International Journal of Computer Assisted Radiology and Surgery (2018) 13:1605–1615
https://doi.org/10.1007/s11548-018-1780-0

ORIGINAL ARTICLE

Ultrasound-based liver tracking utilizing a hybrid template/optical flow approach

Tom Williamson[1] · Wa Cheung[2] · Stuart K. Roberts[3] · Sunita Chauhan[1]

Received: 26 November 2017 / Accepted: 26 April 2018 / Published online: 5 June 2018

Abstract

Purpose With the ongoing shift toward reduced invasiveness in many surgical procedures, methods for tracking moving targets within the body become vital. Non-invasive treatment methods such as stereotactic radiation therapy and high intensity focused ultrasound, in particular, rely on the accurate localization of targets throughout treatment to ensure optimal treatment provision. This work aims at developing a robust, accurate and fast method for target tracking based on ultrasound images.
Methods A method for tracking of targets in real-time ultrasound image data was developed, based on the combination of template matching, dense optical flow and image intensity information. A weighting map is generated from each of these approaches which are then normalized, weighted and combined, with the weighted mean position then calculated to predict the current position. The approach was evaluated on the Challenge for Liver Ultrasound Tracking 2015 dataset, consisting of a total of 24 training and 39 test datasets with a total of 53 and 85 annotated targets throughout the liver, respectively.
Results The proposed method was implemented in MATLAB and achieved an accuracy of 0.80 ± 0.80 (95%: 1.91) mm and 0.74 ± 1.03 (95%: 1.85) mm on the training and test data, respectively. Tracking frequencies of between 8 and 36 fps (mean of 22 fps) were observed, largely dependent on the size of the region of interest. The achieved results represent an improvement in mean accuracy of approximately 0.3 mm over the reported methods in existing literature.
Conclusions This work describes an accurate and robust method for the tracking of points of interest within 2D ultrasound data, based on a combination of multi-template matching, dense optical flow and relative image intensity information.

Keywords Motion compensation · Tracking · Accuracy · Optical Flow · Template Matching · Ultrasound

Introduction

Many surgical procedures are undergoing a significant shift toward reduced invasiveness, whether toward laparoscopic and endoscopic minimally invasive techniques, image guided needle-based tissue ablation or totally non-invasive methods such as high intensity focused ultrasound (HIFU) [1,2] or stereotactic body radiation therapy (SBRT) [3,4]. However, for thoracic and abdominal procedures the motion of the organs during the performance of these non-invasive approaches introduces a number of challenges, not least that accurate provision of treatment relies on the ability of the treatment device to accurately locate and track the target within the body without direct visualization of the site. Many approaches have been proposed toward this end; external markers (either natural anatomical features or attached artificial markers) can be tracked with high accuracy utilizing commonly available stereo tracking cameras, with the motion of the internal organs inferred from the tracked external motion through modeling or nonlinear correlation [5]. The use of external tracking information has also been utilized for gating, based on the assumption that the target is at some known position when the external marker configuration matches that of the initial configuration. However, these external tracking methods have several drawbacks. In

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11548-018-1780-0) contains supplementary material, which is available to authorized users.

✉ Tom Williamson
tom.williamson@monash.edu

[1] Department of Mechanical and Aerospace Engineering, Monash University, Lab 298, New Horizon Building, Wellington Rd, Clayton, Melbourne, VIC 3800, Australia

[2] Department of Radiology, The Alfred, Commercial Road, Melbourne, Australia

[3] Department of Gastroenterology, The Alfred, Commercial Road, Melbourne, Australia

the case of attached markers, these must be placed prior to preoperative imaging and planning and may shift between treatments. Furthermore, external tracking often does not capture the complex motion of the organs within the body and even slight errors in the model or correlation function can lead to significant deviations at the target site [6].

In order to overcome these challenges, the use of a number of medical imaging techniques, including planar x-ray, MRI and ultrasound, has been investigated. While x-ray imaging can be performed in real time, the applied ionizing radiation dose and lack of soft tissue contrast make it a poor choice for long-term tracking of soft abdominal organs [7]. While MRI can provide high-resolution and excellent tissue contrast, real-time MRI imaging remains somewhat uncommon and the generated magnetic fields mean that specialized equipment must be utilized [8–10]. Conversely, improvements in image quality over the last decades, in combination with real-time image acquisition and a lack of ionizing radiation or generated magnetic fields, make ultrasound an interesting viable alternative for target tracking during non-invasive treatment. To this end, this work describes the development of an accurate and robust method for target tracking in planar ultrasound images and the evaluation of the described method based on the CLUST2015 2D dataset.

Data description and previous work

A number of groups have evaluated ultrasound-based organ tracking in the past, with a large portion of recent work performed based on 2D ultrasound data generated as part of the challenge on ultrasound liver tracking (CLUST). The most recent version (CLUST2015) of this annotated collection of ultrasound time series currently contains a total of 24 training and 39 test datasets with a total of 53 and 85 annotated positions, respectively. A previous version of the dataset (CLUST2014) contained a subset of that present in the more recent collection. The location of the tracked plane and target positions vary throughout the liver, with vessel centers and other regions of high contrast typically specified as points of interest. Image resolution, frame rate, and quality vary widely throughout the collection with a wide array of feature morphologies around the target position. The challenge therefore is to accurately track a target point throughout the entire image series, with the tracked position compared to the ground truth manually annotated position [11].

