	0.79	0.94	1.11	0.04	6.63
10-2_1	0.67	0.27	1.13	0.07	2.04
10-2_2	0.50	0.26	0.96	0.03	1.50
10-2_3	1.11	1.72	5.84	0.02	7.19
ICR					
05_1	1.51	0.42	2.20	0.53	3.02
05_2	1.05	0.37	1.70	0.24	2.36
06_1	1.90	0.61	2.77	0.64	7.89
06_2	1.78	1.30	4.70	0.01	9.67
07_1	0.97	0.29	1.46	0.18	1.91
07_2	1.73	1.21	4.12	0.03	10.92
08_1	2.45	3.02	9.73	0.08	12.37
08_2	0.81	0.30	1.30	0.12	1.72
08_3	0.72	0.25	1.14	0.12	1.38

Individual Scores

POI	Mean	σ	95%	Min	Max
MED1					
06-1_1	1.05	0.83	2.55	0.07	5.17
06-1_2	0.92	0.29	1.39	0.10	1.86
06-1_3	1.03	0.64	2.08	0.04	4.09
06-1_4	0.82	0.26	1.25	0.21	2.31
06-2_1	0.83	0.55	1.98	0.05	3.37
06-2_2	1.04	0.33	1.61	0.34	2.22
06-2_3	1.03	0.65	2.44	0.06	3.45
07-1_1	0.78	0.44	1.63	0.04	2.38
07-1_2	1.02	0.33	1.55	0.03	2.25
07-1_3	0.57	0.28	1.05	0.06	1.30
07-2_1	0.61	0.38	1.30	0.01	1.82
07-2_2	0.82	0.32	1.36	0.11	1.89
07-2_3	0.58	0.29	1.06	0.03	1.57
07-3_1	1.80	1.33	4.94	0.02	5.65
07-3_2	0.88	0.43	1.70	0.03	2.12
07-3_3	0.52	0.34	1.22	0.04	1.97
07-4_1	1.30	0.86	3.11	0.12	3.92
07-4_2	0.59	0.29	1.11	0.03	1.50
07-4_3	0.82	0.33	1.37	0.03	1.89
07-4_4	0.80	0.25	1.21	0.12	1.53
MED2					
08-1_1	0.59	0.27	1.06	0.08	1.50
08-1_2	0.78	0.36	1.36	0.07	1.98
08-1_3	0.64	0.38	1.27	0.04	2.58
08-2_1	0.55	0.22	0.94	0.08	1.38
08-2_2	0.64	0.30	1.21	0.03	1.81
08-2_3	1.13	0.83	2.93	0.07	4.89

Average scores per set

POI	Mean	σ	95%	Min	Max
CIL	2.07	2.41	7.90	0.02	15.06
ETH	1.09	2.18	2.30	0.02	25.55
ICR	1.43	1.33	3.49	0.01	12.37
MED1	0.89	0.61	1.93	0.01	5.65
MED2	0.72	0.48	1.54	0.03	4.89

Average scores for all POI

POI	Mean	σ	95%	Min	Max
—	1.09	1.75	2.42	0.01	25.55

There was a number of additional strategies that we attempted as described below, without any significant gain on average tracking error. The organ motion is rather coherent in nature, thus the relative positions of points do not change substantially. We attempted to leverage this by restricting relative positions of tracked points to one another, with some degree of relative movement allowed; nevertheless this did not improve the average tracking performance. We also attempted to reinitialize the reference frame when the tracking is judged to be correct and the motion is in the same phase, in order to take into account image appearance changes over time: For this, we used a simple approach of reinitializing a new reference when when RT is valid and all points show low error, however it was not possible to reliably detect incorrectly tracked points and reinitialization would thus create even a higher drift. Consequently, the use of only the initial reference frame (without any reinitialization) yielding the best results on average.

4 Conclusions

Our proposed algorithm has shown superior results to methods published in the previous CLUST challange. The average tracking error of 1.09 mm is relevant in liver motion-tracking for radiation and focused therapy applications. Our method runs in real-time, with average latencies of [20..70] ms in the given sequences. Aside from the given parametrization on the training dataset, no further per-machine or per-sequence parameter tuning is required.

References

1. Bar-Shaloom, Y., Fortmann, T.E.: Tracking and Data Association. New York: Academic. (1988)
2. Crimi, A., Makhinya, M., Baumann, U., Szekely, G., Goksel, O.: Vessel tracking for ultrasound-based venous pressure measurement. In: IEEE Int Symp Biomedical Imaging (ISBI). pp. 306–9 (2014)
3. Crimi, A., Makhinya, M., Baumann, U., Thalhammer, C., Szekely, G., Goksel, O.: Automatic measurement of venous pressure using B-mode ultrasound. IEEE Trans Biomedical Engineering (2015), http://dx.doi.org/10.1109/TBME.2015.2455953
4. DeLuca, V., Benz, T., Kondo, S., König, L., Lübke, D., Rothlübbers, S., Somphone, O., Allaire, S., Bell, M.A.L., Chung, D.Y.F., Cifor, A., Grozea, C., Günther, M., Jenne, J., Kipshagen, T., Kowarschik, M., Navab, N., Rühaak, J., Schwaab, J., Tanner, C.: The 2014 liver ultrasound tracking benchmark. Physics in Medicine and Biology 60(14), 5571 (2015)
5. Friedland, N., Adam, D.: Automatic ventricular cavity boundary detection from sequential ultrasound images using simulated annealing. IEEE Transactions on Medical Imaging 8, 344–353 (1989)
6. Guerrero, J., Salcudean, S., McEwen, J., Masri, B., Nicolaou, S.: Real-time vessel segmentation and tracking for ultrasound imaging applications. IEEE Transactions on Medical Imaging 26(8), 1079–1090 (2007)
7. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Procs of Imaging Understanding Workshop. pp. 121–130 (1981)

Robust Liver Ultrasound Tracking using Dense Distinctive Image Features

Andre Hallack[1], Bartłomiej W. Papież[1], Amalia Cifor[2], Mark J. Gooding[2], Julia A. Schnabel[3]

[1] Institute of Biomedical Engineering (Department of Engineering Science), University of Oxford, UK
[2] Mirada Medical, Oxford, UK
[3] Division of Imaging Sciences & Biomedical Engineering, King's College London, UK

Abstract. A new framework for tracking of anatomically relevant landmarks on 2D liver ultrasound sequences is presented in this work. It combines logDemons nonlinear registration, which estimates the motion within ultrasound sequences, with a moving window tracking method, that propagates the estimated motion around the region of interest to subsequent frames. Robust and accurate nonlinear registration is obtained by employing the dense Scale Invariant Feature Transform as a similarity measure. The proposed method was evaluated on 24 sequences from the CLUST 2015 challenge. On a total of 62 landmarks within these sequences, a mean target error of 0.91mm was achieved, surpassing the previous challenge best performance, 1.44mm on CLUST 2014.

1 Introduction

Ultrasound provides real-time, safe and affordable imaging of soft tissues, making it one of the most popular techniques for tracking of internal body structures. This however comes at the expense of considerable amounts of noise. Hence, the challenges of using this imaging technique lie very often on the image interpretation. For real time tracking of structures, such as vessels and tumours, this translates to the need of robust and equally efficient image processing methods.

In this work, an efficient solution for tracking of anatomical structures in liver ultrasound sequences is presented. These images are subject to high amounts of motion due to breathing, as well as noise and shadowing effects, causing significant intensity changes in ultrasound structure and appearance [1]. Though not persistent over time, high levels of nonlinear deformation are observed in liver ultrasound. Consequently, motion correction approaches for anatomical structure tracking cannot rely only on rigid registration methods, which are much faster than nonlinear methods.

To solve this problem, we employed a tracking framework where only the regions of interest are analysed, which greatly contributes to the system's efficiency. In this framework, nonlinear motion correction was performed using logDemons diffeomorphic registration, which has already been successfully applied to liver ultrasound tracking [2]. The main contribution of this work is the use of the

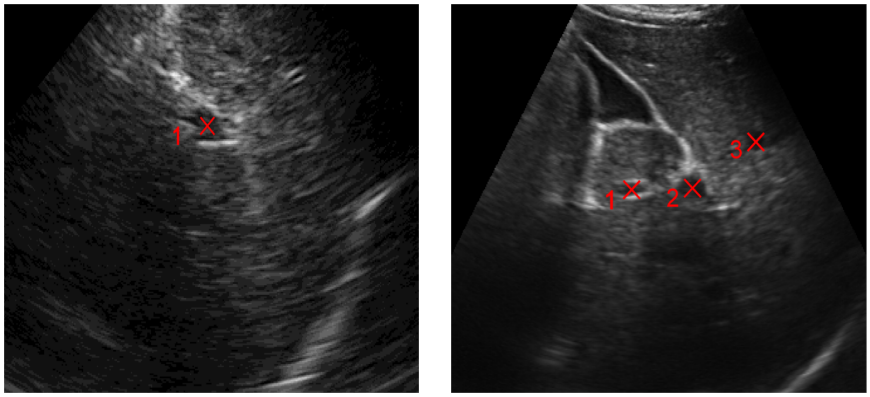


Fig. 1. Sample reference ultrasound images and landmarks from the liver ultrasound tracking challenge CLUST 2015.

dense Scale Invariant Feature Transform (dense SIFT) [3], a similarity metric which has not yet been widely explored for medical imaging registration, but has shown promise in computer vision problems to enhance distinctive features of images and to be very effective on 2D image registration.

This paper is structured as follows: in Sec. 2 the method developed to track annotations by registering liver ultrasound images is presented. This is followed in Sec. 3 with a description of the CLUST 2015[4] data used evaluate the proposed method, as well as the experiments conducted and their respective results. Finally, Sec. 4 concludes this work.

2 Methods

The task addressed in the CLUST 2015 registration challenge can be described as: given a sequence of temporal 2D images I_t, to estimate a set of annotation positions $\boldsymbol{x} = M(t)$ over time based on the initial position $M(1)$ of a relevant structure. Here, $\boldsymbol{x}$ is a 2D spatial location and t the frame index in the sequence. Fig. 1 presents examples of ultrasounds images used in this work and the landmarks being tracked.

