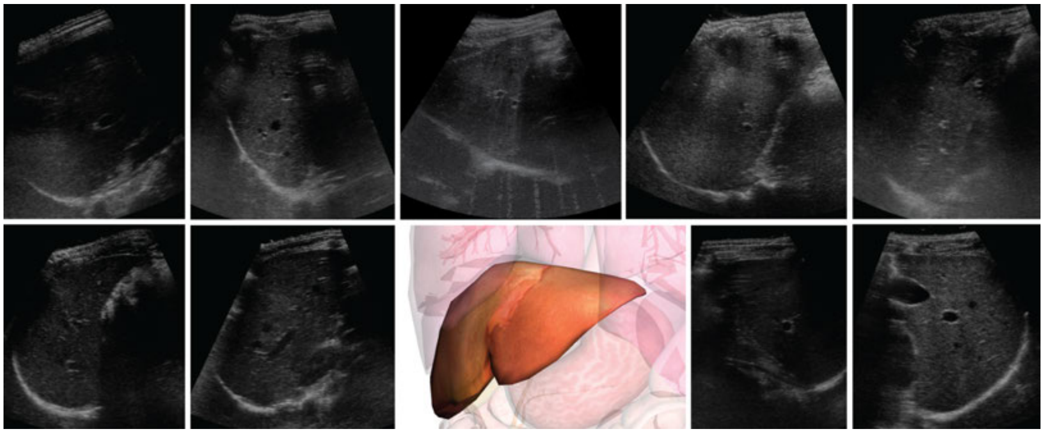
Proceedings of the MICCAI 2014 workshop

Challenge on Liver Ultrasound Tracking

CLUST 2014

Held in Conjunction with MICCAI 2014
Cambridge, MA, USA, September 14, 2014



Preface

Ultrasound (US) imaging is a widely used medical imaging technique. As US has high temporal resolution and is non-invasive, it is an appealing choice for applications which require tracking and tissue motion analysis, such as motion compensation in image-guided intervention and therapy. Specifically, we want to address the issue of respiratory motion in the liver.

While there is a large number of relevant works in motion tracking and tracking of US liver images, it is hard to compare the reported tracking strategies. Open datasets for designing and testing tracking algorithms are missing, and private datasets differ in size, image dimension and sequence length. Critical are also the variation in tracking objective (full organ, anatomical landmarks, tumor) and validation strategies.

The aim of the Challenge on Liver Ultrasound Tracking (CLUST) was to present the current state-of-the-art in automated tracking of anatomical landmarks in the liver and compare between different methods.

We distributed a dataset of 54 sequences of patients and volunteers under free breathing, provided by 6 groups (see pages 61-63). 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 three parts, according to the image dimension and annotation type. The first part is composed of 28 2D sequences from healthy volunteers with point-landmark annotations. The second part contains 10 2D sequences from 5 patients with segmentation annotations. The third part consists of 16 3D sequences with point-landmark annotations from healthy volunteers. The data were anonymized and in the format of sequences of images (.png, .jpeg, NIFTI) or 3D images (.mha). The data were split into a training and a test set. Training data (10% of the sequences) and part of the test data (70%) were available prior to the challenge. 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 was distributed shortly before the MICCAI conference. This helped the organizers and participants to comment on algorithm run-time, parametrization and tuning, flexibility, and feasibility for a real application scenario. The results for this last dataset were not included in this proceedings book, as they were generated after the paper submission deadline.

In response to the call for papers, we had 55 requests for access to the data. A total of 7 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. Workshop attendees were able to present their research and exchange ideas, learn the current state-of-the-art techniques, and gained 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; 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.

September 2014

Valeria De Luca
Amalia Cifor
Muyinatu A. Lediju Bell
Christine Tanner

Workshop Organization

Organizing Committee

Valeria De Luca	Computer Vision Laboratory, ETH Zurich, Switzerland
Amalia Cifor	Institute of Biomedical Engineering, University of Oxford, UK
Muyinatu A. Lediju Bell	Laboratory for Computational Sensing and Robotics, Johns Hopkins University, USA
Christine Tanner	Computer Vision Laboratory, ETH Zurich, Switzerland

Website

`http://clust14.ethz.ch/`

Contents

2D Liver Ultrasound Tracking **5**

2D and 3D Liver Ultrasound Tracking **37**

Liver Feature Motion Estimation in Long High Frame Rate 2D Ultrasound Sequences

Tuathan O'Shea[1], Jeff Bamber[1] and Emma Harris[1]

[1] Joint Department of Physics, Institute of Cancer Research & Royal Marsden NHS Foundation Trust, London and Sutton, UK
tuathan.oshea@icr.ac.uk

Abstract. This study investigates the use a 2D normalized cross-correlation (NCC)-based algorithm to estimate *in vivo* motion of liver features in 2D B-mode ultrasound (US) images. Datasets included 23 volunteer imaging sequences, each containing first frame annotated points of interest (POI). Images had a range of spatial (0.28 – 0.71 mm) and temporal (11 – 25 Hz) resolution. Image quality was also highly variable. A 2D block-matching algorithm was developed to track POI motion throughout the imaging sequence. A correlation and displacement thresholding tracking approach, which used knowledge of previous displacement and (1) linear extrapolation, (2) a regularizing sinusoidal breathing model or (3) hybrid fixed-reference / incremental tracking was use to account for potential tracking errors. The overall mean error in vessel tracking was 2.15 ± 2.7 mm. This approach to motion estimation shows promise for applications such as radiation therapy tumor tracking.

1 Introduction

This study investigates the estimation of liver feature motion in variable quality volunteer 2D ultrasound amplitude demodulated data. In conformal radiation therapy, some form of (intra-fraction) motion management is often required [1]. If motion cannot be minimized using a method such as respiratory gating [2], then this motion should be tracked in as close to real-time as possible. Tracking cardiac and respiratory induced motion requires an imaging method which samples the target position with an adequate temporal resolution. In radiation therapy, most current tracking systems are based on kV x-ray imaging [3], [4]. Ultrasound has two major advantages over these methods: (i) it does not impart ionizing radiation (imaging dose) and (ii) it allows the visualization of soft tissue. Correlation-based techniques have been used to investigate ultrasound speckle and feature-based motion tracking. Ultrasound speckle tracking of respiratory induced phantom motion and *in vivo* feature-based tracking has been studied [5]. Good agreement (mean absolute difference < 2 mm) was found between tracked and manually annotated displacements using a mechanically swept 3D probe limited to a 0.5 Hz imaging rate. Lediju Bell et al. [6] used a 2D matrix array transducer to acquire *in vivo* liver motion data from three volunteers at imaging rates of up to 48 Hz. In the study, volumetric data was acquired at high imaging rates without the restriction of a mechanically swept ultrasound transducer. It was found that volume rates of 8 to 12 Hz were required to track cardiac and respiratory induced liver motion. In many instances out-of-plane motion is small and 2D imaging is a valid approach to tissue motion estimation. De Luca et al. [7] presented a scale adaptive block-matching approach to liver vessel tracking in long 2D ultrasound sequences. The method achieved a mean tracking accuracy of < 1 mm.

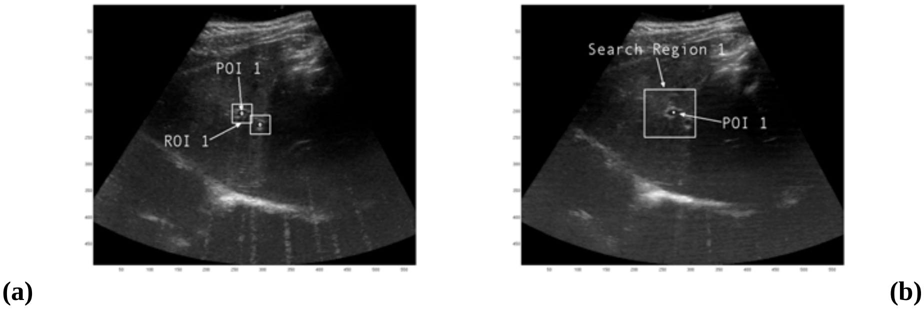


Fig. 1. Ultrasound B-mode data for one of the volunteers to illustrate image quality and method employed to track liver features. The first frame and annotated points of interest (POI) are shown (a). A region of interest (ROI) is defined around each POI. A correlation-based block matching algorithm was used to locate this same POI, within a larger search region, in a subsequent ultrasound frame (b).

In the current study we employ a 2D correlation-based block-matching algorithm to track features (blood vessels center-of-mass) in 2D B-mode ultrasound image sequences (from 23 volunteers). The tracking code was applied to ultrasound data from three different scanners / transducers with a range of image resolutions. We investigated non-incremental (fixed reference) tracking. For non-incremental tracking, the mean inter-frame displacement is greater than for incremental tracking and there is a higher probability of tissue deformation and rotation [4], however, incremental tracking can be prone to drift error accumulation. In speckle tracking, it is known that tissue deformation and rotation corrupt the speckle pattern [8]. Low imaging rates have limited speckle-tracking accuracy to the extent that adequate *in vivo* motion estimation has been obtained by tracking features (blood vessels) only [5]. In the current study, data was acquired at high frame rates (11 – 25 Hz) such that inter-frame rotation and deformation is expected to be small and, additionally, we tracked tissue features which were generally highly visible throughout the imaging sequence. Nevertheless, in 2D images out-of-plane motion can be an issue. The tracking performance of our 2D correlation-based automated tracking code was quantified by comparison with manual annotations of the tissue features throughout the ultrasound sequence.

2 Materials and method

2.1 Ultrasound data

B-mode ultrasound data was provided by the CLUST 2014 ("*MICCAI Challenge on Liver Ultrasound Tracking*") [9]. 2D volunteer liver image data from 23 patients (MED and ETH datasets) was acquired by one of three ultrasound systems (Siemens Antares, DiPhAs Fraunhofer and Zonare z.one). Data had varying spatial (0.28 – 0.71 mm) and temporal resolution (11 – 25 Hz) and sequences lasted from 121.2 – 580.64 s. Examples of the volunteer image data from the Siemens Antares are shown (Fig. 1). Some of the B-mode data contained what appeared to be electronic interference (Fig. 1 (a)) and large shadowing artifacts (Fig. 1 (a) and (b), from ETH-05 dataset). Two of the 23 volunteer datasets (MED-04 and ETH-05) were provided with ground truth annotations (of liver blood vessels) throughout the acquisition sequence which were used to assess tracking code performance and enable code development. Annotations were provided in the following form: frame number, x-pixel (lateral position) and y-pixel (axial position). For the remaining datasets, liver features (blood vessels centers) were annotated in the first frame only. The number of annotations per image sequence ranged from one to five liver features (cf. Table. 1).

2.2 Tracking code

To detect the motion of liver blood vessels center-of-mass an automated (serial) tracking code was developed in MATLAB R2011b (MathWorks, Inc. MA, USA). The code was based on the use of normalized cross-correlation (NCC) as a similarity metric between the current and a reference ultrasound frame. A reference region of interest (*ROI*) was defined around each annotation or point-of-interest (*POI*) in the first ultrasound frame. In subsequent ultrasound frames a larger search region was defined. The position of maximum correlation from the NCC code was used to identify the new position of the the *ROI*. To improve the precision of coarse pixel displacement estimates, the sub-pixel (fine) displacement was calculated by fitting the maximum correlation and two surrounding values in the correlation matrix with a second order function and finding the peak (i.e. when the slope, *m* == 0). The tracking code output tracking results in the format: frame number, x-subpixel (lateral position) and y-subpixel (axial position).

To track annotated POI motion three tracking methods were developed and each was used to track features in a subset of the US sequences: (1) a simple correlation and displacement thresholding (fixed-reference) tracking approach, which used knowledge of previous displacement and linear extrapolation [10], (2) a regularized model-based tracking code using a sinusoidal breathing model (which was applied to two of the US sequences to investigate improvements in tracking results) and (3) a hybrid fixed reference / incremental (updated ROI) tracking approach (further details below). We visually assessed which US sequences were best suited to which method by plotting (overlaying) the raw displacement tracking code output (vessel center-of-mass) on the current ultrasound frame in "real-time". In this way, tracking errors due to, for example false matches within the search area, became obvious. The tracking method which gave the best (visually assessed) results for a particular US sequence was then selected.

For method (1), above, the code monitored the inter-frame displacement (*mdisp*) and correlation (*mcorr*) (via user-specified thresholds) and limited the maximum displacement. In cases when the inter-frame displacement was larger (and *mcorr* smaller) than the threshold values, the current displacement estimate was replaced with displacement predicted by linear extrapolation using the previous two displacement estimates.

For method (2), a model-based (predictive) regularization scheme was developed and used to track feature motion in two of the volunteer image sequences (5 and 14). After a user-specified period of time (number of frames), *t*, the tracking code fit the previous *t* seconds of raw motion estimation data (median filtered, n = 3) with a well known respiratory motion model [11] and this was used to infer the current displacement of the feature (*ROI*). During time periods which exhibited potential tracking errors (as monitored by *mcorr* and *mdisp*), the model-predicted displacement could be used to infer the new position of the *ROI*.

For US sequences 18 – 23, out-of-plane motion, rotation and deformation changed the images to such an extent that standard fixed-reference tracking was not feasible. Instead the code monitored the correlation (*mcorr*) and displacement (*mdisp*) value and used linear extrapolation to calculate displacement and update the reference region (ROI) if the values were below the user-defined thresholds (method (3)).

2.3 Analysis

The tracking code was developed and used to track motion of first frame annotated features (POI) in twenty-three volunteer image sequences. Automated tracking results were evaluated by comparison with manual annotations of liver feature (vessels) throughout each image sequence which were provided after automated tracking was complete. Tracking accuracy was evaluated using the Euclidean distance between tracked points and manually annotated points which was summarized by the mean and standard deviation. The run-time performance of the tracking code was also evaluated by calculating the average run-time for all cases.

Table 1. Volunteer B-mode data sequence spatial and temporal resolution, number of points-of-interest (*POI*) and error (mean ± standard deviation) in tracking code motion estimation (as quantified relative to manual annotations of liver feature motion). Listed for POI with maximum (mean ± standard deviation) error for that specific volunteer data-set only (POI listed in brackets). Patients with annotations available throughout the imaging sequence are highlighted in bold

Sequence number	Name	Im. Res. [mm]	Imaging rate [Hz]	No. of POI	Tracking method	Track. Error mean ± SD [mm]
1	ETH-01	0.71	25	1	(i)	1.9 ± 0.4 (1)
2	ETH-02	0.40	16	1	(i)	0.5 ± 0.2 (1)
3	ETH-03	0.36	17	3	(i)	1.6 ± 1.0 (1)
4	ETH-04	0.42	15	1	(i)	0.9 ± 1.0 (1)
5	**ETH-05**	**0.40**	**15**	**2**	**(ii)**	**1.1 ± 1.1 (1)**
6	ETH-06	0.37	17	2	(i)	0.6 ± 0.3 (2)
7	ETH-07	0.28	14	1	(i)	0.7 ± 0.3 (2)
8	ETH-08	0.36	17	2	(i)	0.9 ± 0.4 (2)
9	ETH-09	0.40	16	2	(i)	0.8 ± 0.6 (2)
10	ETH-10	0.40	15	4	(i)	1.2 ± 1.5 (3)
11	MED-01	0.41	20	3	(i)	1.8 ± 0.6 (3)
12	MED-02	0.41	20	3	(i)	1.8 ± 1.8 (2)
13	MED-03	0.41	20	4	(i)	2.3 ± 1.3 (2)
14	**MED-04**	**0.41**	**20**	**3**	**(ii)**	**3.3 ± 1.7 (3)**
15	MED-05	0.41	20	3	(i)	2.3 ± 1.3 (2)
16	MED-06	0.41	20	3	(i)	6.3 ± 7.5 (3)
17	MED-07	0.41	20	3	(i)	5.3 ± 4.2 (1)
18	MED-08	0.41	20	2	(iii)	4.9 ± 3.5 (2)
19	MED-09	0.41	20	5	(iii)	11.7 ± 5.6 (5)
20	MED-10	0.41	20	4	(iii)	6.6 ± 3.9 (1)
21	MED-13	0.35	11	3	(iii)	4.4 ± 1.4 (3)
22	MED-14	0.35	11	3	(iii)	3.4 ± 2.0 (3)
23	MED-15	0.35	11	1	(iii)	2.4 ± 1.4 (1)

3 Results and discussion

The accuracy with which the automated tracking code could track multiple liver features in the 23 volunteer B-mode imaging sequences is summarized in the final column of table 1 (the values for the POI exhibiting the largest tracking error is listed). The lowest motion estimation error (0.5 ± 0.2 mm) was for an ultrasound sequence containing a relatively large, single centrally located blood vessel. For many of the ultrasound sequences, there was relatively small out-of-plane motion or deformation of the tracked features and therefore a fixed-reference NCC-based approach was adequate. However, on occasions the tracking code detected a false match within the search region, for example the hyperechogenic blood vessel wall would disappear (out-of-plane) leaving only the hypoechogenic vessel centre (blood) and the NCC code would "find" another hyperechogenic feature (i.e. generate a false match) within the search region. When the correlation value (inter-frame displacement) for a POI decreased (increased) below a user-defined threshold (e.g, the current NCC value was < 0.8, inter-frame displacement < 3 mm), linear extrapolation was used to account for the vessel displacement in the time interval. While linear extrapolation may not be the most accurate method [9], it appears adequate in cases when there are no times of sustained tracking errors and at the high frame rates of these data sets.

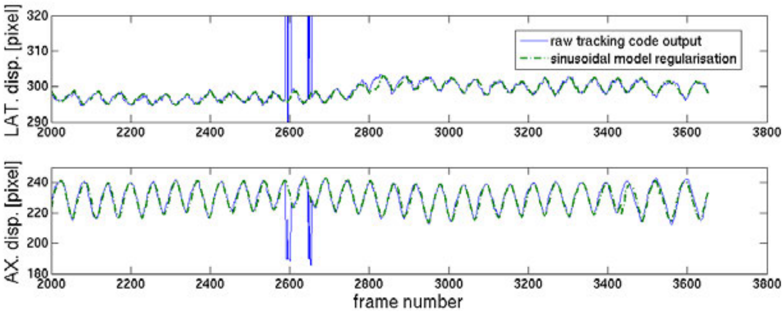


Fig. 2. Example of fixed-reference tracking code raw motion estimation (solid blue line) exhibiting some obvious tracking errors (false matches) and application of model-based regularization to improve motion estimation results (dashed green line)

For data sequences 5 and 14, a fixed reference sinusoid model-based tracking approach was adopted. The model was used to fit the last *t* seconds of ultrasound data to attempt to improve tracking results (Figure 2). We used a value of *t* = 5 s which is the approximate average breathing period of most patients/volunteers. The model was found to work well for relatively regular breathing motion (i.e. sequences 5 and 14) and could detect large tracking errors ("false matches") in drifting breathing signals. However, when the algorithm was applied to US sequences which exhibited large amounts of out-of-plane motion or deformation, it failed to model the liver feature motion.