Accordingly, a range of approaches have been investigated with varying degrees of accuracy (all accuracies are reported as mean ± standard deviation (95th percentile)). Based on the 2014 CLUST data challenge, O'Shea et al. [12] described the use of normalized cross-correlation block matching, achieving an accuracy of 2.61 ± 3.78 (7.98) mm. Somphone et al. [13] described a motion estimation framework called Sparse Demons relying on a Gaussian-convolution model of the deformation field, achieving an accuracy of 2.00 ± 2.87 (5.59) mm. Lübke et al. [14] achieved an accuracy of 1.91 ± 2.47 (5.32) mm utilizing a slightly modified dense optical flow algorithm. Benz et al. [15] evaluated a scale-adaptive, rotation-invariant kernel-based algorithm, demonstrating an accuracy of 1.84 ± 2.42 (5.34) mm, while Kondo [16] achieved an accuracy of 1.83 ± 3.16 (4.82) mm based on a multiple template matching approach. König et al. evaluated an algorithm based on a variational nonlinear image registration method using Normalized Gradient Fields, extended by a moving window registration of the first-frame annotation, achieving an accuracy of 1.51 ± 1.88 (4.06) mm [17], while Röthlubbers et al. [18] utilized a Bayesian approach to target prediction achieving an accuracy of 1.52 ± 1.38 (4.08). More recently, based on additional data provided for CLUST2015, Nouri et al. [19] utilized a convolutional neural network for target tracking achieving an accuracy of 3.35 ± 5.21 (14.19) mm. Kondo evaluated a method based on a kernelized correlation filter, as well as introducing multiple extensions, achieving an accuracy of 2.91 ± 10.52 (5.18) mm [20]. Makhinya et al. demonstrated an accuracy of 1.44±2.80 (3.62) mm using an approach taking advantage of elliptic and template-based models of vessels in the liver, coupled with a robust optic-flow framework [21], while Hallack et al., combined logDemons nonlinear registration, which estimates the motion within ultrasound sequences, with a moving window tracking method that propagates the estimated motion around the region of interest to subsequent frames achieving an accuracy of 1.21 ± 3.17 (2.82) mm [22]. Most recently, Ozkan et al. extended the approach described by Makhinya et al, integrating the use of supportive features to improve accuracy and allow tracking of otherwise occluded features; the improved approach demonstrated an accuracy of 1.04 ± 1.48 (2.26) mm [23]. There is a wide array of other work in the general field of ultrasound tracking. Approaches include correlation-based speckle tracking [24], variations on active contours [25,26], machine learning approaches [27] and deformable meshes [28] among others [29,30].

Materials and methods

Our proposed tracking method utilizes three major pieces of information to create a map of the likely position of the tracked point and subsequently estimate its position.

Image velocity

First, in order to minimize algorithm processing time, the image is reduced to a symmetrical region of interest around the desired point with an initial size of $3 \cdot s_{\text{template}}$ (whereby s_{template} is the size of the template, selected as described in section "Normalization and combination of weighting maps" below); this region remains static but is subsequently allowed to grow as the predicted position approaches its borders. In cases in which the initial position was extremely close to the edge of the U/S image, an initial template offset was applied in both the X and Y directions such that the template did not contain any pixels external to the image; this offset was then removed after computation of the cross-correlation. This offsetting process was performed automatically by detecting the borders of the fan-shaped ultrasound image (filter image, calculate first and last indices of intensity greater than zero for each row, calculate convex hull with selected indices, set points external to hull to maximum, perform Canny edge detection and Hough transform to get edge lines), calculating the distance between the initial point and nearest edge and then applying the shift based on this distance. The velocity of the tracked liver location is then estimated utilizing optical flow, specifically the approach described by Farnebäck [31], in which neighborhoods within each frame are approximated using polynomials with the estimation of motion based on the behavior of an ideal translation of a polynomial, as implemented in the MATLAB Computer Vision System Toolbox (Version 2017a, The Mathworks, USA) with the following parameters:

- *NeighborhoodSize = 5,*
- *PyramidScale = 0.5,*
- *NumPyramidLevels = 3,*
- *FilterSize = 5.*

The average velocity of the pixels within a moving region centered at the current position and with the size of template (s_{template}) is returned as the object velocity. This is modified slightly for cases in which the object is close to the edge of the image, with the averaged region shifted such that it does not overlap with the edge of the image. The position and velocity are then utilized to generate a weighting map consisting of a Gaussian centered at the predicted position. Note that the terminology "weighting map" refers to a map of the relative influence that each pixel position has on the final position estimation; these are similar in concept to a probability map, but do not necessarily sum to one.

$$M_{\text{vel}} = N\left(x_{k-1} + v, \sigma_{\text{vel}}\right) \tag{1}$$

Whereby M_{vel} is the normally distributed weighting map based on velocity and the previous position, x_{k-1} is the previous position, v the current averaged velocity and σ_{vel} the standard deviation of the distribution. The velocity is set to zero in cases in which the region of interest has grown.

An additional 'sanity' check is then performed in cases where a large velocity (magnitude greater than 5) is detected, whereby if the position of the maximum correlation given by the template matching (described below) is far from the new position predicted by the velocity (i.e. the difference between the position predicted by the velocity and the position of the peak in the template match is greater than 20 pixels), then it is assumed to be erroneous and the velocity is set to zero.