In our method, we opted to use an image registration approach. Hence, by computing a nonlinear transformation field T_t that registers I_1 to I_t, $\hat{M}(t)$ can be estimated as $T_t(M(1))$.

One of the difficulties of applying registration methods for such tasks is that over long periods of time, large amounts of complex deformation and displacement will occur in the images, hindering the registration process between far apart acquisitions. On the specific case of liver ultrasound, nonlinear deformations are present for the observed structures, and thus nonlinear registration is necessary to correctly identify the structures over time. However, over short

[4] *CLUST 2015*, http://clust.ethz.ch/

acquisitions, the main persistent type of global motion is rigid. Hence, we opted first for a tracking method which propagates the estimated rigid motion at each time point to the next frame, followed by accurate deformable registration.

The proposed method consists of a tracking framework (similar to [4]), where the images are cropped around the expected annotation location (Sec. 2.1), and a logDemons diffeomorphic nonlinear registration framework (Sec. 2.2), using dense SIFT, which is a highly descriptive image transform (Sec. 2.3).

2.1 Tracking

The main concept of the tracking method used here was to perform image registration between cropped patches of the image sequence. Hence, from each frame I_t, a square region $W_{t-1}(I_t)$ is extracted. W_{t-1} determines the position around where this patch should be extracted. For the first frame, I_1, this is straightforward, since W_0 is the position of the initial annotation $M(1)$. For each subsequent frame I_t ($t > 1$), W_{t-1} is extracted around the previously estimated annotation location $\hat{M}(t-1)$. An overview of this method is shown on Algorithm 1. Each cropped patch has w x w pixels centred on the estimated annotated position. This method does not propagate the whole nonlinear transformation from frame to frame, but only the translation found for the annotation.

Fig. 2 presents an example of how the tracking framework progresses by propagating the previous estimated annotation location to each subsequent frame. This method was based on the work by König *et al.* [4]. Unlike that work, here, for the cases where several different annotated structures are present in the same imaging sequence, each annotation was tracked independently. Another difference to that framework is that no upper motion bounds were applied to the obtained transformations.

Data: Liver ultrasound 2D sequence I_t and the initial landmark position $M(1)$.
Result: Sequence of estimated landmark positions $\hat{M}(t)$ for each ultrasound sequence frame.

$W_0(I_1) \leftarrow crop(I_1, M(1))$
$\hat{M}(1) \leftarrow M(1)$
$t \leftarrow 2$
while $t <$ *number of frames* **do**
  $W_{t-1}(I_t) \leftarrow crop(I_t, \hat{M}(t-1))$
  $T_t \leftarrow register(W_{t-1}(I_t), W_0(I_1))$
  $\hat{M}(t) \leftarrow T_t(M(1))$
  $t \leftarrow t + 1$
end

Algorithm 1: Tracking method for liver ultrasound landmarks.

2.2 Nonlinear Registration

For each image I_t of the sequence, its cropped subregion around the expected annotation position $W_{t-1}(I_t)$ is registered to the reference cropped image $W_0(I_1)$,

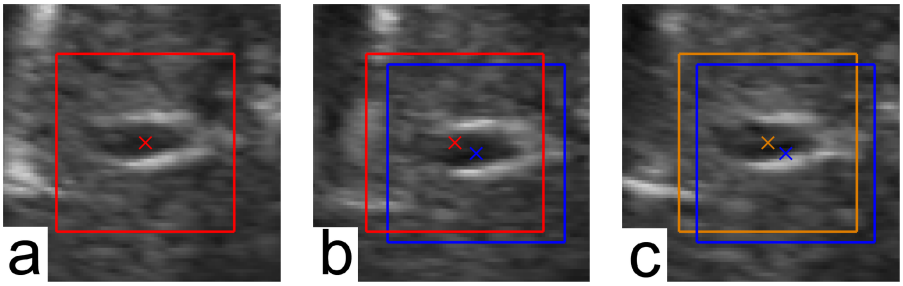


Fig. 2. Sequence of images exemplifying the tracking framework used in this work. On the first (reference) frame (a), a region of interest (red square) is cropped around the known landmark ($M(1)$ - red cross). This location is used as the region of interest for the next frame (b), indicated by the red square. Image registration between these two cropped regions (a and b - red squares) leads to a new estimation of the annotation position (blue cross). For the next frame (c), the latest estimated annotated position is used as the centre of the region of interest (blue square and cross) which will be used to register to the reference region of interest (a), and after motion correction finds a new position for the tracked structure (orange cross). This process is repeated for all frames of an ultrasound sequence.

estimating the nonlinear transformation T_t. This transformation can then be used to estimate the annotation position for the current frame, $\hat{M}(t) = T_t(M(0))$. For this nonlinear registration step we employed a diffeomorphic logDemons framework [5]. Demons is an iterative registration method which finds a non-regularised deformation field T_c by minimizing a similarity measure (Sim), but at the same time computes a smooth regularised version (T_s) by applying a Gaussian filter to it [6]. This is described by the following optimization problem:

$$T = \arg\min_{T_s}\Big(\mathrm{Sim}\left(W_0(I_1), T_c(W_{t-1}(I_t)\right) + \mathrm{Dist}(T_s, T_c) + \mathrm{Reg}(T_s)\Big) \qquad (1)$$

where usually $\mathrm{Dist}(T_s, T_c) = ||T_s - T_c||^2$ and $\mathrm{Reg}(T_s) = ||T_s||^2$.

The logDemons version of this method ensures that the obtained transformation is invertible by restricting T_s to a subspace of diffeomorphisms (see details in [5]).

2.3 Dense SIFT

Due to the intensities distortions mentioned in Sec. 1, most intensity-based measures of similarity lack the robustness to accurately register such ultrasound image [2]. Hence, in this work, image intensities were not directly compared during the logDemons registration steps. At each registration iteration, these images were transformed using the dense Scale Invariant Feature Transform (dense SIFT), a modified version of SIFT for dense image analysis [3, 7, 8]. This method computes at each voxel a descriptor vector based on the histogram of

gradients around its neighbourhood, generating a vector-valued images. This enhances distinctive characteristics of the ultrasound images and therefore allows for more accurate registration.

Transformed images using dense SIFT (SIFT($I(\boldsymbol{x})$) can then be locally compared as the sum-of-square-differences (SSD) of the dense SIFT feature vectors at each voxel $\boldsymbol{x}$:

$$\mathrm{Sim}_{\mathrm{SIFT}}(I_1(\boldsymbol{x}), I_t(\boldsymbol{x})) = \left|\left|\mathrm{SIFT}(I_1(\boldsymbol{x})) - \mathrm{SIFT}(T(I_t(\boldsymbol{x})))\right|\right|_2^2 \quad (2)$$

This similarity measure is used within the logDemons framework (Eq. 1).

2.4 Parameters

The cropped region around each annotation was of 51 by 51 (w x w) pixels, the size of this region was chosen to safely contain the whole annotated structure. For nonlinear registration, the logDemons framework was applied with three resolution levels with 20 iterations at each level and transformation field smoothing $\sigma_{\mathrm{diff}} = 2$ pixels, these parameters were not optimised for this problem. The SIFT Flow library was used for dense SIFT with the standard parameters: cell size = 2 and 8 bins [3].

3 Experiment, Results and Discussion

The proposed framework was evaluated on 24 2D+t liver ultrasound sequences (from 4 different scanners) designated for the CLUST 2015 challenge. These sequences showed a spatial resolution between 0.30mm and 0.55mm, number of frames ranging from 895 to 5586 and image rate from 11Hz to 23Hz. A total of 62 landmarks were provided (between 1 and 4 per sequence) at the initial frame to be tracked over the whole sequence. The method was assessed by the challenge organizers in terms of the mean target error (MTE) and standard deviation (σ) between the computed and ground truth landmarks at selected frames, as well as its 95th percentile. For each of these sequences, the detailed results are presented in Tab. 1 and the overall outcome for each different scanner is shown in Tab. 2.

The MTE over all the sequences and landmarks was 0.91mm with standard deviation of 1.66mm. These results were below the ones obtained on the previous ultrasound liver tracking competition, CLUST 2014, where the best reported result was 1.44mm MTE and 2.04mm stadard deviation on a similar dataset [9].

Despite using the same parameters for all analysis, the results obtained over different scanners did not vary much; for the best case (dataset ETH) the MTE was 0.59mm and for the worse case (CIL) it was 1.74mm. A noteworthy result was for sequence MED 0.6-1, where for one of the annotations (2) this framework clearly lost track of the landmark and showed very large errors. This highlights one of the possible flaws of the proposed tracking framework: by reducing the analysis to cropped regions around the expected annotation location we are prone to failures if the motion between frames is close to the size of the region

of interest (this effect is further illustrated in Fig. 3). Increasing the size of the cropped regions could help reducing this error, however, it will lead to longer computation time.

This algorithm was implemented on a single thread C++ program and tested on a Intel i7-3770 computer with 3.40GHz, Ubuntu Linux 12.04 operating system. The average frame processing speed was of 4.8 images per second, which is close, but short of the acquisition rate of these sequences. Since the analysis for each of the landmarks in a sequence is done independently, this speed is directly proportional to the number of annotations, and the experiments showed a rate of 11.8 frames per second per annotation.