Imaging sequences 16 – 23 exhibited relatively large out-of-plane motion and vessel deformation and this is reflected by the increased mean tracking error (Table 1). While it has previously been found that NCC-based incremental tracking of liver features in 3D US images was not as accurate as fixed-reference tracking [5], large changes over long imaging sequences meant standard fixed-reference tracking could not be used. Incremental tracking is also known to suffer from drift [5]. A hybrid fixed reference / incremental tracking method which updated the reference ROI if a low correlation value was detected was found to marginally improve results but some large errors remained (e.g. volunteer sequence 9 mean error from POI 5: 11.7 mm). In the future, motion estimation accuracy could benefit from a code which combines model-based regularization and hybrid tracking. The sinusoidal model employed in this work increased run-time by a factor of 3. Other methods, such as linear regression prediction which could be used to update the ROI location and account for displacement during times of known errors, are known to be fast and accurate [10]. Detailed analysis of the optimum use of inter-frame correlation and displacement thresholds to infer tracking errors, would help limit drift accumulation during hybrid/incremental tracking.

The MATLAB tracking code took approx. 190 ms per POI per frame pair (on a single Intel® Core™ 2 Duo E6750 2.66GHz CPU) using a 150 x 150 pixel search region. With model fitting disabled the run-time was approximately 60 ms per POI per frame pair with (*mdisp* and *mcorr* thresholding enabled).

4 Summary and conclusion

Long sequences of B-mode volunteer ultrasound data of variable quality have been used to develop automated tracking algorithms which were used to track annotated liver features in 23 US sequences. The tracking code used correlation and displacement thresholds to identify potential errors and linear extrapolation to account for displacement during these events. For two US sequences, previous motion estimation data were used to generate a breathing sinusoidal model which was used to account for potential future tracking errors. For six of the US sequences, out-of-plane motion, rotation and deformation meant standard fixed reference tracking was not feasible. Instead a hybrid fixed reference/incremental tracking approach was employed. Comparison of manually and automatically tracked displacement of liver features (blood vessels) for sequences (generally longer than a standard external beam radiation therapy delivery) have shown promise (overall mean error 2.15 ± 2.7 mm). Future work will investigate the optimum code parameters, including other regularization models and methods [10], and address the relatively long run-time.

The tracking code model could also be extended to allow future prediction and account for tracking algorithm latencies should this be a significant issue for radiation dose delivery.

References

1. Webb S.: Motion effects in (intensity modulated) radiation therapy: a review. Physics in medicine and biology 51 (2006) (13) R403
2. von Siebenthal M., Szekely G., Lomax A. and Cattin P.: Systematic errors in respiratory gating due to intrafraction deformations of the liver. Medical Physics 34 (9) (2007) 3620-3629
3. Hoogeman M., Prévost J. B., Nuyttens J., Pöll J., Levendag P., and Heijmen B.: Clinical accuracy of the respiratory tumor tracking system of the cyberknife: assessment by analysis of log files. International Journal of Radiation Oncology* Biology* Physics 74 (1) (2009) 297-303
4. Ng J., Booth J., Poulsen P., Fledelius W., Worm E., Eade T., Hegi F., Kneebone A., Kuncic Z., and Keall P.: Kilovoltage intrafraction monitoring for prostate intensity modulated arc therapy: first clinical results. International Journal of Radiation Oncology* Biology* Physics 84 (5) (2012) e655-e661
5. Harris E., Miller N., Bamber J., Symonds-Tayler R. and Evans P.: Speckle tracking in a phantom and feature-based tracking in liver in the presence of respiratory motion using 4D ultrasound. Physics in medicine and biology 55 (12) (2010) 3363
6. Bell M. A. L, Byram B., Harris E., Evans P. and Bamber J.: 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) (2012) 1359
7. De Luca V., Tschannen M., Székely G. and Tanner C.: A Learning-Based Approach for Fast and Robust Vessel Tracking in Long Ultrasound Sequences. Medical Image Computing and Computer-Assisted Intervention–MICCAI (2013) 518-525
8. Meunier J.: Tissue motion assessment from 3D echographic speckle tracking. Physics in medicine and biology 43 (5) (2012) 1241
9. http://clust14.ethz.ch/
10. Sharp G., Jiang S., Shimizu S. and Shirato H.: Prediction of respiratory tumour motion for real-time image-guided radiotherapy . Physics in medicine and biology 49 425-440
11. Lujan A., Larsen E., Balter J. and Ten Haken R.: A method for incorporating organ motion due to breathing into 3D dose calculations. Medical Physics 26 (5) (1999) 715-720

Liver Ultrasound Tracking Using Long-term and Short-term Template Matching

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 template matching and uses multiple templates called long-term template and short-term template. A template to track the target tissue is adaptively selected from the long-term template and the short-term template. The tracking performance is assessed on 21 sequences of 2D ultrasound with 54 regions of interests. Mean tracking error is 1.71 mm. We also confirm that tracking can be performed in about 84 msec per frame using a personal computer.

Keywords: Ultrasound, Liver, Tracking, Template matching, Multiple templates

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. Template matching is widely used because it is relatively simple and gives high performance. Template matching is also used for ultrasound video sequences. One of the important items to design template matching is a method for selecting templates. There are two major methods for selecting templates. One is to use a surrounding area of a tissue specified at the first frame as a template and the template is never updated until the end of the sequence [5]. And the other is to use a tracked region at the most recent frame as a template and the template is updated at every frame [6]. The former has a disadvantage that tracking fails when the shape of the tracking target changes. The latter can overcome the problem of the former, it has a disadvantage that small tracking errors are accumulated. Fig. 1 shows examples of a target vessel in ultrasound images. As can be seen in Fig. 1, Fig. 1(b) is better than Fig. 1(a) to be used as a template when template matching is performed for Fig. 1(c). A method to solve these problems has been proposed in [7] and the method adaptively selects

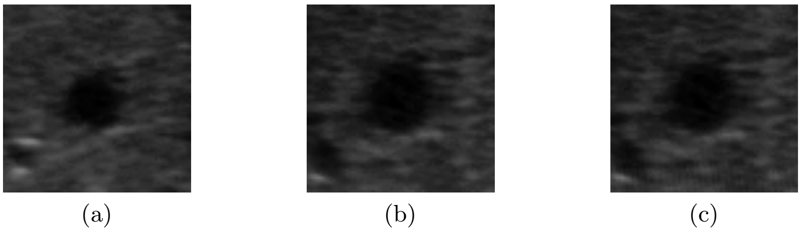


Fig. 1. Examples of ultrasound images in a sequence. These images show the target vessel in different frames. (a) 1st frame. (b) 37th frame. (c) 38th frame.

a template from the first frame or the most recent frame. Moreover, the idea in [7] is applied to ultrasound liver tracking in [8].

While it is possible to perform template matching with high correlation when the template is selected from the most recent frame, it still has a drift problem if it cannot estimate the motion in high accuracy, i.e. sub-pixel order.

In this paper, we propose a tracking method of tissues in long ultrasound sequences of liver. In the proposed method, we pay attention to the following characteristics that liver ultrasound video sequences have: 1) Each region of interest moves to almost same direction, and 2) Motion of each region has a high periodicity. Under these observation, we adopt the following methods in the proposed method. 1) We estimate a global motion for the whole frame and utilize the estimated global motion for estimating motion of each ROI. 2) To avoid drift, templates obtained at the first frame are used preferentially. 3) To track ROI even when texture and shape of the ROI are changed from the first frame, we select additional templates from past neighborhood frames. 4) Search ranges of template matching changes adaptively depending on the motion of the past frames.

Though the proposed method is the same as the methods proposed in [7] and [8] in terms of using a plurality of templates selected from both the first frame and the most recent frame, we propose a method to select a template from a plurality of recent frames by paying attention to the characteristic that the movement of the liver tissue by breathing is periodic.

2 Proposed Method

2.1 Overview

Fig. 2 is an overview of the procedure of our proposed method and Fig. 3 is a schematic diagram of the proposed method. Our proposed method is based on template matching. We use multiple templates called long-term and short-term templates. We will describe the details about each process in the following sections.

- At the first frame
  - Select global and long-term templates (Step 1)
- At each following frame
  - Estimate global motion (Step 2)
  - For each tracker (ROI)
    - Estimate long-term motion (Step 3)
    - Estimate short-term motion (Step 4)
    - Obtain the final tracking result (Step 5)

Fig. 2. Overview of the procedure of the proposed method.

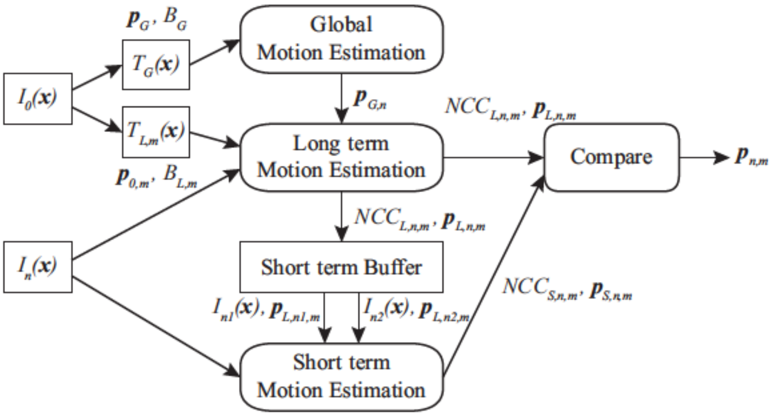

$\boldsymbol{p}_G$, B_G
$T_G(\boldsymbol{x})$
Global Motion Estimation
$I_0(\boldsymbol{x})$
$\boldsymbol{p}_{G,n}$
$NCC_{L,n,m}$, $\boldsymbol{p}_{L,n,m}$
$T_{L,m}(\boldsymbol{x})$
Long term Motion Estimation
Compare
$\boldsymbol{p}_{n,m}$
$\boldsymbol{p}_{0,m}$, $B_{L,m}$
$NCC_{L,n,m}$, $\boldsymbol{p}_{L,n,m}$
$I_n(\boldsymbol{x})$
Short term Buffer
$NCC_{S,n,m}$, $\boldsymbol{p}_{S,n,m}$
$I_{n1}(\boldsymbol{x})$, $\boldsymbol{p}_{L,n1,m}$
$I_{n2}(\boldsymbol{x})$, $\boldsymbol{p}_{L,n2,m}$
Short term Motion Estimation

Fig. 3. A schematic diagram of the proposed method.

2.2 Selection of global and long-term templates (Step 1)

Suppose we have a video sequence of ultrasound images $I_n(\mathbf{x})$ where $\mathbf{x} = (x, y)^T$ are the pixel coordinates and $n = 0, 1, 2, \cdots$ is the frame number. We also have m annotations at pixel positions $\mathbf{p}_{0,m}$ which are the positions of tracking targets at the first frame $I_0(\mathbf{x})$, where $\mathbf{p} = (p_x, p_y)^T$.

We select templates of two types at the first frame $I_0(\mathbf{x})$. One is used to determine a motion of the entire frame and it is referred to as a 'global template' $T_G(\mathbf{x})$. The other is a template used in each tracker (ROI) and it is referred to as a 'long-term template' $T_{L,m}(\mathbf{x})$. In addition to a long-term template, each tracker has short-term templates. We will describe the short-term templates later.

A long-term template is obtained as a square area around an annotation point. Since the size of the target tissue, e.g. vessel, is different for each annotation, we decide the size of the long-term template based on variance of the pixel values inside ROI. We first locate a smallest rectangular ROI, which has $B_{Lmin} \times B_{Lmin}$ pixels, around an annotation. The size of the ROI is gradually increased until the maximum size $B_{Lmax} \times B_{Lmax}$ and we calculate a variance of pixel values in the ROI. When we first find a local minimum of the variance,

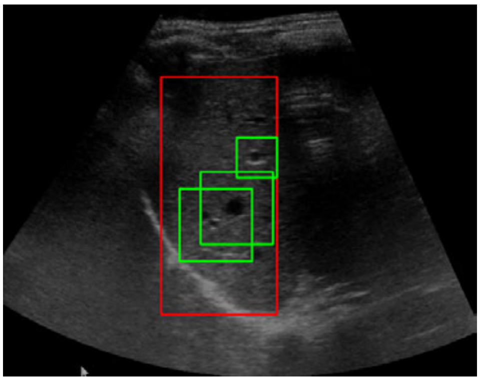


Fig. 4. An example of a set of a global template and long-term templates. The red rectangle shows a global template and the green rectangles show long-term templates.

we stop increasing the size of the ROI and the ROI is the long-term template. We denote a set of pixel coordinates included in the long-term template for m-th annotation as $B_{L,m}$.

A global template is intended to be used to determine a motion of the entire frame. The proposed method extracts a large rectangular area as much as possible in the ultrasound image area by excluding low brightness (shadow) area at the first frame. In the case the long-term template ROIs at the first frame are not included in the global template ROI, we expand the region to include the long-term template ROIs. We denote a set of pixel coordinates included in the global template as B_G.

The global template and the long-term templates are never updated after those are set at the first frame, i.e. all subsequent frames use the same global template $T_G(\mathbf{x})$ and long-term templates $T_{L,m}(\mathbf{x})$, selected at the first frame, for template matching.

Fig. 4 shows an example of a set of a global template and long-term templates. The red rectangle shows a global template and the green rectangles show long-term templates.

2.3 Global motion estimation (Step 2)

Global motion estimation is performed at each of the second and subsequent frames. Template matching is performed using the global template $T_G(\mathbf{x})$. The template matching is evaluated based on normalized cross correlation (NCC). The displacement giving the maximum NCC, which is found by exhausted search, is the tracked position $\mathbf{p}_{G,n}$ by the global motion estimation as Eq. (1),

$$\mathbf{p}_{G,n} = \arg \max_{\mathbf{p}} \frac{\sum_{\mathbf{x}\in B_G} T_G(\mathbf{x}) \cdot I_n(\mathbf{x}+\mathbf{p})}{\sqrt{\sum_{\mathbf{x}\in B_G} T_G(\mathbf{x})^2 \cdot \sum_{\mathbf{x}\in B_G} I_n(\mathbf{x}+\mathbf{p})^2}}. \tag{1}$$

2.4 Long-term motion estimation (Step 3)

Long-term motion estimation is executed for each tracker (ROI). The long-term motion estimation is performed using a long-term template $T_{L,m}$ which is obtained at the first frame as in Eq. (2),

$$\mathbf{p}_{L,n,m} = \arg \max_{\mathbf{p}\in S_L} \frac{\sum_{\mathbf{x}\in B_{L,m}} T_{L,m}(\mathbf{x}) \cdot I_n(\mathbf{x}+\mathbf{p})}{\sqrt{\sum_{\mathbf{x}\in B_{L,m}} T_{L,m}(\mathbf{x})^2 \cdot \sum_{\mathbf{x}\in B_{L,m}} I_n(\mathbf{x}+\mathbf{p})^2}}, \tag{2}$$

where S_L is a set of pixel coordinates in the search area for the long-term motion estimation.

The search range S_L depends on the results of the global motion estimation in Step 2. When the maximum NCC value of the global motion estimation is higher than a threshold $Th_{NCC,G}$, the long-term motion estimation is performed in the vicinity of $\mathbf{p}_{G,n}$. Otherwise, we use a default value of the search range for the motion estimation S_d. The tracked position $\mathbf{p}_{L,n,m}$ and the maximum NCC value $NCC_{L,n,m}$ are stored in the short-term buffer shown in Fig. 3.

2.5 Short-term motion estimation (Step 4)

Short-term motion estimation is performed using short-term templates. Short-term templates are selected from images of the long-term tracked positions until the most recent frame. Here, we use the tracking results with the long-term motion estimation to select short-term templates and the tracking results with the short-term motion estimation are not used at the subsequent frames to avoid drift.

We estimate a cycle of motion from the past tracking results. Fig. 5 shows an example of temporal changes of tracked positions from 300th frame to 500th frame in a ultrasound video sequence. As can be seen in Fig. 5, motions of tissue in liver have periodicity. Two short-term templates for n-th frame are selected from the frames during the recent cycles. Suppose the cycle of the motion is L_c, one short-term template is selected from $(n - L_c)$th frame to $(n - 1)$th frame (period #1 in Fig. 5) and the ROI with the maximum $NCC_{L,n,m}$ is selected as the first short-term template. Another short-term template is selected from $(n - L_c/2 \times 3)$th frame $(n - L_c/2)$th frame (period #2 in Fig. 5) and the ROI with the closest position to the tracked position at $(n - 1)$th frame is selected as the second short-term template.

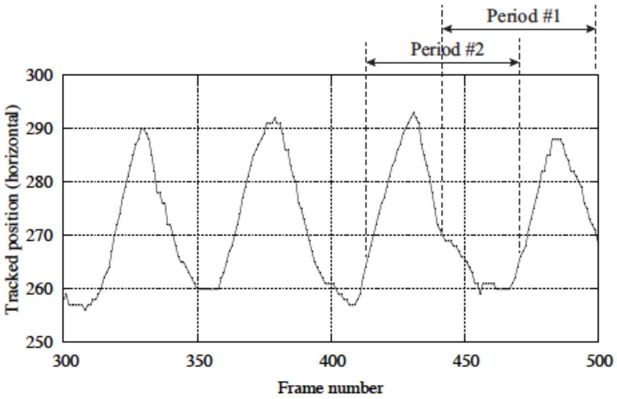


Fig. 5. An example of tracked positions in an ultrasound video sequence.

When we fail to obtain the period of the motion, e.g. in the beginning of the sequence, the short-term motion estimation is performed for a default search range S_d. Otherwise, we apply principal component analysis to the trajectory of the tracking positions. And we perform the motion estimation only for the direction of the eigen vector with the biggest eigen value. The displacement which gives the maximum NCC, $NCC_{S,n,m}$, in the search range is the result of the short-term motion estimation, $\mathbf{p}_{S,n,m}$

2.6 Final tracking result (Step 5)

We compare the maximum NCC value of the long-term motion estimation $NCC_{L,n,m}$ obtained in Step 3 and the maximum NCC value of the short-term motion estimation $NCC_{S,n,m}$ obtained in Step 4. The motion estimation result $\mathbf{p}_{n,m}$ is obtained with Eq. (3),

$$\mathbf{p}_{n,m} = \begin{cases} \mathbf{p}_{L,n,m} \ (NCC_{L,n,m} \geq \alpha \times NCC_{S,n,m}) \\ \mathbf{p}_{S,n,m} \ (\text{otherwise}) \end{cases}, \tag{3}$$

where α is a weight less than 1.0 which prioritizes the long-term motion estimation.

3 Experimental Results

We evaluated the performance of the proposed method using the 2D point-tracking test data. The test data was provided by organizers of CLUST 2014, MICCAI Challenge on Liver Ultrasound Tracking.