Template matching

Secondly, throughout the estimation process, a template of the region around the initial point and a vector of alternative templates is maintained. These templates represent the morphology of the region around the current (or some past) position. Prior to updating, the contrast of the (inverted) template image is reduced such that values below $\mu_{\text{intensity}} - \sigma_{\text{intensity}}$ (where $\mu_{\text{intensity}}$ are the mean intensity of the template and $\sigma_{\text{intensity}}$ the standard deviation of the template intensity values) are saturated, removing outliers and noise in the templates. The contents of the alternatives vector are updated periodically; the oldest alternative template is replaced with a template equivalent to the current morphology of the region around the predicted position at a nominal period of τ_{alt}; the template is not updated if it overlaps with the edges of the image or if the peak cross-correlation is greater than 10 pixels from the current predicted position. The 2D normalized cross-correlation is then computed for both the original template and each of the alternatives and the results weighted and summed. This process creates a weighting map allowing the determination of the location within the image that best matches both the initial template and the templates containing any subsequent morphological changes that may have occurred.

$$M_{\text{corr}} = \sum_{i=1}^{n_{\text{alt}}} \left(w_{\text{alt}} \cdot X_i\right) + X_{\text{init}} \tag{2}$$

In which M_{corr} is the weighting map based on normalized cross-correlation, n_{alt} the number of alternative templates, w_{alt} the weighting of the alternatives, X_i the cross-correlation of the alternative template and current frame for alternative i and X_{init} the cross-correlation of the initial template and current frame.

Raw image intensity

Finally, it is assumed that the position to be tracked represents some feature within the image and is subsequently located at the center of a region of particularly high or low pixel intensity (i.e. a particularly light or dark region in the image). If at the center of a region of low intensity (i.e. a dark region, assumed if the intensity at the desired point is lower than the mean intensity throughout the region encompassed by the initial template) the image is inverted before being filtered with a 3 × 3 averaging filter and normalized to between 0 and 1.

This approach biases the predicted position toward the center of nearby regions light (or dark) intensity and allows tracking to continue even in cases in which the shape of the tracked region changes significantly or when the point of interest slides along a vessel with no significant change in velocity (for example case MED-03-1_2) and in which a significantly different point on the same object is subsequently tracked.

Normalization and combination of weighting maps

Each of the generated weighting maps are first normalized such that the minimum value in each becomes zero, and the maximum becomes one, as below.

$$\hat{M}_{\text{any}} = \frac{M_{\text{any}} - \min\left(M_{\text{any}}\right)}{\max\ (M_{\text{any}}) - \min\left(M_{\text{any}}\right)} \tag{3}$$

In which M_{any} refers to any one of the weighting maps, min() refers to the minimum value within that map, max() to the maximum value within, and $\hat{M}_{\text{any}}$ to the normalized weighting map.

First, the generated position/velocity and template matching weighting maps are normalized to between 0 and 1, weighted and multiplied together to provide a combined weighting map.

$$M_{c\_\text{corr}} = \hat{M}_{\text{vel}} \cdot \left(w_{\text{corr}} \cdot \hat{M}_{\text{corr}}\right) \tag{4}$$

In which M_{c_corr} is the combined weighting map, $\hat{M}_{\text{vel}}$ the velocity map normalized to between 0 and 1, w_{corr} the weighting factor for the correlation map and $\hat{M}_{\text{corr}}$ the correlation map normalized to between 0 and 1.

Then, the generated image intensity map is normalized and combined with the position velocity weighting map.

$$M_{c\_\text{im}} = \hat{M}_{\text{vel}} \cdot \left(w_{\text{im}} \cdot \hat{M}_{\text{im}}\right) \tag{5}$$

In which M_{c_im} is the second combined weighting map, w_{im} is the image weighting factor and $\hat{M}_{\text{im}}$ is the normalized image map (essentially the *normalized region of interest*)

The two combined maps are then summed:

$$M_{\text{comb}} = M_{c\_\text{corr}} + M_{c\_\text{im}} \tag{6}$$

Note that there is one special case: if the point is close to the edge of the image such that there is an overlap between the edge and template, only the image-based weighting map is used (i.e. not the cross-correlation template-based approach), as below:

$$M_{\text{comb}} = M_{c\_\text{im}} \tag{7}$$

The resulting values are then raised to a further final weighting power as below.

$$M_{\text{final}} = M_{\text{comb}}{}^{w_{\text{final}}} \tag{8}$$

where M_{final} is the final weighting map, M_{comb} the combined map and w_{final} the weighting factor. This weighting factor tends to minimize the effects of regions of lower weight on the final estimation. The position of the object within the frame is then calculated as the weighted mean position, with the weight values taken from this final map.

$$x_k = \frac{\sum_{i=1}^{n}\left(M_{\text{final}}^{i} \cdot x_i\right)}{\sum_{i=1}^{n} M_{\text{final}}^{i}} \tag{9}$$

In which x_k is the predicted position of the object in frame k, n is the number of pixels in the final weighting map, M_{final}^{i} is the value of the final weighting map for pixel number i and x_i is the position of pixel number i.

The resulting point is then utilized to initialize the velocity-based weighting map of the following frame.