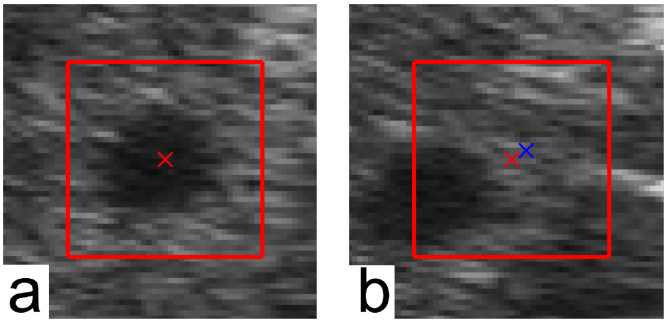


Fig. 3. Illustration of an specific sequence (MED-06-1 - landmark 2) where the proposed method loses track of the structure of interest. On (a) the estimated landmark position (red cross) is centred on the structure of interest. On the subsequent frame (b) there is a large displacement of the structure of interest, since now the estimated landmark position from the previous frame (red cross) and the cropped region of interest (red square) do not contain the whole the structure. Here, the nonlinear registration method fails to follow the observed displacement, and finds an erroneous new estimated landmark position (blue cross).

4 Conclusions

In this work a new method for liver 2D ultrasound tracking was proposed and evaluated within the CLUST 2015 challenge. Our method combined a nonlinear registration method with a tracking method which focused only on the region of interest around the tracked annotation, which then propagated the translational information from previous frames to the next frame. The nonlinear image registration between the cropped reference and moving regions of interest (at each frame) around the observed structure were performed using logDemons. One of the main advantages of the proposed solution is the use of dense SIFT as a similarity measure, a feature transform which led to better characterization of the observed structures than standard intensity based measures.

Table 1. CLUST 2015 2D tracking results of the 62 sequences. The outcome is measured in terms of the mean target error (MTE) and standard deviation (σ) in millimetres for each landmark in each sequence.

	Results per landmark (mm)							
Dataset	MTE_1	σ_1	MTE_2	σ_2	MTE_3	σ_3	MTE_4	σ_4
CIL 03	1.51	1.53	2.92	1.04				
CIL 04	1.35	0.50	0.95	0.47				
ETH 06-1	0.68	0.29						
ETH 06-2	0.73	0.36						
ETH 07-1	0.34	0.16	0.47	0.29				
ETH 07-2	0.43	0.23	1.08	0.46				
ETH 08-1	0.45	0.18	0.62	0.38				
ETH 08-2	0.62	0.21	0.90	0.44				
ETH 09-1	0.59	0.59	0.47	0.52	0.64	0.93	0.54	0.56
ETH 09-2	0.45	0.24	0.86	0.46	0.91	0.43		
ETH 10-1	0.40	0.23	0.47	0.25	0.41	0.18		
ETH 10-2	0.41	0.29	0.56	0.29	0.57	0.57		
ICR 05	0.66	0.22	1.02	0.54				
ICR 06	0.74	0.25	1.29	0.54				
ICR 07	0.66	0.27	0.84	0.63				
ICR 08	0.58	0.48	0.37	0.22	1.03	1.66		
MED 06-1	1.93	0.89	7.73	12.26	1.15	0.50	0.98	1.33
MED 06-2	1.64	0.92	0.73	0.51	1.59	1.05		
MED 07-1	0.98	0.48	0.86	0.31	1.25	0.70		
MED 07-2	1.00	0.48	0.65	0.43	0.79	0.30		
MED 07-3	3.06	1.92	0.86	0.44	0.79	0.42		
MED 07-4	2.85	1.67	0.60	0.38	0.83	0.39	0.36	0.18
MED 08-1	0.70	0.41	1.63	0.60	0.68	0.36		
MED 08-2	0.95	0.37	1.07	0.47	2.57	1.32		

The highly promising results under the CLUST 2015 challenge attested the validity of the proposed method, showing a performance at least comparable to state-of-the-art solutions and close to real-time speed. We also were able to identify the main conditions where our framework fails at this tracking problem, inciting the development of solutions which can handle well large deformations between adjacent frames. Further development can also be made in terms of the optimisation of the method's parameters both to improve its accuracy as well as to reach real-time processing of ultrasound sequences (including mult-thread computing).

5 Acknowledgments

We would like to acknowledge funding from the CRUK/EPSRC Cancer Imaging Centre at Oxford.

AH also gratefully acknowledges the support of the Research Council UK Digital Economy Programme grant number EP/G036861/1 (Oxford Centre for Doc-

Table 2. CLUST 2015 2D tracking results for each ultrasound scanner, as well as the final overall result. The outcome is given by the mean target error (MTE), standard deviation (σ) and minimum (Min) and maximum (Max) error in millimetres for each scanner group.

	Combined results (mm)					
Dataset	MTE	σ	95th	Min	Max	Scanner Type
CIL	1.74	1.27	4.23	0.04	6.58	Ultrasonix MDP
ETH	0.59	0.44	1.29	0.01	12.45	Siemens Antares
ICR	0.80	0.73	1.81	0.01	7.78	Elekta Clarity - Ultrasonix
MED 1	1.53	3.25	3.99	0.01	35.20	DiPhAs Fraunhofer
MED 2	1.27	0.95	3.48	0.02	6.48	Zonare z.one
Overall	**0.91**	**1.66**	**2.20**	**0.01**	**35.20**	

toral Training in Healthcare Innovation) and the funding provided by CAPES Foundation, process N^o BEX 0725/12-9.

References

1. J. A. Noble, N. Navab, and H. Becher, "Ultrasonic image analysis and image-guided interventions," *Interface focus*, vol. 1, no. 4, pp. 673–685, 2011.
2. A. Cifor, L. Risser, D. Chung, E. M. Anderson, J. Schnabel, *et al.*, "Hybrid feature-based diffeomorphic registration for tumor tracking in 2-d liver ultrasound images," *Medical Imaging, IEEE Transactions on*, vol. 32, no. 9, pp. 1647–1656, 2013.
3. C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE T Pattern Anal*, vol. 33[5], pp. 978–94, 2011.
4. L. König, T. Kipshagen, and J. Rühaak, "A non-linear image registration scheme for real-time liver ultrasound tracking using normalized gradient fields," *Challenge on Liver Ultrasound Tracking CLUST 2014*, p. 29, 2014.
5. T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Symmetric log-domain diffeomorphic Registration: a demons-based approach." *MICCAI*, vol. 11[1], pp. 754–61, 2008.
6. J. Thirion, "Image matching as a diffusion process: an analogy with Maxwell's demons," *Med Image Anal*, vol. 2[3], pp. 243–60, 1998.
7. A. Hallack, B. W. Papież, J. Wilson, L. M. Wang, T. Maughan, M. J. Gooding, and J. A. Schnabel, "Correlating tumour histology and *ex vivo* mri using dense modality-independent patch-based descriptors," *Submitted Work*, 2015.
8. D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2. IEEE, 1999, pp. 1150–7.
9. V. De Luca, T. Benz, S. Kondo, L. König, D. Lübke, S. Rothlübbers, O. Somphone, S. Allaire, M. a. Lediju Bell, D. Y. F. Chung, a. Cifor, C. Grozea, M. Günther, J. Jenne, T. Kipshagen, M. Kowarschik, N. Navab, J. Rühaak, J. Schwaab, and C. Tanner, "The 2014 liver ultrasound tracking benchmark." *Physics in medicine and biology*, vol. 60, no. 14, pp. 5571–99, July 2015.

Tracking of Non-rigid Targets in 3D US Images: Results on CLUST 2015

Lucas Royer[1,2,3], Guillaume Dardenne[1], Anthony Le Bras[1,4], Maud Marchal[1,3], Alexandre Krupa[1,2]

[1] Institut de Recherche Technologique b-com, Rennes, France
[2] Inria Rennes - Bretagne Atlantique, France
[3] INSA de Rennes, France
[4] CHU de Rennes, France

Abstract. In ultrasound-guided procedures, such as high-intensity focused ultrasound or radio-frequency ablation, non-rigid clinical targets may undergo displacements due to physiological motions. To cope with that issue, the accurate estimation of the target motion is required in order to adjust the position of medical tools. In this paper, we present a robust approach that allows to track in real-time deformable targets in 3D ultrasound images. Our method combines visual motion estimation with a mechanical model of the target. Our approach is evaluated on the MICCAI CLUST'15 challenge 3D database. We achieved a mean tracking error of 1.78 mm with an average computation time of 350 ms per frame.

1 Introduction

Over the last few years, minimally-invasive procedures, such as high-intensity focused ultrasound (HIFU), or radio-frequency ablation (RFA) have gained significant attention due to the shorter recovery time compared to conventional therapies. However, the quality of these therapies can strongly depend on both the deformations and displacements of the clinical targets since the surgeon needs to continuously adjust the positions of medical tools. Thus, to ensure the target visibility under ultrasound (US) image guidance, several target tracking methods have been developed [8–14]. We recently presented in [7] a robust approach that combines a mechanical model and visual estimation. The good performance of this method has been showed on data obtained from a deformable soft tissue phantom. In this paper, we demonstrate that this method can achieve high accuracy on real-data by testing our algorithm on the database proposed by the MICCAI CLUST'15 challenge. The rest of the paper is organized as follows. In section 2, we present the method that allows to track deformable target in 3D ultrasound images. In section 3, we describe the performance of our approach on real-data. Finally, section 4 concludes the paper.

2 Method

The objective of our approach is to track the motions of a clinical target in 3D ultrasound sequence. To do so, we manually segment the target within the first 3D ultrasound image and we generate a corresponding 3D tetrahedral mesh model (section 2.1). Once the model is defined, we need to estimate the target motions over the consecutive frames. For this purpose, the model displacements are obtained by iteratively summing the internal displacements estimated from a mechanical component (section 2.2), and the external displacements computed from an intensity-based approach (section 2.3).