In the experiment, we used the following values for parameters: B_{Lmin} = one-tenth of smaller size of image width and height, $B_{Lmax} = 120$ pixels, $Th_{NCC,G} = 0.95$. $S_d = 15$ pixels for both horizontal and vertical directions and $\alpha = 0.95$.

Table 1. 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
ETH	0.79	1.13	1.83	0.00	33.47
MED	2.62	3.84	6.85	0.02	39.46
All	1.71	2.98	4.47	0.00	39.46

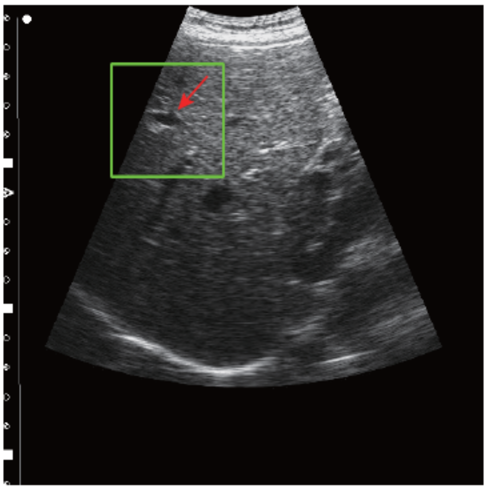


Fig. 6. An example of failure case. The red arrow points to a tracking target and the green box shows the ROI of a long-term template.

The tracking results for each sequence group (ETH and MED) are shown in Table 1. Although we do not show the result for each sequence, the mean errors for all ROIs in ETH sequence group are less than 2 mm and 30 out of 37 ROIs in MED sequence group are less than 3 mm.

In MED sequence group, there are some sequences which have tracking target near the boarder. Fig. 6 shows an example of a tracking target and a long-term template ROI in this case. As shown in Fig. 6, the long-term template includes non-ultrasound image area and it is the main reason to fail the tracking. Also, we confirmed that tracking sometimes failed when the tracking target is small tissue.

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 C++ using OpenCV and OpenMP. Note that the maximum number of threads in OpenMP is the same as the number of the annotations (trackers) for each sequence. The average processing time was about 84 msec/frame. Note that our implementation of the motion estimation utilizes an OpenCV function and it is not optimized for the proposed method yet.

4 Conclusion

In this paper, we proposed a tracking method of target tissues in long ultrasound sequences of liver. The proposed method uses multiple templates, i.e. long-term and short-term templates. The experimental results using 21 sequences of 2D ultrasound showed the proposed method had good accuracy. We also confirmed that tracking can be performed in about 84 msec per frame using a personal computer.

Items for future research are to improve the accuracy of tracking tissues near the boarder and small tissues, improve the processing speed by optimizing the motion estimation processing, 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. Bohs, L. N., Trahey, G. E.: A novel method for angle independent ultrasonic imaging of blood flow and tissue motion. IEEE Trans. on Biomedical Engineering, 38(3), 280–286 (1991)
5. Krupa, A., Fichtinger, G., Hager, G. D. Full motion tracking in ultrasound using image speckle information and visual servoing. In IEEE International Conference on Robotics and Automation, 2458–2464 (2007)
6. Revell, J., Mirmehdi, M., McNally, D: Computer vision elastography: speckle adaptive motion estimation for elastography using ultrasound sequences. IEEE Trans. on Medical Imaging, 24(6), 755–766 (2005)
7. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE Trans. on Pattern Analysis and Machine Intelligence, 26(6), 810–815 (2004)
8. De Luca, V., Tschannen, M, Szekely, 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.) MICCAI 2013, LNCS, 8149, 518–525. Springer Berlin Heidelberg (2013)

Kernel-based Tracking in Ultrasound Sequences of Liver

Tobias Benz[1], Markus Kowarschik[2,1] and Nassir Navab[1]

[1] Computer Aided Medical Procedures, Technische Universität München, Germany
[2] Angiography & Interventional X-Ray Systems, Siemens AG, Healthcare Sector, Forchheim, Germany

Abstract. *Objective:* Object tracking in 2D ultrasound sequences of liver to infer real-time respiratory organ movement and offer motion compensation in image-guided abdominal interventions.
Methods: A kernel-based tracking algorithm that is adaptive to scale and orientation changes of the tracking target is applied to 54 vessel targets in 21 ultrasound sequences acquired from volunteers under free breathing. Tracking performance is evaluated based on manually annotated ground truth information.
Results: Tracking results show that the algorithm is able to track the assessed targets in a precise and robust manner in real-time performance. The overall mean tracking error is 1.43 ± 1.22 mm.

1 Introduction

Object tracking in ultrasound (US) sequences of liver under respiratory motion is a challenging task with several applications in, for instance, motion compensation in abdominal interventions like needle biopsies, radio frequency ablations, and radiation therapy.

In this work, we present the application of a scale and orientation adaptive mean shift procedure to track vessel targets in long 2D US sequences acquired under free breathing.[3]

The mean shift procedure was first introduced by Fukunaga et al. [6] for data clustering. Cheng et al. [1] and Comaniciu et al. [3] later applied it to the task of visual object tracking. Recently, Ning et al. proposed modifications to make the mean shift tracker adaptive to orientation and scale [8]. In medical image processing, the mean shift algorithm was used for vessel segmentation in CT data [10] and for blood cell segmentation in images of blood smear [2]. In regard to tracking in US sequences, the mean shift was employed to myocardial border tracking [5]. An application to vessel tracking in US series of liver has, to our knowledge, not been presented before.

[3] The US image data was obtained from the ”CLUST 2014 MICCAI Challenge on Liver Ultrasound Tracking” (`http://clust14.ethz.ch/`).

2 Methods

2.1 Kernel Density Estimation and the Mean Shift

Given a set of samples assumed to be drawn from some probability distribution, kernel density estimation (KDE) is a method to obtain a non-parametric estimate of the underlying probability density function. The kernel density estimator $\hat{f}_h(x)$ at location $x \in \mathbb{R}^D$ of a function f is

$$\hat{f}_h(x) = \frac{1}{Nh^D} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right), \tag{1}$$

where N is the number of samples $x_i \in \mathbb{R}^D$ within the kernel K with window size h. Using the kernel profile k of the radially symmetric kernel K which satisfies $K(x) = c_k k(||x||^2)$ (c_k is a normalization factor), Eq. (1) can be rewritten into

$$\hat{f}_h(x) = \frac{c_k}{Nh^D} \sum_{i=1}^{N} k\left(\left\|\frac{x - x_i}{h}\right\|^2\right). \tag{2}$$

The output of KDE is a function that is a smoothed representation of the given sample distribution and can intuitively be understood as a generalization of weighted histograms.

To find modes in a given KDE, mean shift procedures can be applied [3,6]. The mean shift is a gradient ascent on the gradient of the density estimate

$$\nabla\hat{f}_h(x) = 2\frac{c_k}{Nh^{D+2}} \sum_{i=1}^{N} (x - x_i) k'\left(\left\|\frac{x - x_i}{h}\right\|^2\right), \tag{3}$$

$$\overset{g(x)=-k'(x)}{=} 2\frac{c_k}{Nh^{D+2}} \sum_{i=1}^{N} x_i g\left(\left\|\frac{x - x_i}{h}\right\|^2\right) - 2\frac{c_k}{Nh^{D+2}} \sum_{i=1}^{N} x g\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \tag{4}$$

$$= 2\frac{c_k}{Nh^{D+2}} \sum_{i=1}^{N} g\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \underbrace{\left[\frac{\sum_{i=1}^{N} x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{N} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x\right]}_{=m_K(x)}. \tag{5}$$

The second term in Eq. (5) is also referred to as the generalized mean shift vector $m_K(x)$. We can find modes in the gradient of the density estimate obtained by kernel K by iteratively shifting the center of K from an initial location by $m_K(x)$. When mode-seeking is applied to images, where pixels form a regular grid the generalized mean shift has to be extended to introduce the notion of pixel density. This can be achieved by employing a weighted mean shift, where each pixel location x_i is assigned a weight w_i derived from, for instance, the pixel's intensity.

2.2 Mean Shift for Tracking in Ultrasound Sequences

To apply the mean shift mode-seeking procedure to visual tracking, the tracking target is selected in the first frame and represented in a suitable feature space to obtain a model of the target. For each subsequent frame a weight image is computed by assigning a weight to each pixel which depends on the probability of the pixel belonging to the target. On this weight image, which is also referred to as a target confidence map, the mean shift algorithm is initialized with the target location in the previous frame and an appropriate kernel size. After mean shift convergence the found mode is taken as the target location in the current frame.

For mean shift tracking in US sequences we used normalized weighted intensity histograms to represent the target model $q = \{q_u\}_{u=1\dots m}$ and the target candidate model $p(y) = \{p_u\}_{u=1\dots m}$ at location y, where m is the number of bins. The weights that determine the contributions of each pixel to a histogram bin u are based on a radially symmetric kernel K. For the target model location $y = (0, 0)$ and size $h = 1$ is assumed by using normalized pixel locations x_i^*:

$$q_u = \frac{1}{C} \sum_{i=1}^{N} k(\|x_i^*\|^2) \delta[b(x_i^*) - u], \tag{6}$$

where δ is the Kronecker delta function, k the kernel profile of K and $b(x)$ a function that maps the image intensity at location x to a bin number. C is the sum of the kernel weights at all locations such that the sum of all q_u is 1. For the target candidate model in the current frame the same model representation is used, but with the kernel of size h shifted to the current target location y:

$$p_u(y) = \frac{1}{C_h} \sum_{i=1}^{N} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u], \tag{7}$$

where C_h is the sum of the kernel weights of all locations on the regular pixel lattice within the kernel with window size h.

Intuitively, the histograms q and p give the probability of a pixel's intensity belonging to the target and the target candidate model, respectively. The kernel assigns smaller weights to pixel locations farther away from the center. This increases robustness since pixels closer to the center are also closer to the target center and pixel locations close to the target center offer more reliable features due to, for instance, changes in the appearance of the target propagating from its boundaries towards the center.

Since the aim in the current frame is to find the target candidate model $p(y)$ that best matches the target model q, a similarity metric is introduced next. We follow Comaniciu et al. [4] and use the discrete Bhattacharyya coefficient [7] $\rho(y)$ to compare the target model q and the candidate model $p(y)$ at location y:

$$\rho(y) = \rho\left[p(y), q\right] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}. \tag{8}$$

Intuitively, the Bhattacharyya coefficient is a measure for the amount of overlap between two sample distributions.

In each frame t, the procedure to find the location $\hat{y}$ that maximizes $\rho(y)$ is started at location $\hat{y}_0$, which in the beginning is set to the position of the target in the previous frame $\hat{y}_{t-1}$. By linearization through a Taylor series expansion around y_0, $\rho(y)$ can be approximated as

$$\rho(y_o) \approx \frac{1}{2}\rho\left[p(y_0), q\right] + \frac{1}{2}\sum_{u=1}^{m} p_u(y)\sqrt{\frac{q_u}{p_u(y_0)}} \tag{9}$$

$$\overset{(7)}{=} \frac{1}{2}\rho\left[p(y_0), q\right] + \frac{1}{2}\sum_{u=1}^{m} \underbrace{\frac{1}{C_h}\sum_{i=1}^{N} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)\delta[b(x_i)-u]}_{p_u(y)}\sqrt{\frac{q}{p(y_0)}} \tag{10}$$

$$= \underbrace{\frac{1}{2}\rho\left[p(y_0), q\right]}_{\text{independent of } y} + \underbrace{\frac{1}{2C_h}\sum_{i=1}^{N} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}_{\text{KDE obtained with kernel } K \text{ at location } y}, \tag{11}$$

where

$$w_i = \sum_{u=1}^{m} \delta\left[b(x_i)-u\right]\sqrt{\frac{q_u}{p_u(y)}}. \tag{12}$$

The first term in Eq. (11) is independent of y, whereas the second term is a KDE obtained using the kernel K at location y and weights w_i, which can be maximized using the mean shift algorithm (cf. Section 2.1). Maximizing this KDE means maximizing the Bhattacharyya coefficient, which finally leads to the minimization of the distance between $p(y)$ and q.

The mean shift iteration step to move the kernel center position from $\hat{y}_0$ to the new position $\hat{y}_1$ is

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0-x_i}{h}\right\|^2\right)}. \tag{13}$$

A favorable choice for the kernel K is the Epanechnikov kernel

$$K_E(x) = \begin{cases} \frac{1}{2}\frac{D+2}{c_D}(1-\|x\|^2) & \text{if } \|x\| < 1 \\ 0 & \text{else} \end{cases}, \tag{14}$$

where $x \in \mathbb{R}^D$ and c_D is the volume of the $D-$dimensional unit sphere. The Epanechnikov kernel minimizes the mean integrated squared error between the KDE and the true density [9] and, since the profile k_E of K_E is half-triangular we see that $g(x) = -k'_E(x) = 1$. Thus, Eq. (13) can be reduced to

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i}. \tag{15}$$

After each mean shift iteration, convergence is checked based on maximum number of iterations and minimum length of the mean shift vector. In case of convergence $\hat{y}_t$ is set to $\hat{y}_1$, otherwise $\hat{y}_0$ is set to $\hat{y}_1$ and the mean shift procedure is repeated.

2.3 Scale and Orientation Adaptive Mean Shift Tracking

In the original mean shift tracking algorithm the kernel window size and orientation remains fixed. This is unfavorable when tracking a target that changes its size and orientation over the course of the image sequence. We follow modifications proposed by Ning et al. [8] to make the procedure adaptive to scale and orientation. To this end, first the target's scale is estimated, which is the area in the target search region occupied by the target. The estimated area is then used to adjust an ellipsoid target descriptor to match the current width, height, and orientation of the tracking target.

Estimating the Target's Scale In each frame t the kernel center position is initialized with the target's position in the previous frame $\hat{y}_{t-1}$. Also the kernel is slightly enlarged by a factor Δd, enabling the algorithm to capture a tracking target that increased in size since the last frame. Since the weight w_i for each pixel within the increased search region (cf. Eq. (12)) gives the likelihood of the pixel being part of the target, the sum of all weights (i.e. the 0^{th} order image moment of the search region in the weight image or target confidence map) $M_{00} = \sum_{i=1}^{N} w_i$ is a good initial approximation of the area of the search region covered by the tracking target. However, if background features are present in the target search region, the weights of pixels within the search region that belong to the target are amplified. This is because the probability of target features in the target search area is decreased in the presence of background pixels, which, as per Eq. (12) (the target candidate intensity distribution $p(y)$ is in the denominator), increases the weights of target pixels. Therefore, the 0^{th} order moment overestimates the size of the tracking target in case background features are present.

On the other hand, the Bhattacharyya coefficient between the target model q and the target candidate model $p(y)$ is a measure for how many target and background features are in the current search region. Therefore Ning et al. [8] proposed to use the Bhattacharyya coefficient to adjust the 0^{th} order moment approximation for the target scale. The estimated area is computed as

$$\hat{A} = \exp\left(\frac{\rho}{\sigma}\right) M_{00}, \tag{16}$$

where σ is a parameter that governs the magnitude of adjustment of the M_{00} estimate given a certain Bhattacharyya value. In the experiments described below σ was empirically set to 0.2.

Estimating the Target's Orientation For estimating the target's orientation an ellipsoid image descriptor (cf. Fig. 1) is introduced which is defined by a covariance matrix based on the first and second order central image moments

$$\text{Cov} = \begin{pmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{pmatrix}, \quad \text{with} \quad \mu'_{pq} = \frac{\sum_{i=1}^{N} (x_{i,1} - \bar{x}_1)^p (x_{i,2} - \bar{x}_2)^q w_i}{\sum_{i=1}^{N} w_i}, \tag{17}$$

where $(\bar{x}_1, \bar{x}_2)$ is the kernel center position. An orthogonal decomposition of Cov

$$\text{Cov} = U \times S \times U^T = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^T \tag{18}$$

yields the semi-major axis a and semi-minor axis b of the target descriptor as column vectors in U and the aspect ratio $\frac{a}{b} = \frac{\lambda_1}{\lambda_2}$ through the singular values in S. Subsequently, a scaling factor k can be introduced such that $a = k\lambda_1$ and $b = k\lambda_2$. Using the previously estimated target area $\hat{A}$ (cf. Section 2.3) and the general area formula for an ellipse, we can further derive

$$\hat{A} = \pi ab = \pi(k\lambda_1)(k\lambda_2) \Longrightarrow k = \sqrt{\frac{\hat{A}}{\pi\lambda_1\lambda_2}}, \tag{19}$$

which finally allows us to adjust the ellipsoid descriptor based on the estimated target scale:

$$\text{Cov} = U \times S \times U^T = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} \frac{\hat{A}\lambda_1}{\pi\lambda_2} & 0 \\ 0 & \frac{\hat{A}\lambda_2}{\pi\lambda_1} \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^T. \tag{20}$$

2.4 Tracking Failure Recovery

Tracking may be lost over the course of the US sequence due to, for instance, drastic change of appearance of the tracking target, too large target displacements between frames, or erroneous estimation of the target's scale and orientation leading to a disadvantageous search area. Based on the assumed periodicity in the motion of liver vessels induced by respiration we integrated strategies to detect frames in which tracking performance is problematical and to recover from these situations. A first check is based on the analysis of the Bhattacharyya coefficient ρ. If it drops below 0.8 we discard the found target position and instead use the target position from the previous frame. Furthermore, the search region is reset to the one set by the user in the first frame. If this check triggers twice in two subsequent frames, the target position is reset to the centroid of the search area selected in the first frame. A second check is based on the analysis of the estimated target size. If the target search area in the current frame is found to be larger than three times the initial target's size, the search area and its position is reset to the one set in the first frame. These failure recovery strategies are rather crude but were found to only be triggered in rare situation where tracking would otherwise fail completely.

3 Results

The scale and orientation adaptive mean shift procedure was applied to track 54 vessel targets in 21 2D US sequences of liver acquired from volunteers under free breathing. In total, the sequences comprised 91619 frames. The overall mean tracking error (MTE) was 1.43 mm with a standard deviation (SD) of 1.22 mm and 95th-percentile 3.67 mm. The minimum tracking error over all frames and tracking targets was 0.01 mm, the maximum tracking error 16.01 mm. The algorithm was developed in MATLAB Release 2013b, and the experiments were conducted on a machine equipped with an Intel i5-3320M processor at 2.6 GHz clock speed and 8 GB RAM. Tracking speed using this hardware set up was about 20 Hz. Table 1 gives an overview of the data set and the results obtained. Fig. 1 gives a visual impression of the tracking of three vessels in series MED-02.