Error detection and algorithm reset

The algorithm is able to reset itself based on the current position (averaged over the last 200 frames) or the starting position and the position of the maximum normalized cross-correlation. The following criteria must be met for a reset to occur:

- The distance between the new position of maximum correlation and the initial or current position is below the threshold t_{dist}
- The current predicted position is relatively far from the initial/mean position (i.e. greater than $\frac{t_{\text{dist}}}{2}$)
- The maximum normalized cross-correlation is significantly better than the correlation at the predicted position (i.e. the difference is greater than threshold t_{diff}).

If each of these is true, the predicted is set to the location of the highest normalized cross-correlation. This allows the algorithm to reset itself if the tracked position drifts, and an appropriate re-initialization point can be located.

Template size selection

Template sizes were selected for each case prior to analysis; however an initial evaluation revealed that significant variations in template size occurred (up to 15 pixels) when selecting templates manually. Thus the selection process, whereby two points were selected that encompassed all the relevant information describing the region around the tracked point and the template size set to the larger of the X and Y

ranges, was completed for each of the training cases a total of 5 times. The average was then taken as the ground truth template size.

Template sizes were then selected automatically: a region of interest (ROI) of 71 × 71 pixels was isolated around the point of interest, and the intensity values along the horizontal and vertical lines passing through the center of this region extracted. The pixel value at the center of the ROI was then compared to each of the values along these lines, and the closest points on either side of the center whereby a difference of one intensity standard deviation was observed were calculated. The difference between these two points represents the horizontal and vertical range of the object; in cases in which the intensity along the center line did not fall outside of the desired range, the appropriate point was set to the minimum or maximum according to which side it was expected. The minimum of the horizontal and vertical ranges was then taken as the base template size.

These automatic template sizes were then compared to the averaged manual (ground truth) template sizes and an offset added such that the difference between the automatic and manual template sizes was minimized (an offset of 17 pixels was determined to be optimal) (Fig. 1). A further offset of five pixels was then added to ensure that all relevant information was captured, with the template size capped at 60 pixels. This automatic process was then repeated for the test data. The template selection process and a comparison of automatic and averaged manual approaches can be seen in Fig. 2. All algorithm implementation and evaluation was performed in MATLAB 2017a, based on the CLUST2015 training and test datasets.

Results

The described hybrid tracking method achieved an accuracy of 0.80 ± 0.80 (1.91) mm and 0.74 ± 1.03 (1.85) mm on the training and test data, respectively. The results for each of the individual cases are attached in appendix; results for each dataset are summarized in Table 1. The achieved results compare favorably with the existing literature, achieving an improvement in mean accuracy of 0.3 mm (approximately 30%) over the previous best published results [23]; the results achieved are compared to the existing published state of the art in Table 2. All parameters, as summarized in Table 3, were set empirically by systematically altering each and minimizing the error observed on the training data. The selected parameter values were subsequently fixed and applied in all cases.

The relationship between tracking speed (i.e. frame rate) and the size of the ROI is shown in Fig. 3; both the calculation time for a single frame for vs the region of interest area for all frames, and the mean frame rate vs the mean ROI size for each of the 53 training datasets is shown. A minimum average frame rate of 8 fps, and maximum average frame rate of 36 fps were observed. An overall mean frame rate of 22 fps was observed on the training data (Intel Xeon E5-1650 @ 3.5GHz, 32 GB RAM). A coefficient of determination (r^2) of 0.8 was observed between mean ROI area and mean frame rate; the remaining variation in frame rate (as well as the significant variations in individual calculation times) is likely due to increased optical flow calculation time during large motion, changes in region of interest size and other additional calculations, such as in the case of position reset.

The largest error (4.55 ± 5.31 (18.26) mm) was observed in case ICR-09_1. The reason for this was likely the significant change in morphology of the target site, including the inversion of the object from dark to light which the algorithm is currently not capable of adequately accounting for.

Algorithm failure was observed in frames approximately 2468–2959, at which point the algorithm was able to reset itself. The initial target morphology, morphology at the point of failure, 250 frames after failure and immediately after reset are shown in Fig. 4. Note that similar errors occurred in the case of ICR-08_1, where significant changes in target intensity relative to the surrounding region were also observed. In this case the algorithm was able to more quickly reset itself (Fig. 5).

Mean errors above 2 mm were observed in four cases in addition to ICR-09_1: MED-07-3_1: 3.16 ± 1.79 (6.57) mm, MED-08-2-3: 2.08 ± 1.19 (4.09) mm, MED-11_1: 2.18 ± 1.09 (4.13) mm and CIL-03_2: 2.51 ± 0.77 (3.82) mm. The reasons for these errors are likely due to one of a number of factors. In one case (MED-07-3_1), a combination of very large object size (relatively few features within template for tracking) and sudden motion of the tracked object (loss of object position and eventual reset) was observed. In two cases (MED-08-2-3, MED-11_1) significant change in object morphology was observed, resulting in loss of object position; in these cases, algorithm reset was delayed until the object returned to approximately the initial morphology. Finally, one case demonstrated partial obstruction of the object view (CIL-03_2) at various times during tracking, with large errors observed while the target was obstructed. No significant relationship was found between mean tracking accuracy and object target size (coefficient of determination $r^2 = 0.09$), resolution ($r^2 = 0.01$) or length of tracking ($r^2 = 0.07$).