2.1 Piece-wise Affine Model

In 3D US images, a clinical target can be represented by a continuous set of N_v voxels that is delimited by a visible border. Typical examples are shown in Fig. 2. In order to define the target, we first extract its shape at the initial 3D frame of the US sequence by performing a segmentation. To remove sharp edges and discontinuous shapes, a smoothing step is performed on the 3D segmented surface and a corresponding fitted tetrahedral mesh containing N_c vertices is defined. Then, in order to represent the displacements of the voxels, we propose to use a piece-wise affine warp function. Our piece-wise affine warping is parameterized from both the vertice positions and an affine interpolation that uses barycentric coordinates as proposed in [15]. In this way, we can relate all the voxel positions with all the vertices as follows:

$$\mathbf{p}_{im} = \mathbf{M}.\mathbf{q} \tag{1}$$

where $\mathbf{M}$ is a $(3 \cdot N_v) \times (3 \cdot N_c)$ constant matrix defining the set of barycentric coordinates. $\mathbf{p}_{im}$ is a $(3 \cdot N_v) \times 1$ vector defining all the voxels positions, and $\mathbf{q}$ is a $(3 \cdot N_c) \times 1$ vector containing all the vertices positions. Thanks to Eq. (1), we can update the positions of the target when the vertices of the model are displaced. To compensate the lack of smoothness as well as the poor estimation of vertice positions in US images, we combine a mechanical model to the estimation of displacement.

2.2 Mechanical Component

Our approach combines a mass-spring-damper system to the mesh model previously described. Thus, the vertice displacements are constrained by linking each connected vertice pair with a spring ensuring physically-plausible and coherent displacements of the vertices. Furthermore, the mass-spring-damper system can be specifically characterized by setting a mass value to each vertex, together with elastic and damping coefficients on each spring depending on the soft-tissues homogeneities. In this work, these values have been set empirically but another solution could consist in estimating these parameters from elastography images. However, a further study is required in order to define the impact

on the accuracy of the approach. From this model, we can compute the force $\mathbf{f}_{ij} = [f_{x_{ij}}\ f_{y_{ij}}\ f_{z_{ij}}]^T$ applied on a vertice $\mathbf{q}_i$ from a neighbor vertice $\mathbf{q}_j$. This force can be expressed as follows:

$$\mathbf{f}_{ij} = K_{ij}(d_{ij} - d_{ij}^{init})(\mathbf{q}_i - \mathbf{q}_j) + D_{ij}(\dot{\mathbf{q}}_i - \dot{\mathbf{q}}_j) \circ (\mathbf{q}_i - \mathbf{q}_j) \tag{2}$$

where d_{ij} and d_{ij}^{init} respectively represent the distance between the vertices $\mathbf{q}_i$ and $\mathbf{q}_j$ at their current positions and at their initial positions. The $\circ$ operator expresses the Hadamard product, K_{ij} is a scalar value denoting the stiffness of the spring that links the two vertices while D_{ij} is the damping coefficient value. By combining the previous equation for all the vertices, we can express the total amount of forces $\mathbf{f}_i$ exerted on each vertice $\mathbf{q}_i$ of the mesh model as follows:

$$\mathbf{f}_i = \sum_{n=0}^{N_i} \mathbf{f}_{in} + G_i \dot{\mathbf{q}}_i \tag{3}$$

N_i denotes the number of neighbors vertices connected to the vertice $\mathbf{q}_i$. G_i represents the velocity damping coefficient associated to the vertice $\mathbf{q}_i$. In order to obtain the displacements $\Delta\mathbf{d}$ associated to the mass-spring-damper system, we integrate the forces expressed in Eq. (3) with a semi-implicit Euler integration scheme. Such mechanical constraint can ensure the smoothness warping function of the deformation and limits the noise sensitivity of the intensity-based approach.

2.3 Additive Update Tracking

Let us recall that the main objective of our approach is to estimate the new positions of the target in 3D US sequence. To do so, we use an intensity-based method that consists in evaluating the vertice displacements by minimizing a dissimilarity function E. Therefore, we can express the cost function which minimizes image dissimilarity from the relationship described in Eq. (1) such that:

$$C(\Delta\mathbf{q}) = E(I_t(\mathbf{p}_{im}(t)), I_{t_0}(\mathbf{p}_{im}(t_0))) = E(I_t(\mathbf{M}\mathbf{q}(t)), I_{t_0}(\mathbf{M}\mathbf{q}(t_0))) \tag{4}$$

where I_{t_i} is a vector representing the US intensity of the volume acquired at time index t_i. $\Delta\mathbf{q}$ denotes the vertices displacements. $\mathbf{p}_{im}(t_i)$ and $\mathbf{q}(t_i)$ represent respectively the voxel positions and the vertice positions at time index t_i. In order to determine the dissimilarity function E, we assume that the intensity values of soft tissues are consistent over the time. Consequently, we propose to use the Sum of Squared Differences (SSD) in order to measure the image error. The cost function can now be expressed as:

$$C(\Delta\mathbf{q}) = (I_t(\mathbf{M}(\mathbf{q}(t))) - I_{t_0}(\mathbf{M}(\mathbf{q}(t_0))))^2 \tag{5}$$

The objective is to find iteratively the vertice displacements by minimizing the cost function C. To do so, we perform a Taylor expansion of the previous equation that leads to:

$$C(\Delta\mathbf{q}) \approx (\mathbf{J}\Delta\mathbf{q} + I_t(\mathbf{M}(\mathbf{q}^{k-1}(t))) - I_{t_0}(\mathbf{M}(\mathbf{q}(t_0))))^2 \tag{6}$$

where $\mathbf{q}^{k-1}(t)$ represents the estimation of the parameters at time t at iteration $k-1$ of the optimization algorithm. $\mathbf{J}$ denotes the Jacobian matrix associated to the cost function. This matrix relates the variation of the parameters $\Delta\mathbf{q}$ with the intensity variation of I. It can be computed as follows:

$$\mathbf{J} = \nabla I.\mathbf{M} \tag{7}$$

where ∇I denotes the gradient of the current 3D US frame. In order to obtain the optimal displacements of the vertices, we chose to use a forward-additive steepest gradient strategy as it is computationally efficient since it does not require the calculation of pseudo-inverse of large Jacobian matrix. We therefore obtain:

$$\Delta\mathbf{q} = -\alpha\mathbf{J}^t[I_t(\mathbf{M}(\mathbf{q}^{k-1}(t))) - I_{t_0}(\mathbf{M}(\mathbf{q}(t_0)))] \tag{8}$$

where $\alpha > 0$ denotes the step size of the minimization strategy. $\mathbf{J}^t$ represents the transpose matrix of the Jacobian $\mathbf{J}$. As stated previously, in order to prevent inaccurate results, we propose to combine this motion estimation with the internal displacements of the mass-spring-damper system. This can be performed by iteratively estimate the optimal displacement as follows:

$$\mathbf{q}_k(t) = \mathbf{q}_{k-1}(t) + \Delta\mathbf{q} + \Delta\mathbf{d} \tag{9}$$

where $\Delta\mathbf{d}$ is the internal displacements obtained from the integration of forces expressed in Eq. (3). $\Delta\mathbf{q}$ represents the external displacements from the steepest gradient strategy in Eq. (8). $\mathbf{q}_{k-1}(t)$ denotes the estimation of vertice position at iteration $k-1$ and at time index t. In order to balance the influence of the mass-spring-damper system regarding to the motion estimation between $\Delta\mathbf{q}$ and $\Delta\mathbf{d}$, we can tune the α coefficient that represents the step size of the minimization strategy in Eq.(8).

3 Results

3.1 Description of our Evaluation Environment

Our approach has been tested on real-data and has been implemented with C++/GPU code by using Cuda and VTK libraries. The segmentation step in the first volume is performed with the ITK-SNAP [16] software and can be executed in less than 3 minutes. The mesh is generated thanks to the tetGen [17] software. The resulted computation time of the online tracking is 350 ms allowing thus real-time capabilities. The code was executed on a Windows 7 machine with an Intel core i7-3840qm(2.80GHz).

3.2 Validation Results on Real-data

In order to evaluate our method, we used the database provided by the workshop MICCAI CLUST'15 challenge. The main goal of this challenge is to compare

different state-of-the-art methods for tracking anatomical landmarks in US sequences. For this purpose, a database containing 2D/3D ultrasound sequences of volunteers under free breathing is provided. Furthermore, in order to generate ground truth data, the positions of the target landmarks are identified from expert annotations for each frame. Thus, a comparison can be performed between the ground truth landmark positions and the point position estimated from the tracking task over each frame. The point position at each frame is retrieved by using the equation (1). Is is worth mentioning that both the ultrasound sequences and the annotations are provided by several research institutes. Thus, the approach has been tested by tracking 32 different anatomical features acquired from 16 3D US sequences. For each experiment, we set the elastic and damping parameters such that $K_{ij} = 3.0$ and $D_{ij} = 0.1$ for all the springs, along with $G_i = 2.7$ for all vertices. These values have been set empirically by comparing the tracking accuracy with different set of parameters. This evaluation has been performed on the training database of MICCAI CLUST'15 challenge. The step size of the steepest gradient method has been set to $\alpha = 2 \times 10^{-6}$. In future work, we plan to automatically estimate these parameters by using elastography images. The preliminary results are reported in Fig. 1. From this figure, we can

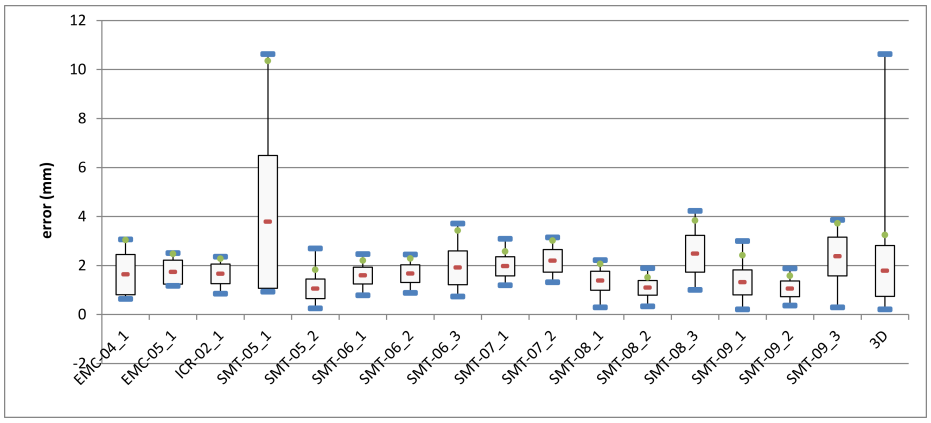


Fig. 1. Tracking error results for each tracking task. The x-axis and y-axis represent respectively each ultrasound sequence and the associated tracking error expressed in millimeters. The name of the sequence (e.g. SMT-05_1) represent both the acronym of the institute and the sequence index. The right box plot "3D" represents the average results for all the sequences (Red) Mean tracking error estimated from euclidean distance. (Black box) Mean error $\pm$ standard deviation. (Whiskers) Minimum and maximum errors. (Green dot) 95^{th} percentile of error.

notice that our approach performs accurate tracking since the mean tracking error is under 2.5 mm for most of the ultrasound sequences. However, we can observe that we obtained some unsatisfactory results (SMT-05_01) when the target goes out of the field of view (FOV). In this case, the error is introduced by

the voxels that are not within the field of view since they have a strong negative influence in the cost function (Eq. 5). This issue can be tackled by using prescan data as suggested in [7]. The performance of our approach is also illustrated in Fig. 2 showing the tracking results at different frames on four landmarks representing hepatic vein bifurcations. In addition, we can notice that our method remains robust with empirical parameters.