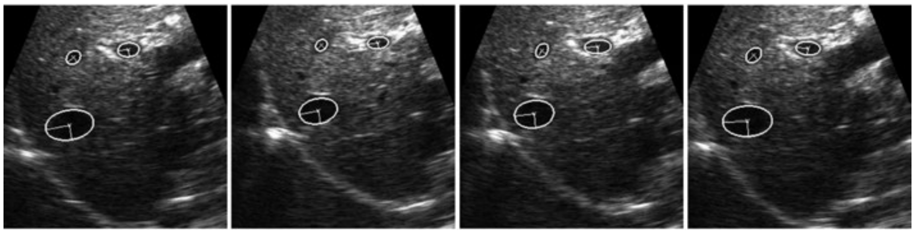


Fig. 1: Ellipsoid target descriptors of three tracking targets overlaid on four frames of the US sequence MED-02. [4]

4 Conclusion

Tracking of vessel targets in 2D US series of liver under free breathing using a scale and orientation adaptive kernel-based tracking algorithm is feasible, fast, robust and precise. For future work the incorporation of a target descriptor that is adaptive to the outline of the tracking target seems worthwhile. By suggestion of one reviewer of an initial draft of this article, we will also look into making the histogram representations adaptive to global illumination changes. The application to native 3D US is also desirable. Furthermore, other feature spaces for model representation could be investigated with a focus on, for instance, gradient-based descriptors and joint-histograms. Finally, a more sophisticated failure recovery strategy based on, for instance, a collection of keyframes or predictive motion regularization could be advantageous.

[4] Link to video: `http://campar.in.tum.de/files/benz/CLUST2014/MED-02.webm`

Sequence information						Results				
Sequence	**Targets**	**Frames**	**Resolution**	**Probe freq.**	**FPS**	**MTE**	**SD**	**95%**	**Min**	**Max**
			[mm/px]	[Hz]	[Hz]	[mm]	[mm]	[mm]	[mm]	[mm]
ETH-01	1	14 516	0.71	2.22	25	2.47	1.29	4.15	0.16	11.14
ETH-02	1	5244	0.40	2.00	16	0.60	0.38	1.26	0.04	2.64
ETH-03	3	5578	0.36	1.82	17	1.34	0.69	2.32	0.07	10.34
ETH-04	1	2620	0.40	2.22	15	1.05	0.80	2.10	0.05	7.33
ETH-06	2	5586	0.37	1.82	17	2.67	1.17	4.65	0.04	8.21
ETH-07	1	4588	0.28	2.22	14	0.85	0.54	1.95	0.01	3.21
ETH-08	2	5574	0.36	1.82	17	1.53	0.54	2.56	0.06	4.89
ETH-09	2	5247	0.40	1.82	16	0.85	0.46	1.67	0.01	5.89
ETH-10	4	4587	0.40	1.82	15	0.83	1.05	1.72	0.01	16.01
All ETH sequences						**1.46**	**1.31**	**3.77**	**0.01**	**16.01**
MED-01	3	2470	0.41	5.50	20	0.67	0.49	1.60	0.01	5.07
MED-02	3	2478	0.41	5.50	20	1.04	0.67	2.48	0.04	6.29
MED-03	4	2456	0.41	5.50	20	1.17	0.66	2.43	0.04	4.31
MED-05	3	2458	0.41	5.50	20	1.17	0.65	2.31	0.07	4.33
MED-06	3	2443	0.41	5.50	20	1.84	0.94	3.54	0.10	5.89
MED-07	3	2450	0.41	5.50	20	1.52	0.88	3.15	0.04	6.72
MED-08	2	2442	0.41	5.50	20	1.46	0.81	2.89	0.05	4.76
MED-09	5	2436	0.41	5.50	20	1.29	0.78	2.90	0.05	10.73
MED-10	4	2427	0.41	5.50	20	1.79	1.20	4.16	0.03	13.02
MED-13	3	3304	0.35	4.00	11	1.21	0.70	2.48	0.03	6.32
MED-14	3	3304	0.35	4.00	11	1.73	0.98	3.51	0.05	8.23
MED-15	1	3304	0.35	4.00	11	2.62	1.36	5.10	0.13	6.50
All MED sequences						**1.40**	**1.13**	**3.49**	**0.01**	**13.02**
All sequences						**1.43**	**1.22**	**3.67**	**0.01**	**16.01**

Table 1: Overview of the data set comprising 54 vessel targets in 21 US sequences.

References

1. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE PAMI 17(8), 790–799 (1995)
2. Comaniciu, D., Meer, P.: Cell image segmentation for diagnostic pathology. In: Advanced algorithmic approaches to medical image segmentation, pp. 541–558. Springer (2002)
3. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE PAMI 24(5), 603–619 (2002)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE PAMI 25(5), 564–577 (2003)
5. Comaniciu, D., Zhou, X.S., Krishnan, S.: Robust real-time myocardial border tracking for echocardiography: an information fusion approach. IEEE TMI 23(7), 849–860 (2004)
6. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEE TIT 21(1), 32–40 (1975)
7. Kailath, T.: The divergence and Bhattacharyya distance measures in signal selection. IEEE TCT 15(1), 52–60 (1967)
8. Ning, J., Zhang, L., Zhang, D., Wu, C.: Scale and orientation adaptive mean shift tracking. CV, IET 6(1), 52–61 (2012)
9. Scott, D.W.: Multivariate density estimation: theory, practice, and visualization, vol. 383. John Wiley & Sons (2009)
10. Tek, H., Comaniciu, D., Williams, J.P.: Vessel detection by mean shift based ray propagation. In: MMBIA Workshop. pp. 228–235. IEEE (2001)

A Non-Linear Image Registration Scheme for Real-Time Liver Ultrasound Tracking using Normalized Gradient Fields

Lars König, Till Kipshagen and Jan Rühaak

Fraunhofer MEVIS Project Group Image Registration, Lübeck, Germany
lars.koenig@mevis.fraunhofer.de

Abstract. We propose a novel scheme for annotation tracking in long liver ultrasound sequences. It is based on a variational non-linear image registration method using Normalized Gradient Fields, extended by a moving window strategy based on registrations to the provided annotation on the first frame. By this we achieve robustness against error accumulation, while handling large deformations at the same time. The method is evaluated on 21 datasets with up to five annotations as contribution to the MICCAI CLUST14 challenge. We achieved a mean tracking error of 1.31 mm with a standard deviation of 1.63 mm, while running at close to real-time speed, exceeding acquisition rate in ten cases with up to 44 frames per second on standard hardware.

Keywords: tracking, non-linear image registration, normalized gradient fields, liver ultrasound, real-time, CLUST14

1 Introduction

Ultrasound imaging provides unbeaten acquisition speed while having low requirements in component setup. This makes ultrasound a preferable choice where real-time information about patient condition is needed, e.g. for fusion of intra-operative ultrasound images to pre-operative CT images [9] or motion compensation in image guided radiation therapy [5].

To enable fusion of real-time image sequences to planning data, often tracking of relevant features in ultrasound images is needed. Especially in long time series, due to noise and breathing motion, this can be a challenging task [2]. Many different approaches to ultrasound tracking exist, ranging from optical flow methods and speckle tracking up to different forms of image registration [1]. In image registration, especially deformable methods are of interest, as they provide deformation models that are able to represent non-linear deformations in soft tissue. As ultrasound images are typically acquired at high frame-rates, common non-linear image registration schemes are not capable of achieving real-time performance. However, due to recent developments of highly efficient computation schemes [6], even sophisticated variational methods have become an attractive choice for real-time tracking.

In this paper, we present a new tracking scheme based on a fast non-linear image registration algorithm that allows real-time ultrasound tracking. The algorithm does not rely on image segmentations, makes no assumptions about the expected motion and does not require a training phase. By computing registrations on moving image windows, which are related to the given annotation of the time-series, we achieve robustness against error accumulation, while handling large deformations at the same time. We evaluated this new scheme participating in the MICCAI CLUST14 liver ultrasound tracking challenge.

2 Method

The proposed tracking scheme is based on a variational image registration approach [7]. It is embedded in a specialized framework allowing for processing of image sequences and efficient compensation of breathing motion. In Section 2.1, we first describe the non-linear registration algorithm, that is then used as a basis for the tracking algorithm described in Section 2.2.

2.1 Image Registration

Let $\mathcal{R} : \mathbb{R}^2 \to \mathbb{R}$ denote the fixed reference image and $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}$ the moving template image with compact support in domain $\Omega \subseteq \mathbb{R}^2$. The goal of image registration is to find a *transformation* $y : \Omega \to \mathbb{R}^2$ that encodes the spatial correspondence between the two images $\mathcal{R}$ and $\mathcal{T}$. In variational approaches, this is modeled by an objective function $\mathcal{J}$ called *joint energy function* which typically consists of a distance term $\mathcal{D}$ describing image similarity and a regularizer $\mathcal{S}$ which penalizes implausible deformations [7]. Image registration then translates to minimizing the functional

$$\mathcal{J}(y) = \mathcal{D}(\mathcal{R}, \mathcal{T}(y)) + \alpha \mathcal{S}(y). \tag{1}$$

Here, the regularization parameter α enables a balance between data fit and deformation regularity.

As image edges are prominent features in ultrasound images, we choose the edge-based Normalized Gradient Fields (NGF) distance measure [4]

$$\mathcal{D}(\mathcal{R}, \mathcal{T}(y)) := \int_{\Omega} 1 - \left(\frac{\langle \nabla \mathcal{T}(y(x)), \nabla \mathcal{R}(x) \rangle_\eta}{\|\nabla \mathcal{T}(y(x))\|_\eta \, \|\nabla \mathcal{R}(x)\|_\eta} \right)^2 \, \mathrm{d}x \tag{2}$$

with

$$\langle f, g \rangle_\eta := \sum_{j=1}^{2} f_j g_j + \eta^2 \qquad \text{and} \qquad \|f\|_\eta := \sqrt{\langle f, f \rangle_\eta}.$$

Note that $\| \cdot \|_\eta$ does not define a norm in the mathematical sense as $\|0\|_\eta \neq 0$ for $\eta \neq 0$. Roughly speaking, NGF measures the angle between reference and template image intensity gradients at each point and aims for alignment of these image gradients. The *edge parameter* η is introduced to suppress the influence of small edges e.g. caused by noise. The choice of the parameters η and α is discussed in Section 2.4.

As we generally expect smooth deformations between the ultrasound time frames, we select the *curvature regularizer* as proposed in [3] which is based on second order derivatives. With the decomposition $y(x) = x + u(x)$, the curvature regularizer is given by

$$\mathcal{S}(y) := \frac{1}{2} \int_{\Omega} \|\Delta u_x\|^2 + \|\Delta u_y\|^2 \, \mathrm{d}x,$$

where u_x, u_y denote the components of the displacement in x- and y-direction, respectively. The curvature regularizer penalizes the Laplacian of the displacement components, thus generating very smooth deformations.

The minimization of (1) is performed following the discretize-then-optimize paradigm [7]. In this ansatz, all components (distance measure, regularizer and transformation) are first carefully discretized, yielding a continuous, yet finite dimensional optimization problem. This enables the usage of standard algorithms from numerical optimization [8]. We employ the quasi-Newton L-BFGS optimization scheme to minimize the objective function $\mathcal{J}$ for its speed and memory efficiency. The implementation is based on the two-loop recursion formulation as presented in [8]. The occurring linear equation system in each iteration step of the Newton scheme is solved using a conjugate gradient method. Furthermore, to avoid local minima, the iteration scheme is embedded in a multi-level approach [7], where the optimization problem is solved consecutively on coarse to fine image resolution levels.

The evaluation of the objective function $\mathcal{J}$ together with its derivative is performed using the algorithm presented in [6]. This approach includes an explicit calculation rule for the derivative of $\mathcal{J}$, which does not require any storage of Jacobian matrices and allows for a full pixelwise parallel computation.

2.2 Tracking Algorithm

The non-linear registration described in Section 2.1 is embedded in a larger framework to enable efficient usage in annotation tracking on ultrasound sequences. By calculating registrations of moving windows on each image of the time series, we enable the tracking of an annotation $a_1 \in \mathbb{R}^2$, given on the first frame, over time. The proposed tracking scheme is illustrated in Figure 1.

Let $I_k \in \mathbb{R}^{M \times N},\ k = 1, \ldots, T$ denote the k-th frame of the ultrasound sequence of length T. We then define $W_n(I_k) : \mathbb{R}^{M \times N} \to \mathbb{R}^{w_1 \times w_2},\ n, k = 1, \ldots, T$ as a window of I_k with extent $w_1, w_2 \in \mathbb{N}, w_1 \leq M, w_2 \leq N$ and center position $c_n \in \mathbb{R}^2$. Starting with the original annotation $a_1 \in \mathbb{R}^2$ on I_1, window $W_1(I_1)$ with center $c_1 = a_1$ is chosen. The extent w_1, w_2 is kept constant throughout the algorithm and will be discussed in Section 2.4. Initially, a registration between $W_1(I_1)$ as reference and $W_1(I_2)$ as template image is performed where the continuous image representation, as defined in Section 2.1, is obtained by bilinear interpolation. Using the registration result y_1, the initial annotation a_1 is then

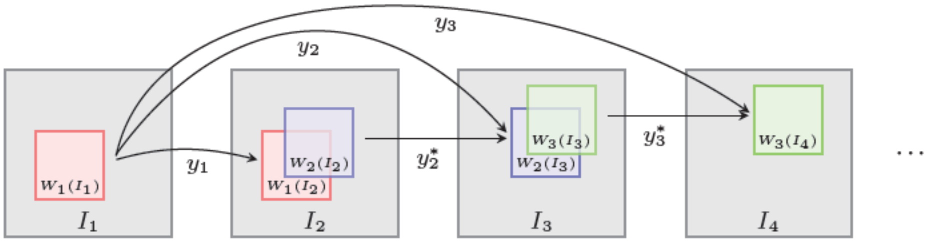


Fig. 1. Proposed tracking scheme. The registration between window $W_{n-1}(I_n)$ on current frame and window $W_1(I_1)$ on first frame is denoted by y_n. If this first registration fails, we compute y_n^* by registering corresponding windows on current and previous frame.

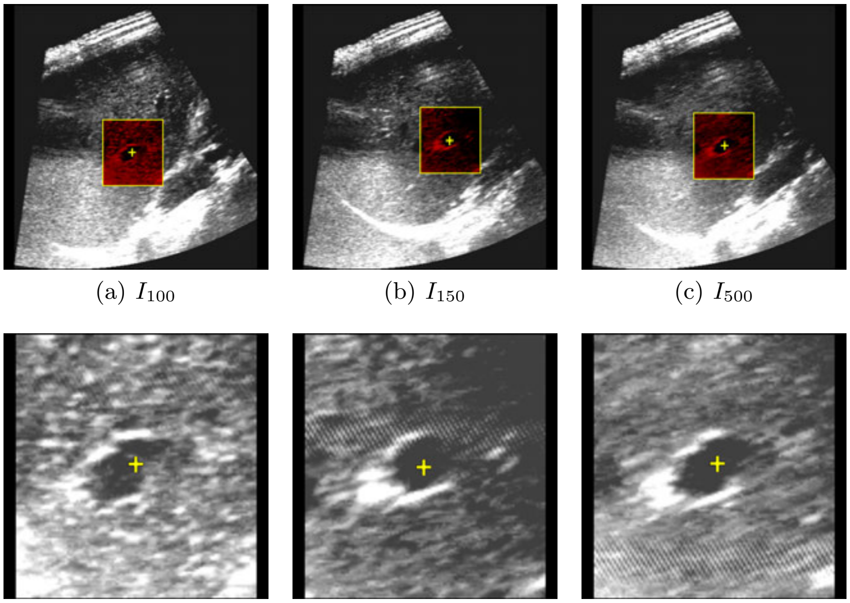
transformed as $a_2 = y_1(a_1)$, yielding the moved annotation a_2 tracked to frame I_2. After the transformation step, a new window $W_2(I_3)$ is computed, now with $c_2 = a_2$. Window $W_2(I_3)$ is then registered to $W_1(I_1)$, yielding transformation y_2 that is used to compute moved annotation $a_3 = y_2(a_1)$. This process is then repeated for all frames. A pseudocode of this scheme is shown in Algorithm 1.

Algorithm 1 Pseudocode for the tracking algorithm of a single landmark

```
load $I_1, a_1$                                              ▷ Load first image and original annotation
for $n$ in $[2, T]$ do                                       ▷ Loop over all frames
    load $I_n$
    $W_{n-1} \leftarrow$ window around $a_{n-1}$
    $y_{n-1} \leftarrow$ registration$(W_1(I_1), W_{n-1}(I_n))$       ▷ Register to first window
    if registration was successful then
        $a_n \leftarrow y_{n-1}(a_1)$                                ▷ Compute new annotation
    else                                                     ▷ change of strategy
        $y_{n-1}^* \leftarrow$ registration $(W_{n-1}(I_{n-1}), W_{n-1}(I_n))$   ▷ Register to prev. window
        $a_n \leftarrow y_{n-1}^*(a_{n-1})$                          ▷ Compute new annotation
    end if
end for
```

Additionally, a safeguard procedure is included in the algorithm. In each registration step, the success of the current registration is checked by determining the final value of the distance measure (2). If this value is above a certain threshold θ, described in detail in Section 2.4, the registration is considered having failed. In this case, the registration paradigm is switched. Instead of registering $W_{n-1}(I_n)$ onto $W_1(I_1)$, we now use window $W_{n-1}(I_{n-1})$ of the previous frame as reference image and register $W_{n-1}(I_n)$ onto $W_{n-1}(I_{n-1})$ instead, yielding y_{n-1}^*. This step is based on the interpretation of a_{n-1} as the last successfully tracked annotation. It enables landmark tracking in cases where large local differences compared to the first window exist, but consecutive frames are still relatively similar. If this procedure has to be repeated multiple times, error accumulation may occur, as the tracking relies on tracked annotations from previous steps, that may themselves contain errors. However, as soon as the differences to the first frame decrease, the original scheme takes over and possibly accumulated errors are discarded by deforming the ground truth annotation a_1 again. This mechanism enables a successful tracking also in situations where the difference to the initial frame is temporarily large.

The proposed algorithm has several benefits. It does not require a training phase, avoids error accumulation, makes no assumptions about motion periodicity and does not rely on image segmentations. By choosing $W_1(I_1)$ as a fixed reference window, we always refer to the given annotation a_1 throughout the whole tracking process. While possibly being at completely different locations, the image contents inside the windows only exhibit small movements as larger movements of the structure have already been compensated for by shifting the window according to the deformation obtained in the prior iteration, see Figure 2. This procedure generates an excellent starting point for the underlying non-linear registration scheme.


(a) I_{100} (b) I_{150} (c) I_{500}

(d) $W_{99}(I_{100})$ (e) $W_{149}(I_{150})$ (f) $W_{499}(I_{500})$

Fig. 2. Selected frames of dataset ETH-07. While the colored window (tracked annotation at its center) experiences globally large movements (top row), the contents of the respective windows remain similar (bottom row).