Discussion and conclusions

The achieved results represent an improvement in mean accuracy of approximately 0.3 mm when compared to the current published state of the art [23]. The developed method is comparatively robust, accurate and reasonably fast, without

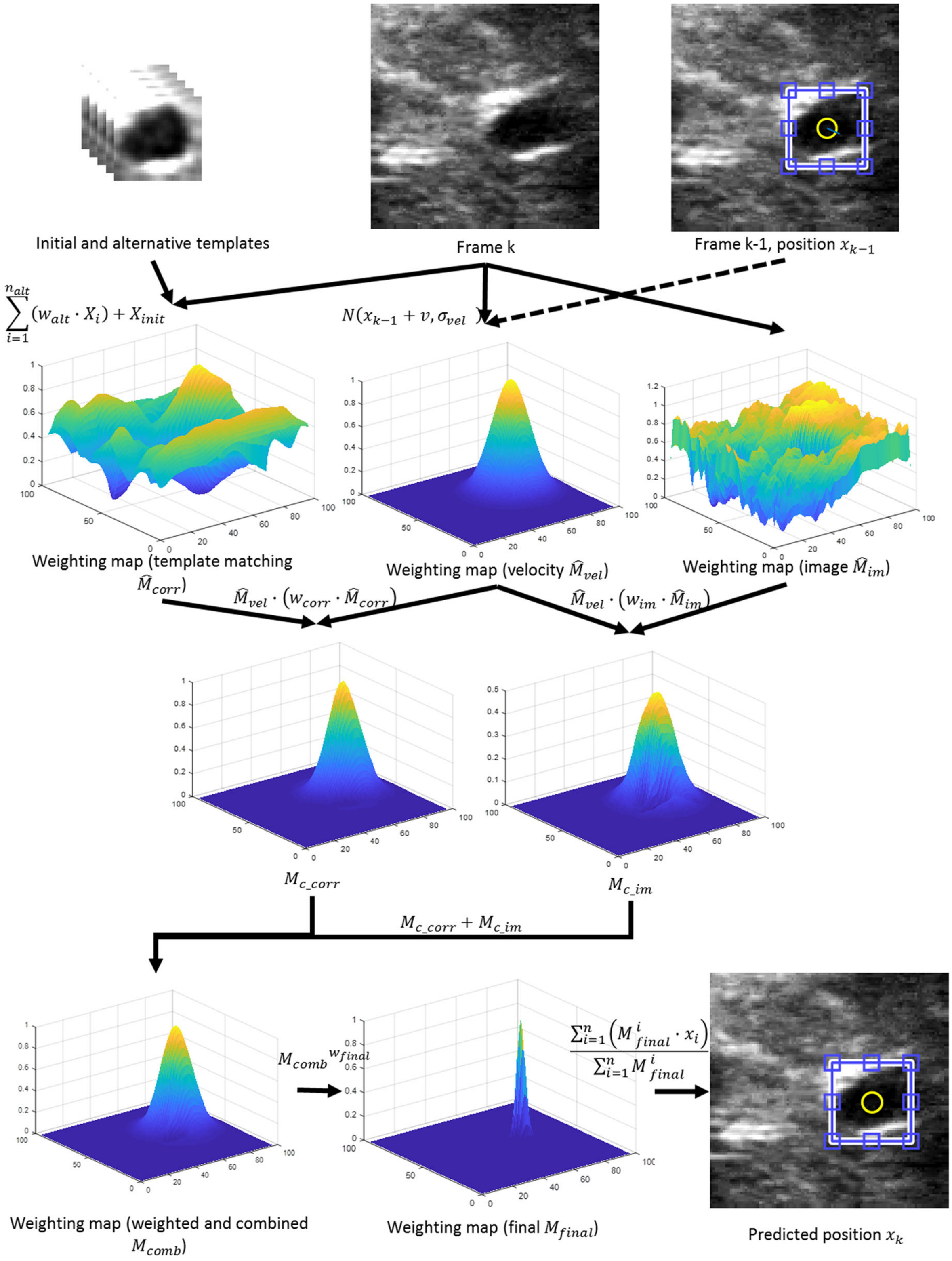
Initial and alternative templates
Frame k
Frame k-1, position x_{k-1}
$\sum_{i=1}^{n_{alt}}(w_{alt} \cdot X_i) + X_{init}$
$N(x_{k-1} + v, \sigma_{vel})$
Weighting map (template matching $\hat{M}_{corr}$)
Weighting map (velocity $\hat{M}_{vel}$)
Weighting map (image $\hat{M}_{im}$)
$\hat{M}_{vel} \cdot (w_{corr} \cdot \hat{M}_{corr})$
$\hat{M}_{vel} \cdot (w_{im} \cdot \hat{M}_{im})$
M_{c_corr}
M_{c_im}
$M_{c\_corr} + M_{c\_im}$
$M_{comb}{}^{w_{final}}$
$\frac{\sum_{i=1}^{n}(M^{i}_{final} \cdot x_i)}{\sum_{i=1}^{n} M^{i}_{final}}$
Weighting map (weighted and combined M_{comb})
Weighting map (final M_{final})
Predicted position x_k

Fig. 1 The prediction of the object position is based on the combination of normalized cross-correlation between the current frame and an array of templates (top row left, middle), an estimate of the image velocity based on the previous frame and object location (top row right) and the current frame intensity (top row middle), used to create weighting maps (second row). These maps are then normalized, weighted and combined (third row, fourth row left) before a final weighting factor is applied (bottom row middle) and the weighted mean position calculated (bottom row right)