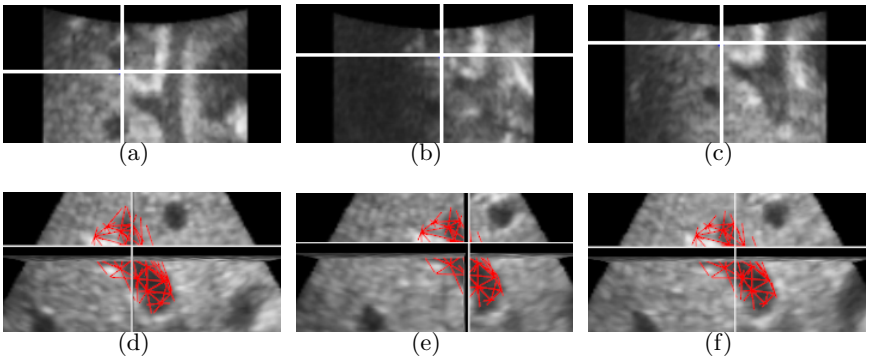


Fig. 2. Example of the tracking task on two sequences. (a-b-c) Tracking of landmark representing portal vein birfucation at frame index 00 (a), 23 (b), 59 (c). (d-e-f) Tracking of landmark representing first degree bifurcation of hepatic bile duct. The red model represents the associated 3D mesh model at frame index 00 (d), 28 (e), 78 (f).

4 Conclusion

In this paper, we presented a method for tracking and automatically compensating the displacements of a deformable target in 3D ultrasound images. The robustness of our tracking method is ensured by combining a mechanical model to the displacement estimation. We evaluated the good performance of our approach through CLUST'15 challenge database. In future work, we plan to automatically estimate the elastic parameters by using elastography images.

References

1. D. Angelova and L. Mihaylova, "Contour segmentation in 2d ultrasound medical images with particle filtering," *Machine Vision and Applications*, vol. 22, no. 3, pp. 551–561, 2010.
2. F. Yeung, S. F. Levinson, D. Fu, and K. J. Parker, "Feature-adaptive motion tracking of ultrasound image sequences using a deformable mesh," *IEEE Trans. on Medical Imaging*, vol. 17, no. 6, pp. 945–956, 1998.
3. A. Basarab, H. Liebgott, F. Morestin, A. Lyshchik, T. Higashi, R. Asato, and P. Delachartre, "A method for vector displacement estimation with ultrasound imaging and its application for thyroid nodular disease," *Medical Image Analysis*, vol. 12, no. 3, pp. 259–274, June 2008.

4. I. Mikic, S. Krucinski, and J. D. Thomas, "Segmentation and tracking in echocardiographic sequences: active contours guided by optical flow estimates," *IEEE Trans. on Medical Imaging*, vol. 17, no. 2, pp. 274–284, 1998.
5. D. Lee and A. Krupa, "Intensity-based visual servoing for non-rigid motion compensation of soft tissue structures due to physiological motion using 4d ultrasound," in *Proc. of IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 2831–2836.
6. O. Somphone, S. Allaire, B. Mory, and C. Dufour, "Live Feature Tracking in Ultrasound Liver Sequences with Sparse Demons," in *Proc. of MICCAI Workshop on Challenge on Liver Ultrasound Tracking*, 2014, p. 53.
7. L. Royer, M. Marchal, A. Le Bras, G. Dardenne, and A. Krupa, "Real-time Tracking of Deformable Target in 3d Ultrasound Images," in *IEEE Int. Conf. on Robotics and Automation*, 2015.
8. V. De Luca, M. Tschannen, G. Szkely, and C. Tanner, "A learning-based approach for fast and robust vessel tracking in long ultrasound sequences," in *Medical Image Computing and Computer-Assisted InterventionMICCAI 2013*. Springer, 2013, pp. 518–525.
9. V. De Luca, T. Benz, S. Kondo, L. Knig, D. Lbke, S. Rothlbbers, O. Somphone, S. Allaire, M. A. Lediju Bell, D. Y. F. Chung, A. Cifor, C. Grozea, M. Gnther, J. Jenne, T. Kipshagen, M. Kowarschik, N. Navab, J. Rhaak, J. Schwaab, and C. Tanner, "The 2014 liver ultrasound tracking benchmark," *Physics in Medicine and Biology*, vol. 60, no. 14, pp. 5571–5599, July 2015.
10. F. Preiswerk, V. De Luca, P. Arnold, Z. Celicanin, L. Petrusca, C. Tanner, O. Bieri, R. Salomir, and P. C. Cattin, "Model-guided respiratory organ motion prediction of the liver from 2d ultrasound," *Medical Image Analysis*, vol. 18, no. 5, pp. 740–751, July 2014.
11. M. Lediju, B. C. Byram, E. J. Harris, P. M. Evans, J. C. Bamber, and others, "3d Liver tracking using a matrix array: Implications for ultrasonic guidance of IMRT," in *Ultrasonics Symposium (IUS), 2010 IEEE*. IEEE, 2010, pp. 1628–1631.
12. M. Lediju Bell, B. Byram, E. Harris, P. Evans, and J. Bamber, "In vivo liver tracking with a high volume rate 4d ultrasound scanner and a 2d matrix array probe," *Physics in Medicine and Biology*, 2012.
13. 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 *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on*. IEEE, 2013, pp. 792–795.
14. J. Banerjee, C. Klink, E. D. Peters, W. J. Niessen, A. Moelker, and T. van Walsum, "4d Liver Ultrasound Registration," in *Biomedical Image Registration*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, A. Kobsa, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, D. Terzopoulos, D. Tygar, G. Weikum, S. Ourselin, and M. Modat, Eds. Cham: Springer International Publishing, 2014, vol. 8545, pp. 194–202.
15. I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.
16. P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *NeuroImage*, vol. 31, no. 3, pp. 1116–1128, July 2006.
17. H. Si, "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator," *ACM Trans. on Mathematical Software*, vol. 41, no. 11, 2015.

A combined tracking and registration approach for tracking anatomical landmarks in 4D ultrasound of the liver

Jyotirmoy Banerjee[1,2], Camiel Klink[1], Erwin Vast[1,2], Wiro J. Niessen[1,2], Adriaan Moelker[1] and Theo van Walsum[1,2]

[1] Dept. of Radiology, Erasmus MC, Rotterdam, The Netherlands
[2] Dept. of Medical Informatics, Erasmus MC, Rotterdam, The Netherlands
{*j.banerjee, t.vanwalsum*}*@erasmusmc.nl*

Abstract. In this paper we present a method for tracking of anatomical landmarks in the liver. Our 4D ultrasound tracking method is based on global and local rigid registration schemes. We evaluate our method on the dataset that was presented in the MICCAI 2015 Challenge on Liver Ultrasound Tracking (CLUST 2015). On the test set a mean tracking error of 1.62 ± 0.94 mm is achieved.

1 Introduction

Ultrasound (US) is used by clinicians to image the human anatomy. It is an inexpensive, non-invasive and portable imaging modality. It is widely used in diagnostics. As US imaging is real-time it can be used for interventions and therapy. The anatomy can be tracked real-time. Some of the applications are tissue motion analysis and image guidance during interventions. One of the main purpose of an US tracking approach is to incorporate (pre-operative) planning information (to guide visualization), or to integrate preoperative imaging data during interventions. Tracking or motion compensation algorithms helps to negate the motion caused by the probe or the patient and the breathing motion in particular.

Several methods for tracking of anatomical landmarks [5, 7, 8] and motion tracking of liver [4, 6, 9–12] in US have been proposed in literature. Our method is based on the previous work described in Banerjee et al. [2] and [3]. The previous methods were developed to track/register US liver volumes. In this work these methods are used to perform the specific task of tracking anatomical landmarks in the liver. The method is evaluated on the CLUST 2015 challenge datasets.

2 Tracking anatomical landmarks

We briefly review the register to reference strategy (RTR) [3] in Subsection 2.1 and the register to reference by tracking strategy (RTRT) [2] in Subsection 2.2 which are the core components of our landmark tracking approach. In Subsection 2.3 we discuss the landmark tracking approach, see the block diagram in Figure 1.