2.3 Annotation coupling

The tracking scheme is in principle designed for tracking single landmarks. If multiple landmarks are to be tracked, the choice of window and registration are performed separately for each landmark, yielding independent deformations and windows. In the described scheme, however, dense deformation fields are computed at all times on the area of the chosen windows. We exploit the deformation field for annotations lying close together by coupling a secondary annotation with the window of a primary annotation instead of tracking these annotations independently. The primary annotation is still tracked identically to the scheme described in Section 2.2, while the secondary annotation is deformed using the computed deformation field within the chosen window. This increases tracking performance by enabling tracking multiple landmarks with the same registration.

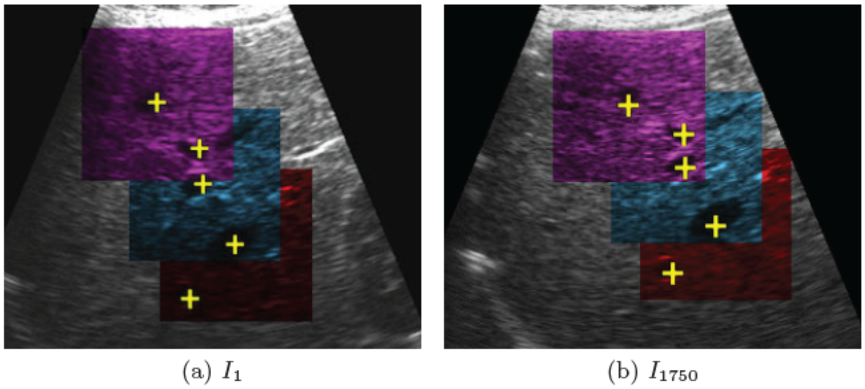

(a) I_1 (b) I_{1750}

Fig. 3. Tracking of five annotations in dataset MED-09, first frame (a) (part of full image shown) and intermediate result (b). The individual windows are shown as colored rectangles. The second annotation is coupled with the third and the fifth with the fourth (numbering from top to bottom).

2.4 Parametrization

Our tracking algorithm provides several parameters to adapt to varying image characteristics. While these parameters were manually determined, it generally suffices to select them once per device and not per dataset.

For the CLUST14 challenge, we focused on parameters fitting all datasets from the same scanner and probe. While better results can be obtained by choosing specialized parameters for each dataset, this would contradict the purpose of annotation tracking in a real-time setting, where future frames are unknown.

For all processed datasets, parameters for optimization, deformation resolution and multi-level scheme were kept constant. The window size was determined as one fourth of the image size in each dimension. For all registrations, we chose two levels in the multi-level scheme, with a downsampled version of half the window resolution on the finest level. The deformation resolution was determined as 17 × 17, thus reducing the computational costs and additionally acting as a regularizer, see [11] for details.

Important parameters that allow adapting to different noise and device characteristics are the regularization factor α and the NGF noise parameter η, see Section 2.1. The threshold θ for the safeguard strategy described in Section 2.2 is adaptively chosen as $\theta = \tau \cdot \frac{a}{4096}$, depending on the window area a. Using all given datasets, the parameters α, η and τ were manually calibrated per device and probe. For the ETH datasets, we used $\alpha = 0.1, \eta = 10$ and $\tau = 1490$, except for ETH-1, where because of the different resolution, we set $\eta = 2.5$. For the MED datasets $\alpha = 0.5, \eta = 5$ and $\tau = 1400$ were used, except for MED-15, where $\tau = 850$ was used to compensate for a single exceptional artifact. The annotations were coupled as follows. ETH-03: $3 \to 2$, ETH-10: $4 \to 3$, MED-03: $2 \to 4$, MED-05: $3 \to 2$, MED-09: $3 \to 2$, $5 \to 1$, MED-10: $2 \to 4$.

3 Results and Discussion

The non-linear image registration was implemented in C++, while the tracking framework was scripted in Python and executed in MeVisLab [10]. Our method was evaluated on all 2D annotation tracking test datasets provided by the organizers of the CLUST2014 challenge [2]. These datasets contained image sequences from 264×313 (ETH-01) to 524×591 (MED-13 – MED-15) pixels in resolution, with number of frames ranging from 2427 (MED-10) to 14516 (ETH-01) frames. Every first frame was provided with up to five annotations.

On the ETH datasets, our method achieved a mean tracking error (MTE) of 0.89 mm with a standard deviation (σ) of 1.84 mm. For the MED datasets, we achieved a MTE of 1.73 mm with $\sigma = 1.25$ mm, resulting in overall values of MTE=1.31 mm and $\sigma = 1.63$ mm. Full results are given in Table 1. It has to be noted, that the overall tracking results were negatively influenced by a tracking failure in dataset ETH-07 that affected only the last ≈ 250 of total 4588 frames.

The algorithm achieved close to real-time performance in all cases, exceeding acquisition rate in ten cases, computed on a three year old Intel i7-2600 PC with 3.40GHz running Ubuntu Linux 12.04, see Table 1 for computation speed. Thus real-time performance is easily within reach when using recent hardware.

Currently the choice of suitable parameters requires significant manual fine tuning since neither spacial image resolution information nor device characteristics such as center frequency were taken into account. Including this information is subject to future research and may enable automatic parameter calibaration.

We developed a fast and accurate ultrasound tracking algorithm capable of achieving real-time performance while relying solely on image information, without any further knowledge like image segmentation or feature recognition required. Furthermore, no prior training phase is needed and no assumptions about the type of movement are made.

The proposed scheme can easily be extended to 3D tracking. Since dense deformation fields are computed in every step, the algorithm can also be used directly for tracking of segmentations.

Acknowledgment: This work was supported by the Fraunhofer Internal Programs under Grant No. MAVO 823 287.

Dataset	IAR	FPS	MTE$_1$	σ_1	MTE$_2$	σ_2	MTE$_3$	σ_3	MTE$_4$	σ_4	MTE$_5$	σ_5
ETH 01	25	43.5	0.87	0.98								
02	16	31.7	0.97	0.46								
03	17	14.6	0.37	0.21	0.64	0.36	0.47	0.24				
04	15	33.8	0.86	1.20								
06	17	14.8	0.62	0.60	1.13	0.85						
07	14	33.3	2.87	7.38								
08	17	15.0	0.59	0.32	0.68	0.45						
09	16	18.3	0.69	0.34	1.01	0.54						
10	15	11.1	1.07	0.74	0.80	0.66	0.93	1.23	0.94	1.28		
MED 01	20	15.5	1.09	0.61	0.94	0.42	1.04	0.61				
02	20	14.3	1.03	0.57	1.30	0.88	1.94	0.46				
03	20	14.9	1.23	0.62	2.72	1.84	1.20	0.75	0.91	0.49		
05	20	22.6	2.02	0.95	2.14	0.86	2.56	1.08				
06	20	14.3	1.71	0.91	1.22	0.55	1.37	0.59				
07	20	13.9	3.39	2.22	1.49	0.90	2.17	1.28				
08	20	21.9	2.03	1.06	2.52	1.52						
09	20	14.2	2.31	1.72	1.23	0.64	1.21	0.81	2.42	0.91	2.71	2.29
10	20	13.0	2.25	1.01	1.67	0.92	2.12	0.96	1.34	0.98		
13	11	11.1	1.08	0.69	2.14	1.35	1.13	0.62				
14	11	11.6	1.72	0.93	1.88	1.02	2.64	1.64				
15	11	29.9	1.32	1.28								

Table 1. Mean tracking error (MTE) and standard deviation (σ) in the CLUST14 2D annotation tracking datasets (empty cells correspond to fewer annotations). Processing speed is given as frames per second (FPS), image acquisition rate in column IAR.

Bibliography

[1] De Luca, V.: Liver motion tracking in ultrasound sequences for tumor therapy. Ph.D. thesis, ETH Zurich (2013)

[2] 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 2013, pp. 518–525. Springer (2013)

[3] Fischer, B., Modersitzki, J.: Curvature based image registration. Journal of Mathematical Imaging and Vision 18(1), 81–85 (2003)

[4] Haber, E., Modersitzki, J.: Intensity gradient based registration and fusion of multi-modal images. Methods of Information in Medicine 46, 292–9 (2007)

[5] Keall, P.J., et al.: The management of respiratory motion in radiation oncology report of AAPM Task Group 76. Medical Physics 33(10), 3874–3900 (2006)

[6] König, L., Rühaak, J.: A Fast and Accurate Parallel Algorithm for Non-Linear Image Registration using Normalized Gradient Fields. In: IEEE International Symposium on Biomedical Imaging: From Nano to Macro (2014)

[7] Modersitzki, J.: FAIR: Flexible Algorithms for Image Registration. SIAM (2009)

[8] Nocedal, J., Wright, S.: Numerical optimization. Springer (1999)

[9] Olesch, J., et al.: Fast intra-operative nonlinear registration of 3D-CT to tracked, selected 2D-ultrasound slices. In: SPIE Medical Imaging (2011)

[10] Ritter, F., Boskamp, T., Homeyer, A., Laue, H., Schwier, M., Link, F., Peitgen, H.O.: Medical image analysis. Pulse, IEEE 2(6), 60–70 (2011)

[11] Rühaak, J., Heldmann, S., Kipshagen, T., Fischer, B.: Highly accurate fast lung CT registration. In: SPIE Medical Imaging (2013)

High Performance Online Motion Tracking in Abdominal Ultrasound Imaging

Dennis Lübke[1] and Cristian Grozea[2]

[1] Fraunhofer MEVIS, Bremen, Germany,
dennis.luebke@mevis.fraunhofer.de,
http://www.mevis.fraunhofer.de
[2] Fraunhofer FOKUS, Berlin, Germany

Abstract. In this paper we describe the algorithms designed for motion tracking and compensation in ultrasound imaging of the liver: a) two algorithms for automatically and accurately inferring the continuous 2D position of landmarks from 2D ultrasound image sequences with processing times between 40-250 ms per frame. b) an algorithm for the continuous prediction of the landmark's position using as input only the motion vectors of the tracking methods by exploiting the quasi-periodic behavior of respiratory motion in the upper abdomen. All proposed algorithms are suitable for on-line usage. The purpose of this combination is to cope with the latency that is inherent during ultrasound image streaming for direct processing and to be capable to compensate for other mechanical latencies that can occur in devices using the tracked positions as input. The performance of the methods has been evaluated on the 2D and 3D point datasets provided by the MICCAI CLUST challenge. The results obtained from the CLUST datasets proved the high accuracy (mean average error (MAE) of respectively 1.82 ± 2.35 mm and1.98 ± 2.78 mm for the 2D datasets, 2.04 ± 2.36 mm for the prediction and 5.24 mm for the 3D datasets) and the synergy of the algorithms proposed.

Keywords: ultrasound, tracking, prediction, motion compensation

1 Introduction and Related Work

Respiratory motion in the upper abdomen is currently an obstacle for minimally and non-invasive medical treatments such as focused ultrasound/HIFU and radiotherapy [1]. The difficulty herein comes from the fairly high amplitude of the organ motion induced by respiration and from other influences such as cardiac motion and drifts resulting from muscle relaxation. Even though it is possible to minimize respiratory motion by breath-holding or by selective ventilation of the lungs, it is desirable to allow the patient to breathe freely without taking into account an impact on the actual treatment [2]. While MR imaging is known for superior image quality it suffers from low frame rates and is not suitable for motion compensation without combining additional respiratory sensors with high temporal resolution. This allows it to overcome the undersampling of the image-data [3]. Compared to MRI, ultrasound imaging bears much promise as input for tracking solutions at high frame rates at the cost of reduced image quality and a smaller field of view. Additionally ultrasound imaging allows isotropic image resolution in 3D and does not suffer from intra-frame motion.

Development of fast tracking methods on continuous ultrasound imaging is currently subject of research. State-of the-art methods for motion estimation in 2D and 3D suggest ultrasonic speckle tracking [4], or rigid [5] and non-rigid registration [6]. More sophisticated approaches make use of statistically validated motion models [7] [8] or combine scale-adaptive block-matching algorithms with learning-based techniques [9].

To achieve on-line motion compensation, it is necessary that the methods work in real-time, i.e. faster than the incoming imaging frame rate. Additionally, it is necessary for some setups to compensate for latencies that might occur during ultrasound image streaming [10] [11] and mechanical latencies [12] [13] (pages 1 to 4). For this reason, it will be useful to combine the tracking method with additional prediction algorithms to obtain a glimpse into the near future.

In this paper we present two algorithms for fast and robust tracking, one of them capable to cope with frame rates of up to 25 Hz, and combine the tracking results with a prediction method that allows a prediction horizon of 200 ms. Additionally, we introduce a real-time capable smoothing of the tracked point's trajectory using polynomial fitting, which improves the prediction.

2 Methods

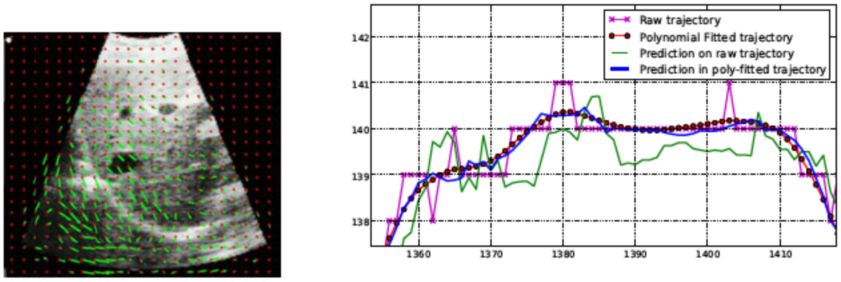
2.1 Datasets

The CLUST challenge provided 23 2D US sequences of the liver of volunteer test subjects under free breathing with a duration of 120 to 580 seconds. The sequences have a temporal resolution of 11-25 Hz and isotropic in-plane resolution of 0.35-0.71 mm. Two of the sequences come with 2 respectively 3 manually labeled ground-truth annotations for approximately 10% of the frames as a training set. A total of 54 points had to be tracked in the test-set where only the initial position was given.

The 3D datasets consist of 11 sequences with a duration of 5.8 to 27 seconds. The temporal resolution varies from 6 to 24 Hz depending on the sequence. The spatial resolution is not necessarily isotropic for all sequences and is in a range of 0.3 mm up to 1.2 mm. 21 annotations have been provided for the first frame as the test-set and 4 points as training data.

2.2 2D Motion Tracking using Dense Optical Flow

The first 2D motion tracking method is using a two-frame motion estimation based on G. Farnebaeck's polynomial expansion [14] and has originally been adapted by us for motion tracking in MR images. We use the first frame as a reference and apply the tracking on each subsequent frame in comparison to the reference. The result of this operation is a dense motion vector field for the entire



(a) Motion vector field

(b) Effect of polyfitting on prediction

Fig. 1. a)Motion vector field; (b)Prediction on polyfitted vs. non-polyfitted tracking

frame - Fig. 1(a). Compared to other methods that spend the entire processing time on tracking one single landmark, this methods was most suitable to deliver motion information for the entire frame without increasing the processing time when considering multiple landmarks. Before generating the motion vector field additional pre-processing and filtering is applied to obtain a more stable result. Histogram equalization is used to adapt the overall contrast and bilateral filtering is employed to reduce noise without blurring the edges of relevant landmarks [15].

In the implementation and hardware used here, the two-frame motion estimation is not capable of processing matrix-sizes of 512x512 in real-time. To achieve real-time capable tracking results we have scaled down the input images to 30 % of their original size and applied the image processing, filtering and tracking on the down-scaled images. By this, it was capable to process up to 25 frames per second. The 30 % scaling has been chosen as a trade-off between calculation time and the potential risk of hiding small vessels. Additionally the bilateral filtering helps to keep small details visible. To obtain the motion vectors in the same scale as the original images, it was necessary to upscale both the matrix size and the length of the motion vectors according to the input scale factor. Upscaling the vector field matrix yields a more blurred motion vector field and compensates for local instabilities leading to a more robust tracking result with the risk of covering local motion.

The dense optical flow is sensitive to high-amplitude trajectories and out-of-plane motion. For certain cases this can result in flipped motion vectors around one or both axes. As respiratory motion has quasi-periodic trajectories it was possible to implement an on-the-fly outlier detection by evaluating the continuous dx-/dy evolution of the motion vectors at the given landmarks. This is done by comparing the current motion vector components to its predecessors and calculating the relative difference. If for any reason the dx-/dy component of the difference is exceeding the given outlier threshold (here 12 px) the current motion vector is considered an outlier and it is discarded. Instead, the previous motion vector is used, under the assumption that with a quasi-periodic trajectory the tracked landmark should again pass the current position at a later time. When plotting the trajectories over time the detected outliers appear as sensor-clipping. By applying real-time capable polynomial fitting (1^{st} order) to the outlier-filtered trajectories, it is possible to compensate for the clipping. To achieve non-linearity on a 1^{st} order polynomial fit, the fit is done on overlapping segments with a constant window-size (number of samples) and then averaging the accumulated fits for each sample. Another aim of the polynomial fitting is to provide smoother input for the prediction described in 2.5. The polynomial fitting tends to under-estimate the motion for the true extreme values but appears to correctly over-estimate the motion for the samples where the outlier detection causes the clipping of the trajectory component.

2.3 2D Motion Tracking on GPU

This heuristic procedure has several parameters that were tuned on the labeled 2D data provided for training. **Preprocessing:** the images were resized for speed such that the width is 160 pixels. Most of the features to track were vessels (roughly a black round area surrounded by a whiter tissue boundary), for which we try to get automatically the cross-section radius (r), in the first image. For cases that look differently, a default $r = 5$ pixels is used. The scale of the image is increased and the estimation repeated until $r \geq 2$. Each tracked point is treated independently, possibly even using the same images resized at a different scale. A mask is computed containing all pixels that change in a dataset, which corresponds to a device-specific viewport. It can be extracted from previous sessions and reused, or extracted from the first image automatically under mild assumptions. **Registration:** a quadratic patch with the edge size of $6r$ (to include the region of interest and a local neighborhood) is then extracted from the scaled first image and used as reference. From 3000 random patches with uniformly random variations of size ($\pm 10\%$) and random skewness ($0 \ldots 40\%$), distributed over the whole field of the current frame, the most similar is computed on GPU. The similarity function used is the minimum of the correlation coefficients of the entire patch and of the left/right/top/bottom parts with the corresponding parts of the reference. A second local search is then performed. This looks for the best variation (same ranges as above) of the patch found in the first pass, after a reduction of the edge's size to 80%. To avoid the flaws of this simple similarity function, a different one is used to evaluate further the best matching patch found - the correlation coefficient of the polar coordinates representation of the central inscribed disc of the patch. For both similarity functions the mask was taken into account in an attempt to improve the tracking behavior next to the boundary of the valid area. This is achieved by ignoring the computations of the pixels known to be outside of the valid area. **Postprocessing:** in our first GPU submission we have filtered the raw position guesses produced by the GPU registration using a simple jump detection, using as threshold $3r$. For a second GPU-based submission, we used the same input but a more refined filtering that looked both for sudden jumps in position and for sudden variations in similarity. The thresholds used were continuously automatically adjusted, assuming a normal distribution of the frame-to-frame distances and of the matching quality (thresholds set to 3 times the empirical standard variance of the populations corresponding to frames where the tracking is believed to be good).