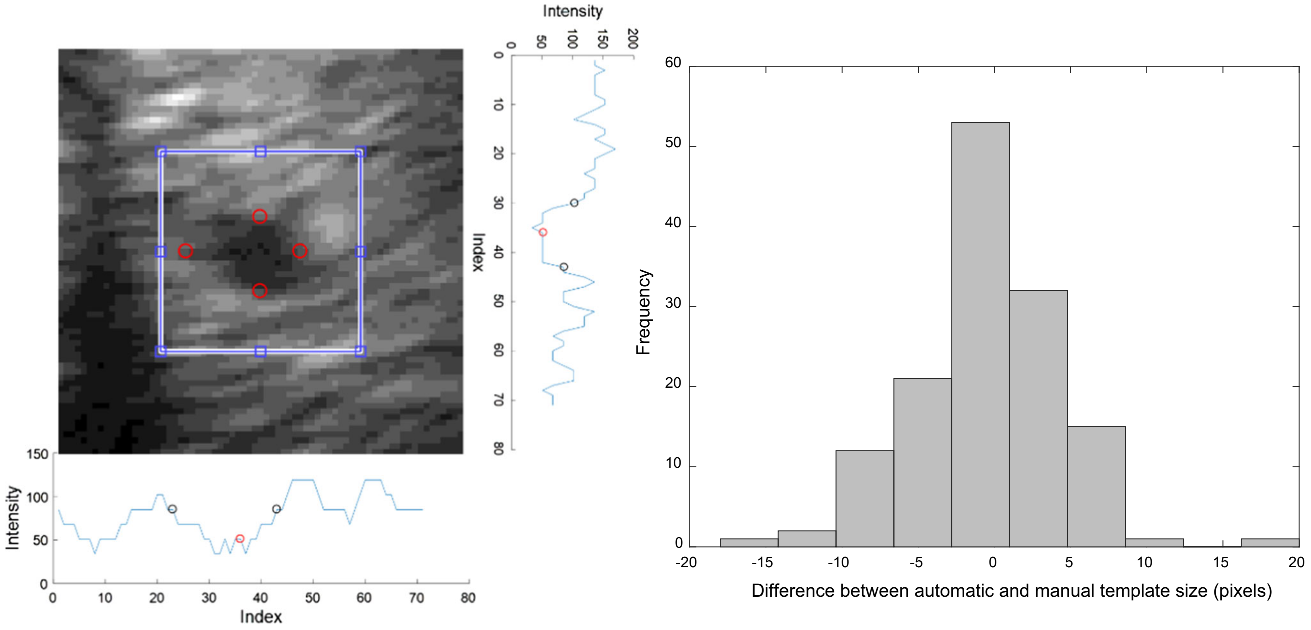

Fig. 2 Selection of object edges, with final offset bounding box (left) and comparison of automatic and manual template size selection for all cases, with final 5 pixel offset removed (right)

Table 1 Results achieved for individual datasets

Dataset	Mean	SD	95%ile	Min	Max
CIL	1.01	0.84	2.81	0.04	5.73
ETH	0.50	0.45	1.13	0.00	16.30
ICR	1.06	2.02	2.45	0.01	19.08
MED1	1.04	1.14	2.75	0.01	16.04
MED2	0.99	0.78	2.38	0.01	9.97
Total	0.74	1.03	1.85	0.00	19.08

Table 2 Comparison of published results with results achieved with described algorithm

Author	Data	Mean	Standard Deviation	95%ile
O'Shea [12]	CLUST2014	2.61	3.78	7.98
Somphone [13]	CLUST2014	2.00	2.87	5.59
Lübke [14]	CLUST2014	1.91	2.47	5.32
Benz [15]	CLUST2014	1.84	2.42	5.34
Kondo [16]	CLUST2014	1.83	3.16	4.82
Rothlubbers [18]	CLUST2014	1.52	1.38	4.08
König [17]	CLUST2014	1.51	1.88	4.06
Nouri [19]	CLUST2015	3.35	5.21	14.19
Kondo [20]	CLUST2015	2.91	10.52	5.18
Makhinya [21]	CLUST2015	1.44	2.80	3.62
Hallack [22]	CLUST2015	1.21	3.17	2.82
Ozkan [23]	CLUST2015	1.04	1.48	2.26
This work	CLUST2015	0.74	1.03	1.85

Table 3 Parameter values for evaluated implementation

Parameter	Value	Description
s_{template}	25–60	Size of template in pixels
σ_{vel}	$\frac{s_{\text{template}}}{2}$	Standard deviation of velocity Gaussian in pixels
n_{alt}	5	Number alternative templates
w_{alt}	0.1	Weight value of alternative templates
τ_{alt}	20	Update period of alternative templates in frames
w_{im}	0.5	Combination weighting factor for image weighting map
w_{corr}	1	Combination weighting factor for correlation weighting map
w_{final}	15	Final weighting factor for weighted mean position calculation
t_{dist}	$\frac{s_{\text{template}}}{2}$	Distance threshold from initial and mean position to maximum correlation and current position for reset in pixels
t_{diff}	$\frac{1+n_{\text{alt}} \cdot w_{\text{alt}}}{n_{\text{alt}}+1}$	Threshold between maximum and current cross-correlation for reset

requiring patient specific algorithm training. The speed of the algorithm depends on the specific size of the region of interest with larger regions requiring longer to calculate, as shown in Fig. 3; however the current implementation can be performed at between 8 and 36 frames per second (latency of approximately 30–125 ms) for a single target including the loading of image data from disk.