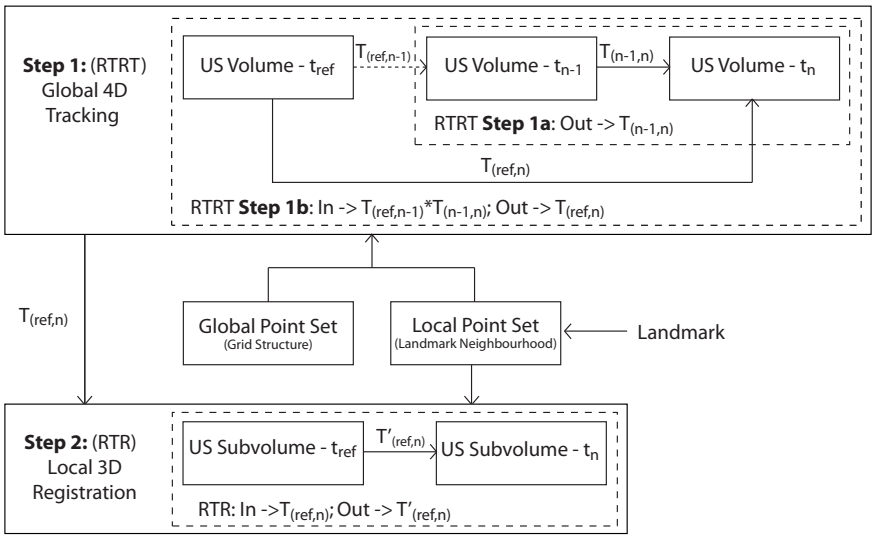


Fig. 1. Block diagram - tracking of anatomical landmarks.

2.1 Step 1: RTR [3]

The RTR approach [3] is a 3D to 3D US registration technique where the streaming input frame (t_n) is directly registered to the reference frame (t_{ref}). It is based on a block-matching scheme followed by an outlier-rejection scheme. For a set of points (generated using a grid structure or a Gaussian distribution, see the global point set and the local point set in Figure 2) located in the fixed image, block-matching is used to find corresponding locations in the moving image. The correspondences from the block-matching are inputs to the outlier rejection scheme. The outlier rejection scheme uses geometric and appearance consistency criteria to determine the block-matching results that can be trusted. The method then uses only the selected block-matching results from the outlier rejection scheme to estimate a rigid transformation using the approach described by Arun et al. [1]. For details refer [3].

2.2 Step 2: RTRT [2]

The RTRT approach [2] is a 4D US registration/tracking technique, where the registration is performed in two steps. In the first step (Step 1a), the streaming input frame (t_n) is registered to the previous frame in the temporal domain (t_{n-1}). In the second step (Step 1b), the previously estimated transformation ($T_{(ref,n-1)}$) and the transformation from the first step ($T_{(n-1,n)}$) are used to initialize the registration between the streaming input frame and the reference frame, by composing the transformations as $T_{(ref,n-1)} * T_{(n-1,n)}$. To reduce the accumulation of error the reference frame is re-registered to the streaming input frame which was earlier transformed using the transformation $T_{(ref,n-1)} * T_{(n-1,n)}$, resulting in the final transformation, written as $T_{(ref,n)}$, see

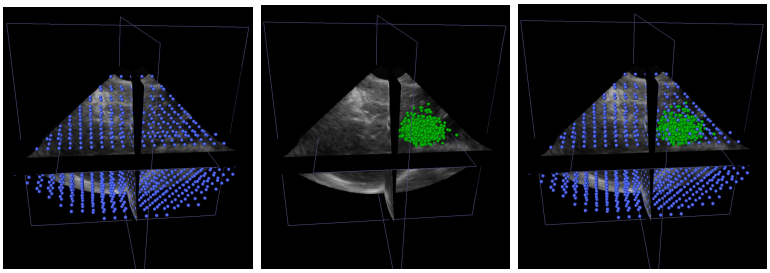


Fig. 2. Global and local point set: Left - Global point set generated using a grid structure (blue color), Middle - Local point set (green color), Right - Global cum local point set generated using a Gaussian distribution (σ=10 mm).

Figure 1. The RTRT approach additionally performs efficient tracking of points in the temporal domain. The tracking starts with a set of ℓ points (generated using a grid structure or a Gaussian distribution, see the global point set and the local point set in Figure 2) located in the fixed/reference image. Points that are consistently tracked are retained and the rest of the points are rejected. Additional points are introduced from a distribution (could be the same distribution as used earlier) if the number of points for tracking is less than ℓ. For details refer [2]. Note that in the next cycle of the RTRT approach, the current estimated transformation $T_{(ref,n)}$ is used in determining the transformation between the reference volume (t_{ref}) and the next US volume (t_{n+1}), written as $T_{(ref,n+1)}$.

2.3 Tracking landmarks

The anatomical landmark tracking approach, see Figure 1, consists of the following two rigid registration steps. First, in the *global* 4D registration/tracking step, the RTRT strategy is used track the whole (liver) US volume ($T_{(ref,n)}$). Second, in the *local* 3D registration step, we refine the tracking result by performing registration using the neighborhood region close to the anatomical landmark ($T'_{(ref,n)}$).

Both the RTR and RTRT strategies use block-matching followed by an outlier rejection scheme to find correspondences between the US volumes. Input to the block-matching module is a point set. The portion/region of the image used for the registration/tracking is determined by the locations of the points in the US volume. As shown in the block diagram in Figure 1, a combination of a global and a local point set is used to perform a global 4D tracking/registration and only a local point set is used to perform a local 3D registration. The global point set is generated using a grid structure spread over the entire US volume, the local point set is a collection of points in the neighborhood of the anatomical landmark (see Figure 2).

3 Experiment and results

The CLUST 2015 challenge dataset is used to evaluate the performance of the method. The challenge contained 16 4D sequences from multiple sources. The

Table 1. Summary of the data

Source	Traning Sequences	Test Sequences	Image size [voxels]	Image res. [mm]	Frame rate [Hz]	Scanner	Probe
EMC	3	2	192x246x117	1.14x0.59x1.19	6	Philips iU22	X6-1
ICR	1	1	480x120x120	0.31x0.51x0.67	24	Siemens SC2000	4Z1c
SMT	4	5	227x227x229	0.70x0.70x0.70	8	GE E9	4V-D

summary of the data is shown in Table 1. The data was divided into a training set of 8 4D sequences and a test set of 8 4D sequences. For tuning the algorithm, annotations (i.e. landmarks) across multiple frames per 4D sequence were provided for the training set. For the test set, one or more annotations in the first frame were provided. These annotated landmarks were tracked over time. The tracking performance of the test set was evaluated by the organizers of the challenge. The Euclidean distance between the tracked points and manual annotations was calculated. The error was summarized by the following statistics: mean, standard deviation, 95 percentile, minimum and maximum distances.

MeVisLab, OpenCL and C++ are used for software development. The OpenCL code was run on a NVIDIA GTX 780 Ti GPU.

Parameter setting : We used a block-size of 11^3 mm^3 for the block-matching. The block is evenly sampled 18x18x18 times. The similarity metric used is normalized cross correlation (NCC), $(\sigma_A, \lambda, \sigma_B) = (0.1, 0.1, 0.1)$ is used as the outlier rejection parameters. These values were optimized in the previous work [3]. The number of points for the block-matching (step one and step two) of the RTRT approach and the RTR approach are set to 100, 200 and 400 points, respectively. The search range for the block-matching (step one and step two) of the RTRT approach and the RTR approach is set to 40^3 mm^3, 10^3 mm^3 and 20^3 mm^3, respectively. The search range (step one and step two) of the RTRT approach and the RTR approach are evenly sampled 60x60x60 times, 15x15x15 times and 30x30x30 times, respectively. The sampling determines the step size for the block-matching. A local point set of 1000 points is generated using a Gaussian distribution with mean located at the anatomical landmark and standard deviation of 10 mm. The adjacent horizontal/vertical nodes of the grid structure used to generate the global point set are 10 mm apart, see Figure 2. The points required for the block-matching in the RTR and the RTRT approaches are sampled from the global and the local point sets.

The training set and the test set results are presented in Table 2 and Table 3, respectively. The mean tracking error for the training set and the test set are 3.26 ± 2.62 mm and 1.62 ± 0.94 mm, respectively. The average run time of the Step 1 (RTRT) and the Step 2 (RTR) of our approach as shown in Figure 1 are 6.68 seconds and 4.18 seconds, respectively. Hence for the given parameter settings the GPU implementation runs at 11 seconds per frame.

Table 2. Training set results

Landmarks	Mean (in mm)	Std (in mm)	95th% (in mm)	Min (in mm)	Max (in mm)
EMC-01_1	0.94	0.51	1.65	0.36	1.79
EMC-02_1	1.19	0.47	1.83	0.80	2.01
EMC-02_2	2.28	1.10	3.62	1.02	3.80
EMC-02_3	2.05	0.58	2.80	1.48	2.96
EMC-02_4	1.80	0.54	2.39	1.14	2.41
EMC-03_1	5.55	2.28	8.20	1.78	8.54
EMC	3.01	2.42	7.85	0.36	8.54
ICR-01_1	1.57	0.56	2.36	0.27	2.83
SMT-01_1	2.06	0.41	2.74	1.08	2.97
SMT-01_2	3.46	0.55	4.43	2.42	4.61
SMT-01_3	3.00	0.42	3.75	1.91	3.89
SMT-02_1	1.65	1.60	2.20	0.6	16.49
SMT-02_2	1.92	0.47	2.74	0.91	3.27
SMT-02_3	3.72	0.70	4.79	2.30	5.59
SMT-03_1	2.29	0.72	3.43	1.19	3.62
SMT-03_2	2.09	0.60	3.14	0.68	3.54
SMT-04_1	8.88	3.82	15.04	0.97	15.31
SMT	3.30	2.64	9.20	0.60	16.49
Tracking Results	3.26	2.62	8.55	0.27	16.49

Table 3. Test set results

Landmarks	Mean (in mm)	Std (in mm)	95th% (in mm)	Min (in mm)	Max (in mm)
EMC-04_1	1.10	0.63	2.28	0.26	2.31
EMC-05_1	1.79	0.36	2.16	1.15	2.17
EMC	1.45	0.61	2.18	0.26	2.31
ICR-02_1	1.65	0.37	2.14	0.80	2.15
SMT-05_1	3.39	2.53	10.13	0.90	10.24
SMT-05_2	0.97	0.36	1.58	0.21	1.91
SMT-06_1	1.56	0.37	2.11	0.57	2.49
SMT-06_2	2.01	0.52	2.77	0.99	3.69
SMT-06_3	1.37	0.35	1.95	0.43	2.18
SMT-07_1	1.83	0.42	2.49	1.02	2.92
SMT-07_2	1.79	0.39	2.46	1.04	2.69
SMT-08_1	1.48	0.41	2.26	0.22	2.44
SMT-08_2	1.09	0.29	1.52	0.37	1.77
SMT-08_3	2.10	0.73	3.37	0.87	3.91
SMT-09_1	1.10	0.35	1.70	0.14	1.89
SMT-09_2	0.96	0.38	1.66	0.10	1.87
SMT-09_3	2.25	0.60	3.16	0.18	3.73
SMT	1.63	0.94	2.86	0.10	10.24
Tracking Results	1.62	0.93	2.84	0.10	10.24