2.4 3D Motion Tracking

The same method as in 2.2 has been used for the 3D tracking, which is here in fact a 2.5D approach. This was done by applying the tracking on the two orthogonal slices that intersect at the given annotation after rotating the volumes such that the first 2D coordinate lies in the XY-plane and the second one in the ZY-plane (depending on the alignment of the input data). This yields two separate tracking results per frame. As both slices share the Y-axis in 3D space, the tracking results for this redundant axis have been averaged for both results and we use the X- and Z-components independently as the final 3D position. As the orthogonal slices are fixed in their Z-coordinate, the method is sensitive to out-of-plane motion of the landmark to track. To compensate for out-of-plane motion it is necessary to adjust the Z-coordinate for one slice according to the X-component of the motion vector from the orthogonal plane (left as future work).

2.5 Prediction

For the 2D motion vectors, we tested a robust on-line prediction method that we developed for respiratory motion compensation (tested previously on the Cyberknife respiratory motion dataset[3]) after a preprocessing described in [13] at page 100, set here for predicting the position of the point of interest 200 ms into the future. As there is only one signal, like there was for the Cyberknife data, the problem is one of pure auto-regression. The algorithm we used is a linear auto-regression (AR). More precisely, we employed iterated stable linear regression (3 iterations, elimination of outliers at quantile 0.95). The auto-regression model was updated once per second, using only data not older than one minute. We used an order that corresponds to 4.5 seconds at 20 Hz sampling rate. From the history window, the AR model was built to depend only on the values $\{T + dt; dt \in \{0, -1, -2, -3, -5, -8, -13, -21, -34, -55, -89\}\}$, a Fibonacci progression that stops at 4.5 s into the past (for 20 Hz sampling rate) from the last known value T. The accuracy we obtained on the Cyberknife database using the Fibonacci auto-regression delays were comparable to the ones obtained using the full history window, but the speed was much increased by a factor of about 10. Less training data had to be collected, as there were less parameters to estimate, therefore the prediction could start earlier. As the current implementation of the prediction involves a learning phase of 30 seconds, we did not try to use this prediction for the 3D tracking due to the insufficient length of the datasets.

2.6 Software Tools and hardware

The methods described in 2.2 were implemented using the Mevislab software, Python, OpenCV and Numpy. The actual tracking and timing measurement has

[3] available at http://signals.rob.uni-luebeck.de/index.php/Signals_@_ROB, by courtesy of Dr. Kevin Cleary and Dr. Sonja Dieterich

been executed on a Intel Core i7-4770k with 32 GB RAM. The GPU method has been implemented in Matlab (CPU Intel Xeon E5540, 24 GB RAM) and CUDA (GPU Nvidia Geforce GTS 450).

3 Results

Tracking Results from Dense Optical Flow: given the fact that focused ultrasound treatment usually involves ablation of a safety margin around the tumor [16], the dense optical flow tracking turned out to work with the precision required for surgery despite the reduced resolution due to down-scaling the images. The mean tracking error (MTE) for the entire test-set of 54 points is 1.82 ± 2.37 mm. Only 5 out of 54 points yielded poor results with a mean error above 3 mm. The high-deviation results can be explained by out-of-plane motion causing the tracked landmark to change its shape or if the landmark is close to the border of the field of view in combination with high amplitude motion. This occasionally causes the motion vectors to flip around one axis for certain cases. The polynomial fitting has almost no impact on the overall outcome (1.82 ± 2.34 mm) and mostly helps to marginally improve results that already have low deviation. The average calculation time for all points is 40 ms depending on the amount of scaling.

GPU Tracking Results: the more refined, self-tuning outliers detection produced better results than the simple threshold detection. The mean tracking error it produced was 1.55 ± 2.78 mm for one 2D subset (outperforming the dense optical flow based method) and 2.40 ± 2.78 mm for the second one. The average processing time per frame was 250 ms (179 ms without using the advanced outliers detection), conditioned by the speed of the GPU.

3D Tracking Results: as the 3D tracking is lacking a Z-coordinate adjustment the results are suffering from out-of-plane motion. The MTE is 5.24 ± 4.34 mm for all 21 points across different datasets. One data subset has a significantly higher error of 7.61 mm. This can be explained by the lower and anisotropic voxel-resolution. As the tracking is performed on two orthogonal slices the average processing time per frame of 60.68 ms is slightly higher than in the 2D datasets but still below the 3D images' frame-rate (real-time) except for one data subset with temporal resolution of 24 Hz.

Prediction Results: the prediction has been executed on both the polyfitted and non-poly-fitted dense optical flow results, however only the prediction on the non-polyfitted results has been submitted for evaluation. The MAE for the prediction results is 2.04 ± 2.36 mm. Given that polynomial fitting has no impact on the quality of the dense optical flow tracking (see 3) it was possible to determine the difference (RMS) of the prediction on the polyfitted and non-polyfitted results. While the prediction on the non-polyfitted tracking results reproduces the high-frequency motion and leads to reduced precision of the prediction, the polynomial fitted trajectories decrease the RMS deviation for the prediction by 65 % for all points - Fig. 1(b)[4].

4 Discussion and Conclusion

As the result of the Dense Optical Flow tracking is a motion vector field for the entire image, multiple points/landmarks can be tracked in parallel without any additional processing time. If the expected trajectory of the landmark is known in advance, it is possible to apply the dense optical flow on a small patch of the input image. This can significantly reduce the processing time and allows to omit the scaling operation to preserve all details in the region of interest. To improve the 3D tracking, it is planned to adjust the slices' Z-coordinate according to the orthogonal X-motion vector. As the outliers from the tracking can be detected on-the-fly, there is a chance to improve the prediction by taking the outlier-indicators into account to reduce the influence of those samples on the prediction. For 2D tracking it is essential to minimize out-of-plane motion by proper alignment of the FOV orthogonal to the dominant axis of motion.

The GPU used is more than three years old and underpowered in comparison to the latest ones, which limited our options in compromising between the speed and the quality of the optimization. The speed of the GPU tracking is completely scalable and can be much higher (reaching real-time capabilities on preliminary tests with a more modern graphic card, Quadro K5000).

We have shown in this paper the results of two tracking algorithms in combination with auto-regressive prediction. While both tracking methods work with the precision required for surgery [16], there is a small advantage in precision for the GPU tracking. The dense optical flow tracking has an advantage in its capability to cope with high imaging frame rates and the additional benefit of being able to track multiple landmarks in the same image without any additional processing time. Even though the GPU tracking is not capable of handling the high frame rate of US imaging, it might be useful to combine it with the dense optical flow tracking and potentially predict intermediate positions from infrequent updates from both tracking methods. The runtime overhead for both the polynomial fitting and the prediction is negligible in comparison to the processing time for the tracking. Interestingly the polynomial fitting has significant influence on the prediction results. As the prediction results for the polynomial fitted tracking are almost indistinguishable from the polyfitted tracking results, the performance of the prediction solely depends on the the quality of its input data.

References

1. J. R. McClelland, D. J. Hawkes, T. Schaeffter, and A. P. King. Respiratory motion models: A review. *Medical image analysis*, 17(1):19–42, 2013.

[4] The MAE for the prediction on the polyfitted tracking is 1.85 ± 2.36 mm. This result is not part of the CLUST competition, being sent for evaluation after the challenge ended – but before the true labels for the test set to be provided to the participants.

2. JE. Kennedy, F. Wu, GR. Ter Haar, FV. Gleeson, RR. Phillips, MR. Middleton, and D. Cranston. High-intensity focused ultrasound for the treatment of liver tumours. *Ultrasonics*, 42(1):931–935, 2004.
3. C. Grozea, D. Lübke, F. Dingeldey, M. Schiewe, J. Gerhardt, C. Schumann, and J. Hirsch. ESWT-tracking organs during focused ultrasound surgery. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE, 2012.
4. M. Pernot, M. Tanter, and M. Fink. 3D real-time motion correction in high-intensity focused ultrasound therapy. *Ultrasound in medicine & biology*, 30(9):1239–1249, 2004.
5. R. J. Schneider, D. P. Perrin, N. V. Vasilyev, G. R. Marx, P. J. del Nido, and R. D. Howe. Real-time image-based rigid registration of three-dimensional ultrasound. *Medical image analysis*, 16(2):402–414, 2012.
6. 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*, pages 792–795, April 2013.
7. E.-J. Rijkhorst, I. Rivens, G. ter Haar, D. Hawkes, and D. Barratt. Effects of Respiratory Liver Motion on Heating for Gated and Model-Based Motion-Compensated High-Intensity Focused Ultrasound Ablation. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2011*, volume 6891 of *Lecture Notes in Computer Science*, pages 605–612. Springer Berlin Heidelberg, 2011.
8. 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*, 18(5):740–751, 2014.
9. V. De Luca, M. Tschannen, G. Székely, and C. Tanner. A Learning-Based Approach for Fast and Robust Vessel Tracking in Long Ultrasound Sequences. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2013*, volume 8149 of *LNCS*, pages 518–525. Springer Berlin Heidelberg, 2013.
10. J. Schwaab, M. Prall, C. Sarti, R. Kaderka, C. Bert, C. Kurz, K. Parodi, M. Günther, and J. Jenne. Ultrasound tracking for intra-fractional motion compensation in radiation therapy. *Physica Medica*, 2014.
11. M. Prall, R. Kaderka, N. Saito, C. Graeff, C. Bert, M. Durante, K. Parodi, J. Schwaab, C. Sarti, and J. Jenne. Ion beam tracking using ultrasound motion detection. *Medical physics*, 41(4):041708, 2014.
12. R. Dürichen, T. Wissel, and A. Schweikard. Optimized order estimation for autoregressive models to predict respiratory motion. *International Journal of Computer Assisted Radiology and Surgery*, 8(6):1037–1042, 2013.
13. F. Ernst. *Compensating for Quasi-periodic Motion in Robotic Radiosurgery.* Springer, New York, December 2011.
14. G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. Springer, 2003.
15. C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, Sixth International Conference on*, pages 839–846. IEEE, 1998.
16. Y.-S Kim, H. Trillaud, H. Rhim, H. K. Lim, W. Mali, M. Voogt, J. Barkhausen, T. Eckey, M. O. Köhler, B. Keserci, et al. MR Thermometry Analysis of Sonication Accuracy and Safety Margin of Volumetric MR Imaging–guided High-Intensity Focused Ultrasound Ablation of Symptomatic Uterine Fibroids. *Radiology*, 265(2):627–637, 2012.

MICCAI CLUST 2014 - Bayesian Real-Time Liver Feature Ultrasound Tracking

Sven Rothlübbers[1], Julia Schwaab[2], Jürgen Jenne[1], Matthias Günther[1]

[1] Fraunhofer MEVIS, Bremen, Germany
[2] Mediri GmbH, Heidelberg, Germany

Abstract. We present the implementation of a Bayesian algorithm for tracking single features throughout ultrasound image sequences, with a focus on real-time applicability. After introducing the general concept of the algorithm, we suggest a sparse description of the target object to allow for rapid computation and semi-automatic target initialization. In 2D and 3D single feature tracking scenarios of the MICCAI challenge for liver ultrasound tracking (CLUST) 2014 we evaluate the algorithm and find mean tracking times of 1.25ms (2D) and 46.8ms (3D) per frame with mean tracking errors of 1.36mm (2D) and 2.79mm (3D).

Keywords: medical imaging, ultrasound, tracking, particle filter

Introduction

Ultrasound imaging offers the opportunity to generate image streams with high frame rates, allowing to track the motion of features for various purposes in medical applications. For real-time applications, the image stream has to be analyzed sufficiently fast and reliably[4, 5]. Particle filter algorithms[1], being capable of handling multiple hypotheses about a target's position, have already been applied successfully[2, 3, 6]. Their performance strongly depends on the quality of the target description. We propose a sparse but sufficiently precise description model, which will allow for real-time applications as well as semi-automatic target initialization.

1 Materials and Methods

Conditional Density Propagation Algorithm A tracking problem may be approached by describing the evolution of a probability density function within the image stream. The density function is represented by a set of samples or particles describing possible states of the target. While tracking, it is continuously updated by estimations and observations. Here, the system state is modeled by independent states defining the N_D independent degrees of freedom. Propagation of states is given by the Markovian assumption that the succeeding state x_{t+1}^d only depends on the current x_t^d instead of all possible predecessors $\boldsymbol{x_t^d}$.

$$p(x_{t+1}^d|\boldsymbol{x_t^d}) = p(x_{t+1}^d|x_t^d) \qquad (1)$$

Stochastic Estimation Model Lacking knowledge about the degrees of freedom or their limitations, we apply a simple stochastic model incorporating drift towards a mean state and random diffusion. The states of different degrees of freedom d are considered independent of each other.

$$p(x_{t+1}^{d}|\boldsymbol{x_t^d}) = \langle x^d \rangle_s + S_0^d \left[x_t^d - \langle x^d \rangle_s \right] + S_1^d \eta \tag{2}$$

The term S_0^d determines drift towards the current mean state, averaged over all samples $\langle x^d \rangle_s$ while the random diffusion term S_1^d sets the strength of a Gaussian random variable η.

Transformation Model Local features exhibit only few degrees of freedom and allow considering rigid transformations only. A transformation model featuring rotation and scaling around a center of mass and translation is chosen.

$$T(s_j) = T_{trans}(s_j) T_{rot}(s_j) T_{scale}(s_j) \tag{3}$$

The transformation matrix $T(s_j)$ translates $N_d = 5$ (T_x, T_y, S_x, S_y, R_z) or $N_d = 9$ (...,T_z, S_z, R_x,R_y) independent degrees of freedom - given by samples s_j - into a transformation matrix which transforms points from observation model space to image space.

Observation Model Real-time applications require a sparse, yet precise description of the target feature. The observation model describes the feature to be tracked and, given a position guess, returns a quality value to that guess. We describe the target feature, a liver vessel for instance, by a set of points with associated descriptors for brightness and darkness.

The descriptors define a local contrast - dark and bright regions of the local feature: Each point $\boldsymbol{r_i}$ in the model is assigned a likelihood of belonging to the dark (p_i^{drk}) and the bright(p_i^{brt}) part of the feature, which later will be derived from absolute brightness values b_i. In order to describe a relative contrast, values are kept normalized over all points (N_P):

$$\sum_{N_p} p_i^{drk} = 1 = \sum_{N_p} p_i^{brt} \tag{4}$$

The quality of a position guess, given by a sample s_j's transformation matrix $T(s_j)$ and the current image b, can be estimated by applying a weighting function such as:

$$w'(s_j) = \sum_{i=1}^{N_p} \left[p_i^{brt} - p_i^{drk} \right] \cdot b\left(T(s_j)\boldsymbol{r_i}\right) \tag{5}$$

For one sample s_j all observation points $\boldsymbol{r_i}$ are transformed into the image with the same transformation matrix $T(s_j)$. Each point i is transformed to its position $T(s_j)\boldsymbol{r_i}$ and has an effective weight $p_i^{eff} = p_i^{brt} - p_i^{drk}$ which may be positive or negative. If the point is expected to be bright ($p_i^{eff} > 0$) and found bright ($b(T(s_j)\boldsymbol{r_i})$ high), this will *increase* the weight $w'(s_j)$. Similarly, if the point is expected dark ($p_i^{eff} < 0$) and found dark ($b(T(s_j)\boldsymbol{r_i}) \approx 0$) this will *not decrease* the weight. In cases the brightness is not as expected, the weight will *not be increased* or even *decreased* respectively, returning a lower weight $w'(s_j)$ for the sample. In the presented algorithm, the final weighting function is set to

$$w(s_j) = \Theta(w'(s_j))w'^2(s_j). \tag{6}$$

Weights are interpreted as relative probabilities for re-sampling and thus can't be negative[3]. Emphasizing samples with higher weight, taking the power of two, shows to increase tracking performance.

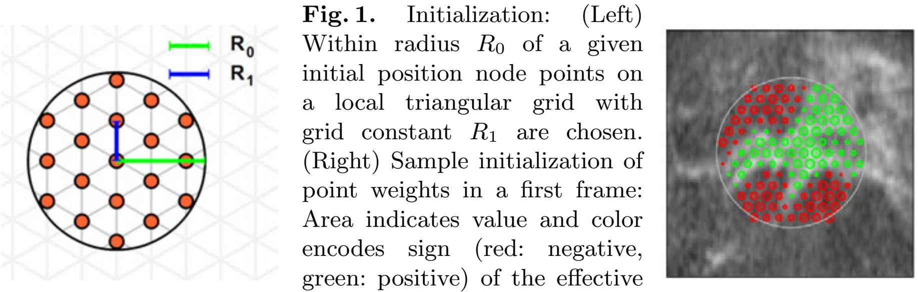


Fig. 1. Initialization: (Left) Within radius R_0 of a given initial position node points on a local triangular grid with grid constant R_1 are chosen. (Right) Sample initialization of point weights in a first frame: Area indicates value and color encodes sign (red: negative, green: positive) of the effective weight p_i^{eff}.

Observation Model Initialization The proposed definition of contrast might be applied to the whole target region, taking every pixel into account. As redundancies can be expected, it is assumed that not the whole target region needs to be stored in the observation model and that it suffices to hold only a few sampling points. A gain in computational speed is the immediate advantage, but the choice of a proper sub-sampling in the region is important. Here, the most simple assumption is explored:

The region of interest is sampled with a uniform triangular (2D) or tetrahedral (3D) grid (fig. 1) to cover space optimally. The two parameters of this grid are the grid radius R_0 around the target position and the grid edge length R_1, describing the distance of adjacent points. The observation model is initialized from the first frame of the sequence and the given target position vector. The brightness values b_i at the initial grid points are used to set the likelihood for brightness and darkness for each observation model point

$$p_i^{brt} \propto (b_i - b_{min}) \qquad\qquad p_i^{drk} \propto (b_{max} - b_i) \tag{7}$$

where b_{max} and b_{min} are maximal and minimal brightness among all points.

[3] Using formula 5 only, they might however appear if the observation is taken at a position which shows inverted brightness values to the target region. The Heaviside function $\Theta(x)$ sets negative weights to zero, excluding the affected sample from re-sampling.

Robustness Against Lag The single position value, returned from the probability density function given by all samples, is the observation model's geometric center averaged over all samples. When rapid motion has to be tracked, the probability density function may spread out and the mean may be left behind leading to visible lag. As precision is considered more important than computational speed some computational power is used execute multiple tracking steps in one frame, denoted as tracking repetitions F_T.