There remains considerable space for optimization through parallelization or GPU implementation, and thus real-time tracking with a latency below 20 ms is expected to be feasible [32,33].

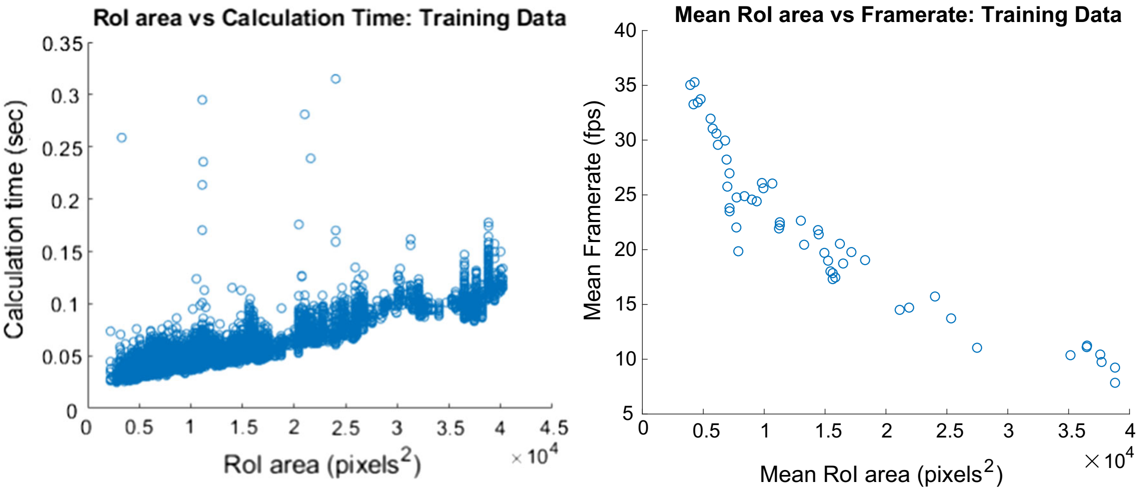

Fig. 3 Initial template size vs average frame rate for all training cases

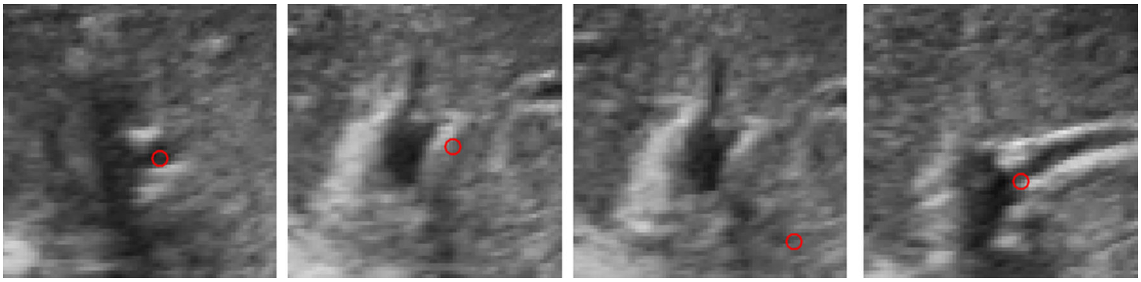

Fig. 4 Target morphology and predicted position for ICR-09_1 at Frame 1 (left), failure at Frame 2468 (left middle), at 250 frames after failure (right middle) and immediately after reset at frame 2959 (right)

The developed approach utilizes a combination of templates, with normalized cross-correlation utilized to determine the similarity between the template and region of interest, optical flow, and intensity information. The use of templates allows the algorithm to provide a simple measure of algorithm confidence: the better the match, the higher the confidence. This may be particularly useful in ensuring that treatment is only applied when the desired target is in the correct location with a high degree of confidence.

The CLUST2015 dataset presents a number of challenges, including dropped frames (resulting in sudden large shifts in target position), sudden large physiological motion, and large changes in target morphology. Furthermore, the locations of the tracked points (vessel centers) are spread throughout the liver; as the liver is such a large organ, it interacts with several others during inhale exhale cycles, including the heart, resulting in movement patterns that may vary considerably. The developed algorithm is generally able to compensate for each of these factors; the implemented reset function detects sudden changes in position, while the integration of gross relative intensity information allows for changes in region shape and intensity. The algorithm will fail, however, in cases in which the intensity of the object relative to the surrounding region switches from high to low or vice versa. This is illustrated by the large errors observed in case ICR-09_1.

While the approach allows tracking in 2D, and the majority of liver motion occurs in the cranial-caudal direction, the structures of the liver move in all three dimensions. Methods for extracting information about the extent of motion in the out of plane direction are currently under investigation. Proposed approaches include the utilization of a secondary perpendicular image plane with the developed algorithm running in both planes, or the use of speckle information to estimate the out of plane position. One other approach could be the use of morphology changes in the target site, in which a larger volume is acquired prior to the start of the tracking process and additional morphology matching performed to determine the depth; this may also have the advantage of reducing errors due to changes in target shape and intensity caused by out of plane motion.