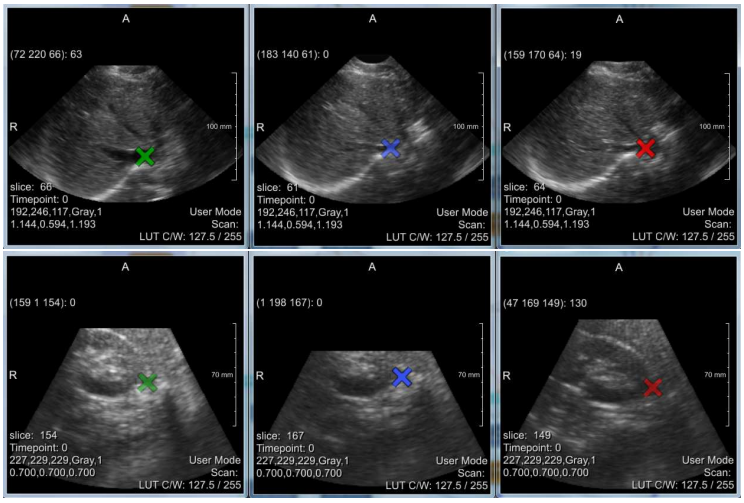


Fig. 3. Training set tracking results: Left - Reference image manual annotation, Middle - Moving image automatic annotation, Right - Moving image manual annotation. Row 1 - For the dataset EMC-03_1 at time point 42, the tracking error is 8.54 mm; Row 2 - For the dataset SMT-04_1 at time point 76, the tracking error is 15.14 mm.

4 Discussion and conclusions

In this paper we perform the task of tracking anatomical landmarks using a combination of previous methods by Banerjee et al. [2] and [3]. A mean tracking error of 1.62 ± 0.94 mm is achieved on the test set. In the first step, the point set used for the global 4D tracking step is a combination of a global point set generated from a grid structure and the local point set generated randomly in the neighborhood of the anatomical landmark. This combination of point set ensures a high percentage of points close to the landmark position during the global 4D tracking step. The local point set is intended to track a specific landmark well, whereas the global point set helps in increasing robustness in tracking. In the second step, the local point set is again used in the local 3D registration step. This step is designed to track the landmark in the presence of local deformations.

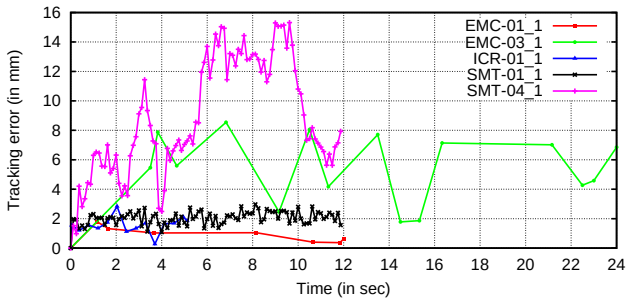


Fig. 4. Tracking examples from the training set.

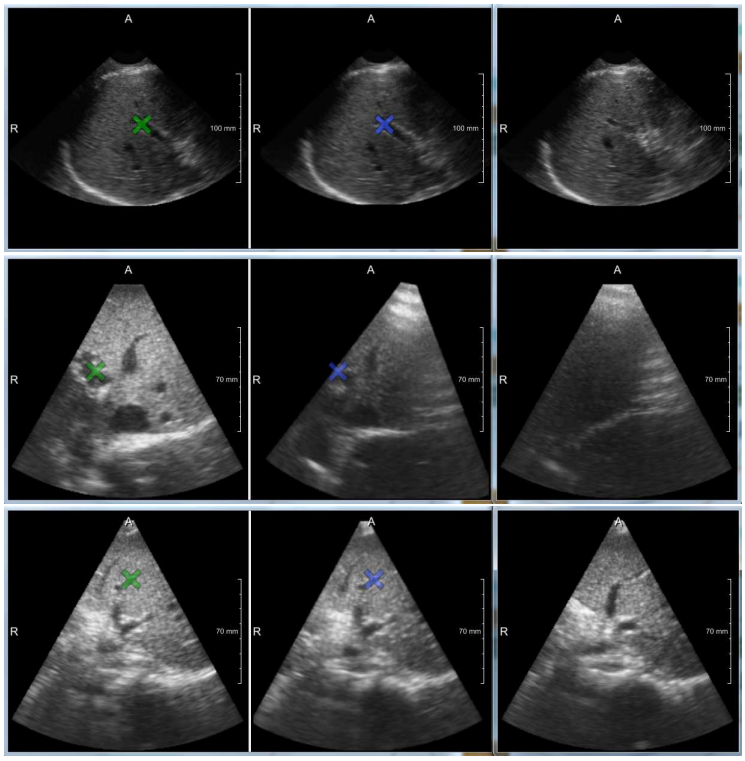


Fig. 5. Test set registered Volumes: Left - Reference image manual annotation, Middle - Registration result automatic annotation, Right - Moving Image. Row 1 - dataset EMC-04_1, time point 124; Row 2 - dataset SMT-05_1, time point 66; Row 3 - dataset SMT-07_2, time point 74.

The mean tracking error for the training set is 3.26 ± 2.62 mm. Two of the datasets (EMC-03_1, SMT-04_1) from the training set have large tracking errors, see Figure 3. In the EMC-03_1 4D US sequence the anatomical landmark is located on a vessel which undergoes large deformations due to blood flow and in the SMT-04_1 4D US sequence the anatomical landmark is located outside the liver. Some of the tracking results from the training set are shown in Figure 4. In the test set the SMT-05_1 4D sequence has large tracking error. In rest of the dataset the tracking performance is satisfactory. Some representative test set registration results are shown in Figure 5.

The speed depends on the number of points, search range size, number of samples in the search range (step size), block size and number of samples in the block. The current approach runs at 11 seconds per frame. For tracking of liver, real-time (faster than image temporal resolution) speed is achieved by Banerjee et al. [2] by selecting appropriate parameters for the US data acquired from Philips iU22 machine with X6-1 probe.

To conclude, we extended our current registration approaches for 3D and 4D US volumes such that it enables tracking of anatomical landmarks in 4D US

sequences. The method is evaluated using CLUST 2015 challenge datasets. For a test set of eight 4D US sequences, an accuracy of 1.62 ± 0.94 mm is achieved.

References

1. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Mach. Intell. 9(5), 698–700 (1987)
2. Banerjee, J., Klink, C., Niessen, W.J., Moelker, A., van Walsum, T.: 4D ultrasound tracking of liver and its verification for TIPS guidance. IEEE Transactions on Medical Imaging (In press)
3. Banerjee, J., Klink, C., Peters, E.D., Niessen, W.J., Moelker, A., van Walsum, T.: Fast and robust 3D ultrasound registration - block and game theoretic matching. Medical Image Analysis 20(1), 173–183 (2015), http://dx.doi.org/10.1016/j.media.2014.11.004
4. Bell, M.A.L., Byram, B.C., Harris, E.J., Evans, P.M., Bamber, J.C.: In vivo liver tracking with a high volume rate 4D ultrasound scanner and a 2D matrix array probe. Physics in Medicine and Biology 57(5), 1359 (2012), http://stacks.iop.org/0031-9155/57/i=5/a=1359
5. Cifor, A., Risser, L., Chung, D., Anderson, E., Schnabel, J.: Hybrid feature-based diffeomorphic registration for tumor tracking in 2-D liver ultrasound images. Medical Imaging, IEEE Transactions on 32(9), 1647–1656 (2013)
6. Lediju, M., Byram, B., Harris, E., Evans, P., Bamber, J.: 3D liver tracking using a matrix array: Implications for ultrasonic guidance of IMRT. In: Ultrasonics Symposium (IUS), 2010 IEEE. pp. 1628–1631 (Oct 2010)
7. Luca, V.D., Benz, T., Kondo, S., König, L., Lübke, D., Rothlübbers, S., Somphone, O., Allaire, S., Bell, M.A.L., Chung, D.Y.F., Cifor, A., Grozea, C., Günther, M., Jenne, J., Kipshagen, T., Kowarschik, M., Navab, N., Rühaak, J., Schwaab, J., Tanner, C.: The 2014 liver ultrasound tracking benchmark. Physics in Medicine and Biology 60(14), 5571 (2015), http://stacks.iop.org/0031-9155/60/i=14/a=5571
8. De Luca, V., Tschannen, M., Székely, G., Tanner, C.: A learning-based approach for fast and robust vessel tracking in long ultrasound sequences. In: MICCAI (1). pp. 518–525 (2013)
9. Øye, O.K., Wein, W., Ulvang, D.M., Matre, K., Viola, I.: Real time image-based tracking of 4D ultrasound data. In: MICCAI (1). pp. 447–454 (2012)
10. Preiswerk, F., Luca, V.D., Arnold, P., Celicanin, Z., Petrusca, L., Tanner, C., Bieri, O., Salomir, R., Cattin, P.C.: Model-guided respiratory organ motion prediction of the liver from 2D ultrasound. Medical Image Analysis 18(5), 740–751 (2014), http://dx.doi.org/10.1016/j.media.2014.03.006
11. Schneider, R.J., Perrin, D.P., Vasilyev, N.V., Marx, G.R., del Nido, P.J., Howe, R.D.: Real-time image-based rigid registration of three-dimensional ultrasound. Medical Image Analysis 16(2), 402–414 (2012)
12. Vijayan, S., Klein, S., Hofstad, E.F., Lindseth, F., Ystgaard, B., Langø, T.: Motion tracking in the liver: Validation of a method based on 4D ultrasound using a nonrigid registration technique. Medical Physics 41(8), 082903 (2014), http://scitation.aip.org/content/aapm/journal/medphys/41/8/10.1118/1.4890091

Data Description

The Challenge on Liver Ultrasound Tracking (CLUST) would not have been possible without images and annotations. This section provides an overview of the data, the contributors and the associated references.