Data Data for performance evaluation is given by the MICCAI CLUST challenge as 2D or 3D liver ultrasound sequences. The 2D sets feature spatial resolutions of 0.36mm-0.55mm in 2427 up to 14516 frames per set. The 3D sets have resolutions of 0.308mm × 0.514mm × 0.6699mm (ICR), 0.7mm isotropic (SMT), 1.144mm × 0.594mm × 1.193mm (EMC) with 54-159 frames per sequence. For each sequence one or more target annotations are given for the first frame, indicating the features to be tracked. The remaining position sequence is to be generated by the tracking algorithm.

Setup Image information of the first frame, the initial position and additional tracker description parameters - region size and resolution - are used to initialize the target representation of the tracker. Additionally, the estimation model is set to constant drift and diffusion terms for all degrees of freedom[4]. Finally, the number of samples N_S and tracking repetitions F_T are set.

Code Execution The core source code for the algorithm is written in C++ and integrated into a module for the image processing and visualization framework MeVisLab (MeVis Medical Solutions, Bremen, Germany). This framework was used for the high level evaluation routines using Python scripts. The code was executed single threaded on a Windows 7 machine with an Intel Core i7-2600 CPU @ 3.4GHz and 32GB RAM.

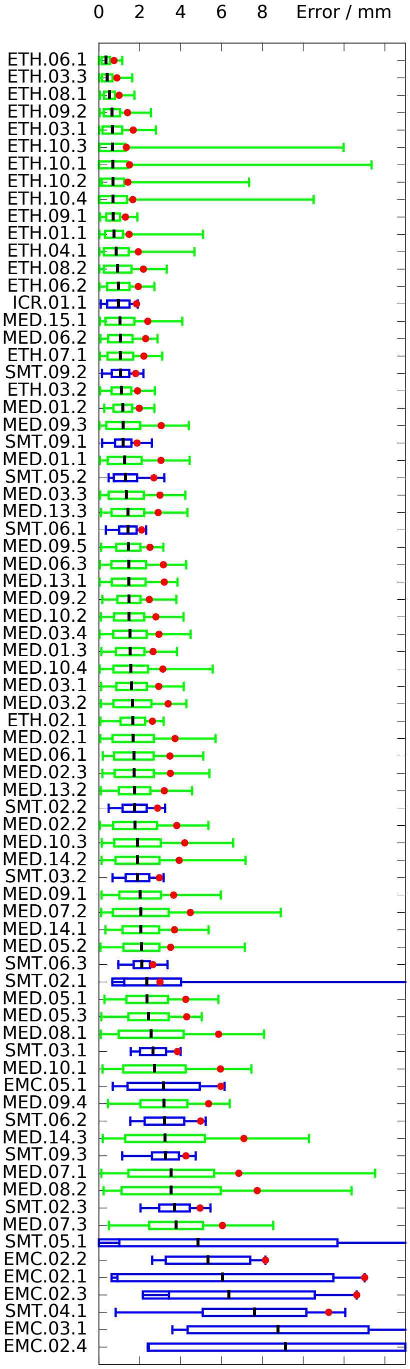
Performance Considerations For each frame computation time is constant, as the amount of computations needed is fixed. Most of the computation is spent for transforming positions for each sample and each point in the observation model. Main contribution of computation time of tracking is given by

$$T_C = C_0 N_s N_p F_T \tag{8}$$

with sample count N_s, point count N_p, tracking repetitions F_T and machine dependent proportionality constant C_0. Using a sparse observation model with low N_p can lead to lower computational cost, but may introduce uncertainty. Similarly, there is a trade-off between precision and speed involved when changing the number of samples N_s. For the challenge, values which allow for fast and reproducible results are explored.

[4] In the presented results, drift terms are set to 1, meaning that no drift is considered. Also, as naturally no rotation and only little scaling are expected of small liver features, we neglect rotation and scaling, setting them to 0. Translation is set isotropic.

2 Results



Data	**Settings**						**Time** / ms	
	S_1^{Tr}	R_0	R_1	N_P	N_S	F_T	t_d	t_f
MED	3.3	26	5.0	117	346	1.6	54.6	1.22
ETH	2.9	18	2.7	172	200	2.0	60.5	1.33
2D	3.2	24	4.3	134	300	1.7	56.4	1.25
ICR	1.0	15	1.6	4735	100	4	41.7	36.2
EMC	1.0	14	1.6	3344	583	4	166.7	121.2
SMT	1.0	11	1.8	2141	129	4	125	15.6
3D	1.0	12	1.7	2608	257	4	122	46.8

Table 1. Mean settings and tracking times for the datasets: Isotropic diffusion of translation (S_1^{Tr}) in arbitrary units. Grid distances R_0, R_1 in voxels and the resulting number of points N_P in the observation model. Number of samples N_S and tracking repetitions F_T. Duration of a frame in the sequence $t_d = 1/FPS$ and measured tracking time per frame t_f.

Data	**Tracking Error** / mm				
	MTE	**SD**	**95%**	**min**	**max**
MED	1.93	1.32	4.48	0.02	13.52
ETH	0.77	0.59	1.85	0.00	13.35
2D	1.36	1.17	3.61	0.00	13.52
ICR	0.95	0.55	1.84	0.09	1.90
EMC	6.28	4.49	14.20	0.68	19.33
SMT	2.70	2.62	7.91	0.15	24.70
3D	2.79	2.74	8.35	0.09	24.70

Table 2. Resulting tracking error averaged over data sets: Mean tracking error (MTE), standard deviation of error (SD), minimum and maximum error (min, max) and 95th percentile. Depicted in more detail in figure 2.

Fig. 2. Distribution of results presented in table 2: Mean (black), standard deviation (box), minimum and maximum error (whiskers) and 95th percentile (red dot) for 2D (green) and 3D (blue) sets. All sets are sorted by their mean performance. The noticeable outliers of set ETH-10 are related to a single frame irregularity in the sequence.

Comparison to Ground Truth The difference between tracking result and ground truth of the challenge was evaluated in several categories (fig. 2, tab. 1 & 2). The 2D sets (fig. 3) exhibit mean errors of 1.93mm (MED) and 0.77mm (ETH). In total, the mean error is 1.36mm with a standard deviation of 1.17mm. Largest errors were caused by a target region including two targets which later move apart (MED-07_1) or vessels changing shape (MED-07_3, also fig. 4). Set ETH-10 shows an irregular frame (03598) causing a temporary deviation, but not affecting the overall tracking performance.

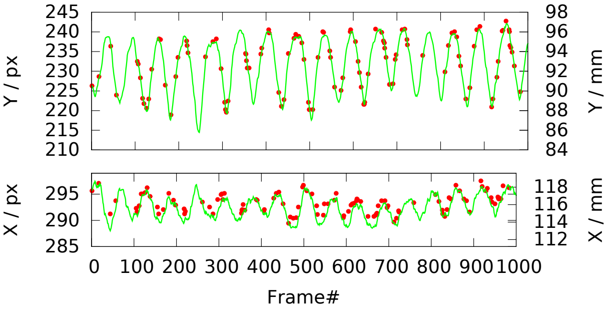
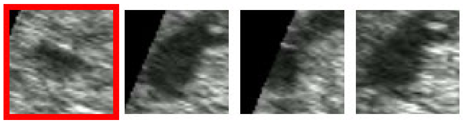


Fig. 3. Sample run on training case ETH-05_2: $S_1^{Tr} = 1$, $R_0 = 13$,$R_1 = 2$. Tracking result (green) and ground truth (red dots).

The straightforward extension of the 2D tracking algorithm to 3 dimensions shows mean errors of 0.95mm (ICR), 2.70mm (SMT) and 6.28mm (EMC). Larger errors in the SMC dataset are related to a target disappearing on the border of the volume (SMT-05_1), and a dataset in which the target region lacks a unique local contrast (SMT-04_1). Similarly, in the EMC sets, the definition of a suitable target region is difficult due to low resolution images and relatively large (non-local) features.

Fig. 4. Sample images of a difficult training sequence (ETH-04_3) in which the target changes the original shape (red) and repeatedly leaves the field of view.



Generally, minimal errors could be achieved if the target feature showed a distinct pattern and strong contrast. Arteries, exhibiting bright borders, could be tracked more reliably than veins with less local contrast. Smaller features returned better results as they fit the assumption of locally rigid transformations.

Two dimensional features changing shape locally (fig. 4) indicate out of frame motion and may be difficult to track for the algorithm. A global change in contrast, however, can be handled by the algorithm as it relies on relative instead of absolute brightness values.

If the observation model includes structures not belonging to the target, like the diaphragm or out-of-volume area, this may spoil tracking performance. While the former can only be dealt with by careful choice of targets, the latter might be handled automatically by a future algorithm.

Computational Speed The presented algorithm is preliminary. Even though it is applicable in or close to real-time it may be further optimized for speed. Not all of its computations are suited for parallelization but the crucial ones, discussed in equation 8, are in particular. A noticeable gain in performance can be expected from a GPU-based acceleration.

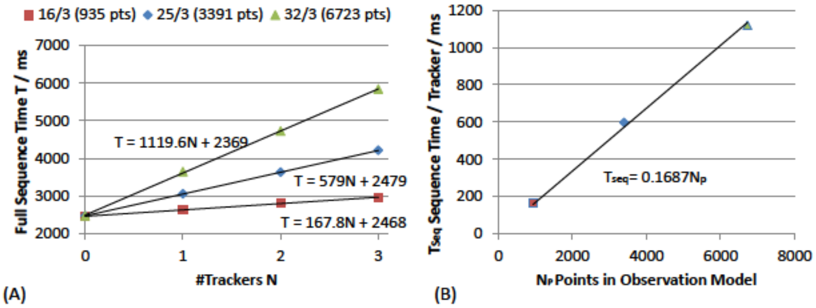


Fig. 5. Performance evaluation on 3D ultrasound data (SMT-02): (A) Time needed to track a 50 volume sequence using 0 to 3 trackers with $N_S = 200$, $F_T = 1$ and differently sized observation models ($R_0 = 16,25,32$, $R_1 = 3$). The time offset of approximately 2350ms is related to volume loading time. The slope indicates actual computation time. (B) Actual tracking time against number of points in observation model. Tracking one point with 200 samples over 50 frames takes around 0.17 ms, yielding $C_0 = 17$ns in equation 8.

Image loading takes a large portion when tracking a sequence, especially when 3D images are loaded. When running multiple trackers in parallel, loading the image in only done once. Acquiring data with variable number of trackers (fig. 5A) leads to a T-intercept representing the image loading time and a slope representing the actual computation tracking time needed per tracker. Using equation 8, a computational cost of $C_0 \approx 17\text{ns} \pm 1\text{ns}$ for a single point, sample, frame and repetition can be computed for the utilized machine (fig. 5B). Such a value may be helpful to estimate the actual computational time and also to balance possible trade-offs when initializing the tracker.

Depending on the settings (tab. 1), the tracking algorithm requires between 0.5ms and 2.5ms to run on a single 2D frame with 1.25ms on average. Merely one case with a high resolution observation model required 6ms. In three dimensions the setup of the observation model has a lot more influence on the required time. Very distinguishable targets may be computed in 1ms - 10ms while the choice of larger regions may lead to times in the order of 30ms or even 300ms.

Discussion We presented an implementation of a Bayesian tracking algorithm which supports semi-automatic initialization and which is able to follow target features fast and precisely. For the MICCAI Challenge on Liver Ultrasound Tracking it was evaluated using the 2D and 3D data sets of the challenge. Tracking was performed with average run-times of 1.25ms±0.82ms/frame in 2D and

1ms - 372ms/frame in 3D. Compared to the challenge's ground truth, 2D and 3D tracking results exhibited mean errors of 1.36mm and 2.79mm respectively, which showed to depend on the data set group or ultrasound device the data was recorded with.

The proposed algorithm shows to work reliably, yet there are ways to optimize it. The performance was found to be independent over a wide range of parameters, but emphasis to either speed or precision may be given by setting the number of samples or resolution of the model. A sparse observation model was applied by under sampling the target region with a local grid without any further information. Deciding which points of the region are actually important for the algorithm by a more elaborate algorithm could help improve efficiency much further - especially in three dimensions.

In conclusion, with the proposed algorithm results could be generated in real-time, by using a simple sparse target representation. Although the results showed high precision in 2 dimensions already, by using a more sophisticated observation model, the algorithm may be improved much further for the 3D case in the future.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n 611889 and was supported by the Fraunhofer Internal Programs under Grant No. MAVO 823 287.

References

1. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. Computer Vision ECCV '96, Springer Berlin Heidelberg
2. Zhang, X., Günther, M., Bongers, A.: Real-Time Organ Tracking in Ultrasound Imaging Using Active Contours and Conditional Density Propagation.
3. Feinberg, D. A., Giese, D., Bongers, D. A, Ramanna S., Zaitsev M.,Markl M., Gnther, M.: Hybrid ultrasound MRI for improved cardiac imaging and real-time respiration control. Magn Reson Med, 2010. 2009 Wiley-Liss, Inc., 290-296
4. Hsu, A., Miller, N.R., Evans, P.M., Bamber, J.C., Webb. S.: Feasibility of using ultrasound for real-time tracking during radiotherapy., Med Phys. 2005 Jun;32(6):1500-12.
5. De Luca, V., Tschannen, M., Székely, G. , Tanner, C.: A Learning-Based Approach for Fast and Robust Vessel Tracking in Long Ultrasound Sequences Medical Image Computing and Computer-Assisted Intervention - MIC- CAI 2013, Lecture Notes in Computer Science, vol. 8149, pp. 518–525 (2013)
6. Günther, M., Feinberg, D. A.: Ultrasound-guided MRI: Preliminary results using a motion phantom. Magnetic Resonance in Medicine 2004-1, 27-32
7. Zhang, X., Schönberg, S.: Development of 3D Real-time Ultrasound Tracking Methods for Motion Compensation, Dissertation, Ruprecht-Karls-Universität Heidelberg, Medizinische Fakultät Mannheim

Live Feature Tracking in Ultrasound Liver Sequences with Sparse Demons

Oudom Somphone, Stéphane Allaire, Benoit Mory, and Cécile Dufour

Medisys Lab, Philips Research

Abstract. We describe methods for feature tracking in temporal image sequences, based on a motion estimation framework called S*parse Demons*. It relies on a Gaussian-convolution model of the deformation field; this model is embedded in a variational formulation with a cost function defined on a finite number of points of interest. The resulting algorithm is fast and suitable for real-time, live feature tracking. Our methods are evaluated on the CLUST'14 database, consisting in 2D and 3D ultrasound liver sequences with landmarks or areas to be tracked.

1 Introduction

In this paper, we present methods for automatic tracking of anatomical features in the liver, in 2D and 3D ultrasound sequences. We apply our methods to the database of the MICCAI CLUST'14 challenge[1]. The features to track are landmarks and regions placed at locations of interest such as portal or hepatic veins bifurcations and tumors. The applicative scenario of the CLUST challenge is intervention and therapy in the liver under real-time ultrasound image guidance. More specifically, we address the issue of real-time compensation of the respiratory motion in the liver. To this end, we propose fast methods that do not rely on access to "images from the future". Moreover, we assume that:

- The ultrasound probe, be it 2D or 3D, does not drastically move during the acquisition.
- The acquisition frame rate of the ultrasound system is high, so that the motion between two consecutive frames is limited to a few millimeters.

For both the 2D and 3D datasets, we use a common motion estimation framework called *Sparse Demons*, described in section 2. Different strategies are adopted according to the objects to track – landmarks or regions, and the nature and quality of the datasets – 2D or 3D, with or without gain control. In 2D (sections 3 and 4), out-of-plane motion is expected, so that the method should be robust to appearing and disappearing features. In 3D (section 5), we designed anti-drift strategies based on the assumption that the respiratory motion is periodic. The results of our methods on the CLUST datasets were evaluated by the organization committee, based on a ground truth made of manual annotations.

[1] http://clust14.ethz.ch/

2 Sparse Demons

Feature tracking along a temporal sequence is regarded as a succession of reference-to-template motion estimation problems; at each incoming template frame the new positions of the tracked features are obtained in a causal manner by propagating the reference positions according to the estimated displacement. *Sparse Demons* is a variational approach to solve each reference-to-template problem. The key idea of our method is to find an optimal dense, non-rigid displacement field by minimizing an energy E defined only on a finite number of points of interest $\{\mathbf{x}_i \mid i \in \mathcal{P}\}$:

$$E = \sum_{i \in \mathcal{P}} \int_{\Omega} \delta(\mathbf{x} - \mathbf{x}_i)\ \mathcal{D}\Big[R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big]\ d\mathbf{x} \tag{1}$$

where R and T are the reference and template images respectively, Ω is the image domain and δ is the Dirac function. $\mathcal{D} : \mathbb{R} \to \mathbb{R}$ is a function that penalizes the dissimilarity between the reference and the transformed template; for instance, $\mathcal{D}(x) = x^2/2$ was used in [1]. As for the displacement field, we adopt a fluid-like regularization, which can be approximated by Gaussian filtering [2]; in this model, $\mathbf{u}$ is assumed to be the result of the convolution of an auxiliary field $\mathbf{v}$ with a Gaussian kernel ω_σ of scale σ:

$$\mathbf{u}(\mathbf{x}) = [\omega_\sigma * \mathbf{v}]\,(\mathbf{x}) = \int_{\Omega} \omega_\sigma(\mathbf{x} - \mathbf{y})\mathbf{v}(\mathbf{y})\ d\mathbf{y} \tag{2}$$

where $\omega_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2}\ e^{\frac{\|\mathbf{x}\|}{2\sigma^2}}$.

Minimizing E w.r.t. $\mathbf{v}$ is done by gradient descent; calculus of variations results in the following evolution equation:

$$\frac{\partial \mathbf{v}}{\partial t} = -\nabla_{\mathbf{v}} E = -\omega_\sigma * \left(\sum_{i \in \mathcal{P}} \delta_i \nabla_{\mathbf{u}} E \right) \tag{3}$$

where $\delta_i(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_i)$ and $\nabla_{\mathbf{u}} E$ is the dense gradient of E w.r.t. $\mathbf{u}$:

$$\nabla_{\mathbf{u}} E(\mathbf{x}) = -\mathcal{D}'\Big[R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big]\nabla T(\mathbf{x} + \mathbf{u}(\mathbf{x})) \tag{4}$$

The subsequent algorithm (see below) has similarities with the *demons* algorithm [3, 4]. Its computational complexity is however lower since image forces are not computed in the whole image domain but only at points $\mathbf{x}_i$.

The following sections describe how we used this general image pair registration framework to track features in a causal manner along 2D and 3D ultrasound liver sequences from the CLUST database. One of the values of this database is to provide long sequences over many breathing cycles, which are challenging for simple $(t-1)$-to-t estimation schemes as errors accumulate and cause inevitable drifting. For every subclass of the database, we investigated several anti-drift mechanisms in a live tracking scenario. In each section, we specify the point inputs, the dissimilarity measure, the respective expression of the subsequent dense energy gradient (4), and propose ways to avoid drifting.

Algorithm 1: Sparse Demons - Gradient Descent

```
Set $k = 0$ and $\mathbf{v}^0 = \mathbf{0}$
repeat
    Compute $\mathbf{u}^k = \omega_\sigma * \mathbf{v}^k$
    for all $\mathbf{x}_i$ do
        Interpolate $T(\mathbf{x} + \mathbf{u}^k(\mathbf{x}_i))$ and $\nabla T(\mathbf{x} + \mathbf{u}^k(\mathbf{x}_i))$
        Compute $\nabla_{\mathbf{u}^k} E(\mathbf{x}_i)$ according to (4)
    end
    Smooth the result to obtain the incremental update
    $\delta \mathbf{v}^k = -\omega_\sigma * \left( \sum_{i \in \mathcal{P}} \delta_i \nabla_{\mathbf{u}^k} E \right)$
    Update $\mathbf{v}^{k+1} = \mathbf{v}^k + \delta t . \delta \mathbf{v}^k$
    $k = k + 1$
until steady state;
```

3 Landmark Tracking in 2D

3.1 Method

In these datasets ("ETH" and "MED"), the gray values are consistent along the sequences. We therefore use the squared difference as dissimilarity measure, *i.e.* $\mathcal{D}(x) = x^2/2$, which yields the following energy gradient:

$$\nabla_{\mathbf{u}} E(\mathbf{x}) = -\Big[R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big] \nabla T(\mathbf{x} + \mathbf{u}(\mathbf{x})) \tag{5}$$

At each frame, the reference points of interest $\mathbf{x}_i$ are chosen in the neighbourhood of the landmarks, based on the amplitude of the image gradient: pixels on edges are selected and those in the flat regions are discarded (see Fig. 1). The neighbourhoods are squares of size Δp, centered on the landmarks.

The tracking consists of two phases:

Initial $(t-1)$-to-t tracking From frame 1 to τ (typically 100), the template is the incoming frame t and the reference is the previous frame $(t-1)$. During this phase, a mean *reference patch* is concurrently built around each initial landmark. Each reference patch consists in a small square image of size Δp, obtained by summing the patches centered on the corresponding landmark's positions at every frame (Fig. 2(b)).

1-to-t patch registration From frame $\tau+1$ onwards, for each incoming frame, we register *template patches* around the current landmark positions (Fig. 2(c)), towards the corresponding reference patches, which yields the position correction from frame $(t-1)$ to t. The aim of this scheme is to prevent drift by error accumulation, that inevitably occurs with a $(t-1)$-to-t scheme when the sequence is long. Moreover, to prevent one drifting landmark from influencing the others, we track each landmark independently.

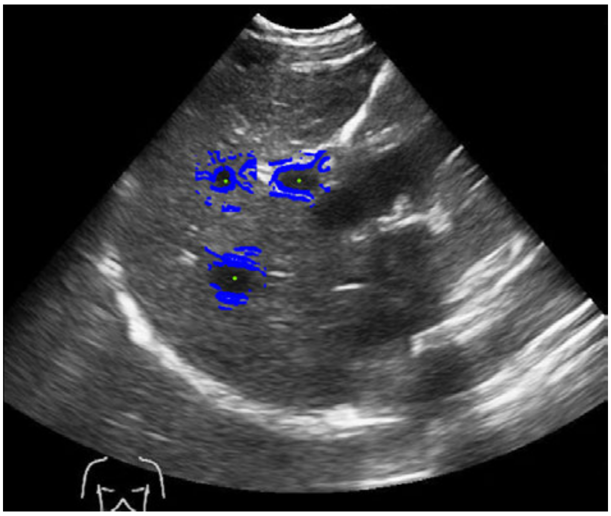
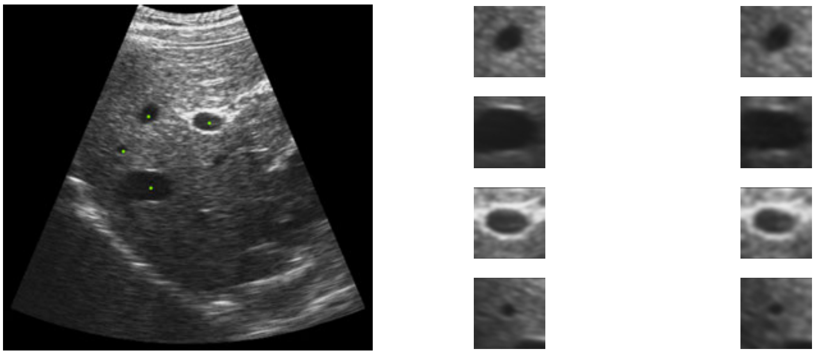


Fig. 1. Reference points (shown on a frame from the MED-13 sequence): the selected points of interest $\mathbf{x}_i$ (blue) are the pixels with larger image gradient in the neighbourhood of the current landmarks (green).



(a) Frame 19 of MED-03 (b) Reference patches (c) Template patches

Fig. 2. Reference and template patches in 2D landmark tracking.

3.2 Results

The test database for this part of the challenge contains 21 sequences of several thousands of frames (from 2424 to 14516) and 1 to 5 landmarks to be tracked. The parameters were tuned to $\sigma = 30$ mm, $\tau = 100$ frames and $\Delta p = 30$ pixels. The tracking error is defined as the Euclidean distance to the manually annotated ground truth. Table 1 below displays mean errors for all landmarks of all sequences for the "ETH" and "MED" datasets.

Dataset	MTE	SD	min	95%	max
ETH	0.98	1.14	0.00	2.45	24.16
MED	2.48	3.59	0.02	6.89	38.88
All2Dpoints	1.74	2.78	0.00	4.67	38.88

Table 1. Results for 2D landmark tracking. Mean Tracking Errors (MTE), Standard Deviations (SD), mininum errors (min), 95^{th} percentiles (95%) and maximum errors (max) are given in [mm].

4 Segmentation Tracking in 2D

4.1 Method

These sequences display some large intensity changes from one frame to the next and using the sum of squared difference as dissimilarity measure is not suitable. Instead, we minimize the entropy of the difference between the reference and the transformed template, which yields the energy:

$$E = -\int_{\mathbb{R}} p_{\mathbf{u}}(a) \log\left(p_{\mathbf{u}}(a)\right) da \tag{6}$$

$p_{\mathbf{u}}$ is the continuous Parzen estimate of the probability density function of the image difference over the the points of interest:

$$p_{\mathbf{u}}(a) = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \int_{\Omega} \delta(\mathbf{x} - \mathbf{x}_i)\ K\Big(R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - a\Big) d\mathbf{x} \tag{7}$$

where K a smooth non-negative normalized Gaussian kernel. Calculus of variations results in the following dense energy gradient:

$$\nabla_{\mathbf{u}} E(\mathbf{x}) = -\left[K * \frac{p'_{\mathbf{u}}}{p_{\mathbf{u}}}\right] \Big(R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big) \nabla T(\mathbf{x} + \mathbf{u}(\mathbf{x})) \tag{8}$$

Like in the previous section, the points of interest $\mathbf{x}_i$ are selected in the neighbourhood of the segmentation boundary, based on the amplitude of the image gradient. Since these sequences are short, the strategy to process the sequence is the $(t-1)$-to-t scheme.

4.2 Results

The test database for this part of the challenge contains 7 short sequences (from 51 to 105 frames) and 1 or 2 areas to be tracked. The Gaussian scale of the displacement was tuned to $\sigma = 30$ mm. The tracking is evaluated through the Dice coefficient between the tracked area and the manually segmented ground truth (Table 2).

Dataset	**MDice**	**SD**	**min**	**max**
OX-01_1	86.76	5.46	74.25	96.54
OX-02_1	85.66	4.99	73.25	97.74
OX-04_1	91.43	6.57	47.22	97.66
OX-05_1	79.93	6.71	61.79	95.84
OX-06_1	76.93	9.36	53.55	94.17
OX-07_1	89.71	4.39	72.41	97.74
OX-07_2	77.42	5.25	67.39	94.42
OX-08_1	88.75	2.83	79.19	98.27

Table 2. Results for 2D area tracking. Mean Dice (MDice), Standard Deviations (SD), minimum (min) and maximum (max) are given in [%].

5 Landmark Tracking in 3D

5.1 Method

In these datasets ("ICR", "SMT", and "EMC"), the gray values are consistent along most sequences. We therefore use the squared difference as dissimilarity measure (5), like in section 3. At each frame, the reference points of interest $\mathbf{x}_i$ are chosen in the neighbourhood of the landmarks on a square grid of size 80mm, regularly spaced by 10mm. Besides more sophisticated ultrasound shadow detectors, points are simply discarded in the darkest regions. The baseline is a $(t-1)$-to-t tracking, where all points of interest in all neighbourhoods are tracked together at once. To prevent drifting, an additional 1-to-t tracking is enabled if any of the two following triggers occurs:

Close histogram trigger From analyzing the differences between image histograms w.r.t. the reference difference level computed between the first two frames of the sequence, we can detect that the current incoming frame is close (not exceeding more than 20% of the reference difference level) in appearance to the initial frame where source annotations were given, which implies that a direct registration shall succeed.

Close location trigger From trajectory analysis, when landmarks positions get close (below 1.8 mm) to the positions of the source annotations given in the first frame, we also deem that a direct registration shall succeed. This trigger relies on the assumption that the probe is not moved during the sequence acquisition.

If triggered, the 1-to-t tracking overrides the $(t-1)$-to-t tracking.

5.2 Results

The database for the 3D landmark data class of the challenge contains 10 sequences of 54 to 159 frames and 1 to 4 landmarks to be tracked. The tracking error is defined as the Euclidean distance to the manually annotated ground truth. Table 3 below displays errors for all landmarks of all sequences per institution.

Dataset series	MTE	SD	min	95%	max
ICR	3.20	2.50	0.58	7.06	7.17
SMT	2.66	2.57	0.26	8.44	16.61
EMC	5.67	5.16	0.41	16.68	17.49
All3Dpoints	2.78	2.72	0.26	9.20	17.49

Table 3. Results for 3D landmark tracking. Mean Tracking Errors (MTE), Standard Deviations (SD), minimum errors (min), 95^{th} percentiles (95%) and maximum errors (max) are given in [mm].

6 Discussion and Conclusion

In terms of run-time estimation, it has to be noted that for both 2D and 3D, the tracking process is today somehow "irregular". Indeed, in 2D for instance, each given feature is first processed during a training period, then a different tracking can start. Likewise in 3D, the anti-drift strategy triggers additional motion estimations on an unsystematic basis. Also, the processing time depends on the number of features to be tracked. As the applicative scenario of the contest is not completely defined (total number of features to be tracked, adding features one by one as incoming frames flow, desired accuracy, etc.), the technical approach is not finalized. We report orders of magnitude of the computational load of these methods, taking into account that they have not been specifically optimized to that respect. On a multithreaded PC platform, the following frame rates are obtained with the described methods: around 40Hz in 2D, and around 10Hz in 3D.

The methods described in this paper are dedicated to live real time ultrasound. They are still under development, and the CLUST contest greatly helps in the design of the suitable approach and technologies. At testing and improving our methods on this challenging data, we stuck to the applicative scenario of a causal system. The progressive evolution of the tracked features been under scrutiny, and a final success criterion has been: whether the tracked features visually drift before the end of the sequence, and in that case whether the anti-drift mechanisms get them back on track w.r.t. the image content of the last frames. This is complementary to the criteria of overall average agreement or bounded disagreement highlighted by the quantitative results. Thus we have identified the drift to be the main issue of the tracking exercise. The 2D segmentation test set and the 3D test set probably do not contain a sufficient number of frames to validate that a live sequence tracking approach is reliable on the long term. In 2D however, the test sets seem long enough so as to establish a proof of concept.

References

1. Somphone, O., Craene, M.D., Ardon, R., Mory, B., Allain, P., Gao, H., D'hooge, J., Marchesseau, S., Sermesant, M., Delingette, H., Saloux, E.: Fast myocardial motion and strain estimation in 3D cardiac ultrasound with sparse demons. In: ISBI'13 Proceedings. (2013) 1182–1185
2. Mory, B., Somphone, O., Prevost, R., Ardon, R.: Real-time 3D image segmentation by user-constrained template deformation. In: MICCAI'12 Proceedings. (2012)
3. Thirion, J.P.: Image matching as a diffusion process: an analogy with Maxwell's demons. Medical Image Analysis **2**(3) (1998) 243–260
4. Mansi, T., Pennec, X., Sermesant, M., Delingette, H., Ayache, N.: iLogDemons: a demons-based registration algorithm for tracking incompressible elastic biological tissues. International Journal of Computer Vision **92**(1) (2011) 92–111

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 1-3 list the details for each sequence. The data, which was released for training and test purposes at different times, is divided into 3 categories, namely 2D sequences with annotation of point-landmarks (see Table 1), 2D sequences with segmentations (see Table 2) and 3D sequences with point-landmarks (see Table 3).

Data Contributors

Six groups provided data and generally also the corresponding annotations for CLUST 2014. These groups and their related publications are listed below, following the order of appearance in Tables 1-3.

ETH	[3, 6]	Computer Vision Laboratory, ETH Zurich, Switzerland
MED	-	mediri GmbH, Heidelberg, Germany
OX	[2]	Institute of Biomedical Engineering, University of Oxford, UK
EMC	[1]	Biomedical Imaging Group, Departments of Radiology and Medical Informatics, Erasmus MC, Rotterdam, The Netherlands
ICR	[4, 5]	Joint Department of Physics, Institute of Cancer Research & Royal Marsden NHS Foundation Trust, London and Sutton, UK
SMT	[7]	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 test set in **green** was released shortly before the MICCAI conference.

	Sequence info					Acquisition info		
Sequence	**Im.size**	**Im.res.**	**No.**	**Im.rate**	**Annotation**	**Scanner**	**Probe**	**Freq.**
	[pix]	**[mm]**	**frames**	**[Hz]**	**No.**			**[MHz]**
ETH-01	264x313	0.71	14516	25	1	Siemens Antares	CH4-1	2.22
ETH-02	462x580	0.40	5244	16	1	Siemens Antares	CH4-1	2.00
ETH-03	462x589	0.36	5578	17	3	Siemens Antares	CH4-1	1.82
ETH-04	472x565	0.42	2620	15	1	Siemens Antares	CH4-1	2.22
ETH-05	490x570	0.40	3652	15	2	Siemens Antares	CH4-1	2.22
ETH-06	475x548	0.37	5586	17	2	Siemens Antares	CH4-1	1.82
ETH-07	473x437	0.28	4588	14	1	Siemens Antares	CH4-1	2.22
ETH-08	466x562	0.36	5574	17	2	Siemens Antares	CH4-1	1.82
ETH-09	469x523	0.40	5247	16	2	Siemens Antares	CH4-1	1.82
ETH-10	464x560	0.40	4587	15	4	Siemens Antares	CH4-1	1.82
ETH-11	462x563	0.42	4615	15	2	Siemens Antares	CH4-1	1.82
ETH-12	478x552	0.45	4284	14	1	Siemens Antares	CH4-1	2.22
MED-01	512x512	0.41	2470	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-02	512x512	0.41	2478	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-03	512x512	0.41	2456	20	4	DiPhAs Fraunhofer	VermonCLA	5.5
MED-04	512x512	0.41	2455	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-05	512x512	0.41	2458	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-06	512x512	0.41	2443	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-07	512x512	0.41	2450	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-08	512x512	0.41	2442	20	2	DiPhAs Fraunhofer	VermonCLA	5.5
MED-09	512x512	0.41	2436	20	5	DiPhAs Fraunhofer	VermonCLA	5.5
MED-10	512x512	0.41	2427	20	4	DiPhAs Fraunhofer	VermonCLA	5.5
MED-11	512x512	0.41	2424	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-12	512x512	0.41	2450	20	3	DiPhAs Fraunhofer	VermonCLA	5.5
MED-13	524x591	0.35	3304	11	3	Zonare z.one	C4-1	4.0
MED-14	524x591	0.35	3304	11	3	Zonare z.one	C4-1	4.0
MED-15	524x591	0.35	3304	11	1	Zonare z.one	C4-1	4.0
MED-16	524x591	0.35	3304	11	2	Zonare z.one	C4-1	4.0

Table 2: Summary of the challenge data for 2D sequences with segmentations of tumor areas. The test set is listed in **black** font. The training sequences, for which all available annotations were provided, are highlighted in **red**. The test set in **green** was released shortly before the MICCAI conference.

	Sequence info					Acquisition info		
Sequence	**Im.size**	**Im.res.**	**No.**	**Im.rate**	**Annotation**	**Scanner**	**Probe**	**Freq.**
	[pix]	**[mm]**	**frames**	**[Hz]**	**No.**			**[MHz]**
OX-01	416x528	0.30	71	12	1	Zonare z.one	P4-1	3.6
OX-02	336x448	0.40	82	12	1	Zonare z.one	P4-1	3.6
OX-03	416x528	0.38	82	12	1	Zonare z.one	P4-1	3.4
OX-04	336x448	0.36	51	14.5	1	Zonare z.one	C6-2	4.4
OX-05	337x448	0.46	101	11.7	1	Zonare z.one	C6-2	3.8
OX-06	337x449	0.55	76	11	1	Zonare z.one	C6-2	3.8
OX-07	338x450	0.50	63	10	2	Zonare z.one	C6-2	3.8
OX-08	337x448	0.46	105	11	1	Zonare z.one	C6-2	3.8
OX-09	338x450	0.50	98	10	2	Zonare z.one	C6-2	3.8
OX-10	337x449	0.55	92	11	1	Zonare z.one	C6-2	3.8

Table 3: 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 test set in **green** was released shortly before the MICCAI conference.

	Sequence info					Acquisition info		
Sequence	**Im.size**	**Im.res.**	**No.**	**Im.rate**	**Annotation**	**Scanner**	**Probe**	**Freq.**
	[pix]	**[mm]**	**frames**	**[Hz]**	**No.**			**[MHz]**
EMC-01	192x246x117	1.14x0.59x1.19	79	6	1	iU22	X6-1	3.2
EMC-02	192x246x117	1.14x0.59x1.19	54	6	4	iU22	X6-1	3.2
EMC-03	192x246x117	1.14x0.59x1.19	159	6	1	iU22	X6-1	3.2
EMC-04	192x246x117	1.14x0.59x1.19	140	6	1	iU22	X6-1	3.2
EMC-05	192x246x117	1.14x0.59x1.19	147	6	1	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] Cifor, A., Risser, L., Chung, D., Anderson, E., Schnabel, J.: Hybrid Feature-Based Diffeomorphic Registration for Tumor Tracking in 2-D Liver Ultrasound Images. IEEE Trans Med Imaging 32(9), 1647 (2013)

[3] 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)

[4] 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)

[5] 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)

[6] 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)

[7] 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)