Tables 1-2 list the details for each sequence. The data, which was released for training and test purposes prior to the MICCAI event, is divided into two categories, namely 2D sequences with annotation of point-landmarks (see Table 1) and 3D sequences with point-landmarks (see Table 2). Additional sequences, not included in the Tables are released during the on-site challenge at the MICCAI event.

Data Contributors

Seven groups provided data and generally also the corresponding annotations for CLUST 2015. These groups and their related publications are listed below, following the order of appearance in Tables 1-2.

CIL	-	Biomedical Imaging Research Lab, CREATIS, INSA Lyon, France
ETH	[2, 5]	Computer Vision Laboratory, ETH Zurich, Switzerland
MED	-	mediri GmbH, Heidelberg, Germany
EMC	[1]	Biomedical Imaging Group, Departments of Radiology and Medical Informatics, Erasmus MC, Rotterdam, The Netherlands
ICR	[3, 4]	Joint Department of Physics, Institute of Cancer Research & Royal Marsden NHS Foundation Trust, London and Sutton, UK
SMT	[6]	SINTEF Medical Technology, Image Guided Therapy, Trondheim, Norway

Table 1: Summary of the challenge data for 2D sequences with annotation of point-landmarks. The test set is listed in **black** font. The training sequences, for which all available annotations were provided, are highlighted in **red**. The on-site set is not included in this table.

	Sequence info					Acquisition info		
Sequence	**Im.size [pix]**	**Im.res. [mm]**	**No. frames**	**Im.rate [Hz]**	**Annotation No.**	**Scanner**	**Probe**	**Freq. [MHz]**
CIL-01	480x640	0.30	1342	22	2	Ultrasonix MDP	4DC7-3/40	4.5
CIL-02	480x640	0.40	1075	17	1	Ultrasonix MDP	4DC7-3/40	4.5
CIL-03	480x640	0.40	1070	18	2	Ultrasonix MDP	4DC7-3/40	4.5
CIL-04	480x640	0.50	895	15	2	Ultrasonix MDP	4DC7-3/40	4.5
ETH-01-1	490x570	0.40	3652	15	2	Siemens Antares	CH4-1	2.22
ETH-01-2	482x608	0.41	4650	15	2	Siemens Antares	CH4-1	2.22
ETH-02-1	472x565	0.42	2620	15	1	Siemens Antares	CH4-1	2.22
ETH-02-2	462x590	0.41	4826	15	1	Siemens Antares	CH4-1	2.22
ETH-03-1	473x437	0.28	4588	14	1	Siemens Antares	CH4-1	2.22
ETH-03-2	464x442	0.28	4191	13	1	Siemens Antares	CH4-1	2.22
ETH-04-1	469x523	0.40	5247	16	2	Siemens Antares	CH4-1	1.82
ETH-04-2	480x652	0.38	4510	14	2	Siemens Antares	CH4-1	1.82
ETH-05-1	462x563	0.42	4615	15	2	Siemens Antares	CH4-1	1.82
ETH-05-2	477x556	0.40	3829	13	2	Siemens Antares	CH4-1	1.82
ETH-06-1	462x580	0.40	5244	16	1	Siemens Antares	CH4-1	2.00
ETH-06-2	476x604	0.38	5165	16	1	Siemens Antares	CH4-1	2.00
ETH-07-1	475x548	0.37	5586	17	2	Siemens Antares	CH4-1	1.82
ETH-07-2	467x568	0.37	5582	17	2	Siemens Antares	CH4-1	1.82
ETH-08-1	466x562	0.36	5574	17	2	Siemens Antares	CH4-1	1.82
ETH-08-2	466x589	0.36	5577	17	2	Siemens Antares	CH4-1	1.82
ETH-09-1	464x560	0.40	4587	15	4	Siemens Antares	CH4-1	1.82
ETH-09-2	479x566	0.42	4590	15	3	Siemens Antares	CH4-1	1.82
ETH-10-1	462x589	0.36	5578	17	3	Siemens Antares	CH4-1	1.82
ETH-10-2	470x595	0.36	5584	17	3	Siemens Antares	CH4-1	1.82
ICR-01	393x457	0.55x0.42	4858	23	3	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
ICR-02	393x457	0.55x0.42	3481	23	2	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
ICR-03	393x457	0.55x0.42	3481	23	3	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
ICR-04	393x457	0.55x0.42	3481	23	4	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
ICR-05	397x485	0.55x0.43	3481	20	2	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
ICR-06	397x485	0.55x0.43	3481	21	2	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
ICR-07	397x495	0.49x0.38	3481	23	2	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
ICR-08	399x495	0.50x0.39	3481	23	3	Elekta Clarity - Ultrasonix	m4DC7-3/40	4.5
MED-01-1	408x512	0.41	2455	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-02-1	408x512	0.41	2458	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-02-2	408x512	0.41	2443	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-02-3	408x512	0.41	2436	20	5	DiPhAs Fraunhofer	VermonCLA	5.5
MED-03-1	408x512	0.41	2442	20	2	DiPhAs Fraunhofer	VermonCLA	5.5
MED-03-2	408x512	0.41	2450	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-04-1	524x591	0.35	3304	11	1	Zonare z.one	C4-1	4.0
MED-05-1	524x591	0.35	3304	11	2	Zonare z.one	C4-1	4.0
MED-06-1	408x512	0.41	2427	20	4	DiPhAs Fraunhofer	VermonCLA	5.5
MED-06-2	408x512	0.41	2424	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-07-1	408x512	0.41	2470	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-07-2	408x512	0.41	2478	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-07-3	408x512	0.41	2450	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-07-4	408x512	0.41	2456	20	4	DiPhAs Fraunhofer	VermonCLA	5.5
MED-08-1	524x591	0.35	3304	11	3	Zonare z.one	C4-1	4.0
MED-08-2	524x591	0.35	3304	11	3	Zonare z.one	C4-1	4.0

Table 2: Summary of the challenge data for 3D sequences with annotations of point-landmarks. The test set is listed in **black** font. The training sequences, for which all available annotations were provided, are highlighted in **red**. The on-site set is not included in this table.

	Sequence info					Acquisition info		
Sequence	**Im.size [pix]**	**Im.res. [mm]**	**No. frames**	**Im.rate [Hz]**	**Annotation No.**	**Scanner**	**Probe**	**Freq. [MHz]**
EMC-01	192x246x117	1.14x0.59x1.19	79	6	1	Philips iU22	X6-1	3.2
EMC-02	192x246x117	1.14x0.59x1.19	54	6	4	Philips iU22	X6-1	3.2
EMC-03	192x246x117	1.14x0.59x1.19	159	6	1	Philips iU22	X6-1	3.2
EMC-04	192x246x117	1.14x0.59x1.19	140	6	1	Philips iU22	X6-1	3.2
EMC-05	192x246x117	1.14x0.59x1.19	147	6	1	Philips iU22	X6-1	3.2
ICR-01	480x120x120	0.31x0.51x0.67	141	24	1	Siemens SC2000	4Z1c	2.8
ICR-02	480x120x120	0.31x0.51x0.67	141	24	1	Siemens SC2000	4Z1c	2.8
SMT-01	227x227x229	0.70	97	8	3	GE E9	4V-D	2.5
SMT-02	227x227x229	0.70	96	8	3	GE E9	4V-D	2.5
SMT-03	227x227x229	0.70	96	8	2	GE E9	4V-D	2.5
SMT-04	227x227x229	0.70	97	8	1	GE E9	4V-D	2.5
SMT-05	227x227x229	0.70	96	8	2	GE E9	4V-D	2.5
SMT-06	227x227x229	0.70	97	8	3	GE E9	4V-D	2.5
SMT-07	227x227x229	0.70	97	8	2	GE E9	4V-D	2.5
SMT-08	227x227x229	0.70	97	8	3	GE E9	4V-D	2.5
SMT-09	227x227x229	0.70	97	8	3	GE E9	4V-D	2.5

Bibliography

[1] Banerjee, J., Klink, C., Peters, E.D., Niessen, W., Moelker, A., van Walsum, T.: 4d liver ultrasound registration. In: 6th International Workshop on Biomedical Image Registration, p. 194. Springer (2014)

[2] De Luca, V., Tschannen, M., Szekely, G., Tanner, C.: A Learning-Based Approach for Fast and Robust Vessel Tracking in Long Ultrasound Sequences. In: Med Image Comput Comput Assist Interv, LNCS, vol. 8149, p. 518. Springer (2013)

[3] Lediju, M., Byram, B., Harris, E., Evans, P., Bamber, J.: 3D Liver tracking using a matrix array: Implications for ultrasonic guidance of IMRT. In: IEEE Ultrason Symp. p. 1628 (2010)

[4] Lediju Bell, M., Byram, B., Harris, E., Evans, P., Bamber, J.: In vivo liver tracking with a high volume rate 4D ultrasound scanner and a 2D matrix array probe. Phys Med Biol 57(5), 1359 (2012)

[5] Preiswerk, F., De Luca, V., Arnold, P., Celicanin, Z., Petrusca, L., Tanner, C., Salomir, R., Cattin, P.: Model-Guided Respiratory Organ Motion Prediction of the Liver from 2D Ultrasound. Med Image Anal 18(5), 740 (2014)

[6] Vijayan, S., Klein, S., Hofstad, E., Lindseth, F., Ystgaard, B., Lango, T.: Validation of a non-rigid registration method for motion compensation in 4D ultrasound of the liver. In: IEEE Int Symp Biomed Imaging. p. 792 (2013)