Furthermore, the described method cannot track positions that become occluded or otherwise invisible in the acquired images. The approach described by Ozkan and Tanner [23], in which supporting features are utilized to estimate the posi-

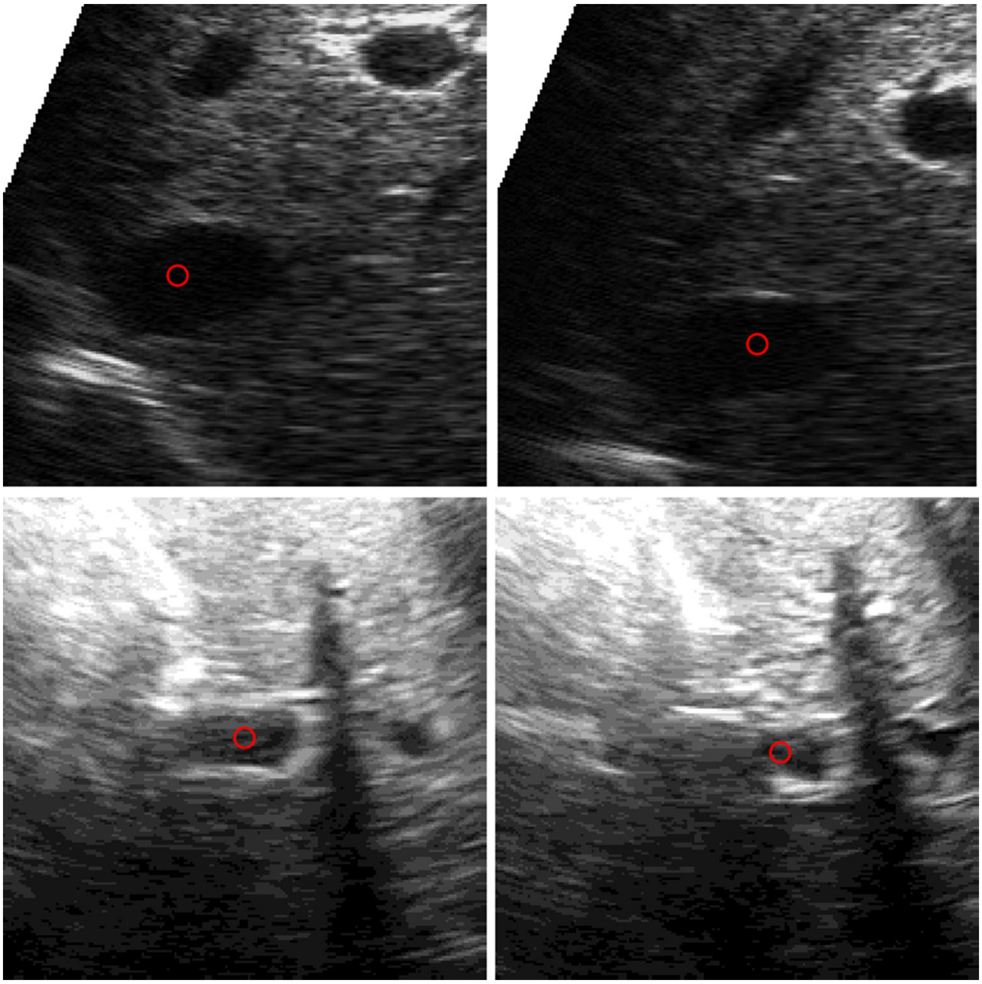

Fig. 5 Additional examples of high error cases. MED-07-3_1 initial frame (top left) and during rapid organ motion (top right). CIL-03_2 initial frame (bottom left) and partially occluded by solid object (rib) along tracking path (bottom right)

tion of occluded objects, could be utilized to overcome this. The utilization of additional features throughout tracking may also lead to improvements in the overall accuracy of the method, and thus the implementation of this approach will be the subject of future work. The implemented algorithm includes a large number of parameters, all of which were tuned empirically based on the provided training data. An initial implementation of a brute force optimization with only three variable parameters (w_{im}, w_{corr}, both varied between 0.25 and 1 in steps of 0.25, and w_{final}, varied between 5 and 30 in steps of 5) resulted in no significant improvement in training accuracy over the empirical tuning. Furthermore, a total of 96 evaluations for the 53 training cases (5088 individual cases) were required, with this analysis needing approximately 10 days to complete. Further optimization is likely possible and a more efficient method for determining the optimal values for all parameters is currently underway. The use of high performance computing hardware (possibly through the use GPU hardware) is being investigated to allow a comprehensive evaluation and optimization, while it may also be possible to optimize parameters for individual cases, for example, based on specific target size and morphology. Finally, while the algorithm has been optimized for use with ultrasound images, the approach could be easily applied for tracking in any imaging modality, assuming the motion of the target is quasi-periodic, with minimal modifications.

This work has described an accurate and robust method for tracking points of interest within 2D ultrasound data based on a combination of multi-template matching, dense optical flow and relative image intensity information. The developed approach was evaluated on the challenge for liver ultrasound tracking (CLUST2015) dataset, consisting of a total of 24 training and 39 test datasets with a total of 53 and 85 annotated targets, respectively. The current MATLAB implementation can be performed in near real-time and represents an improvement in mean tracking accuracy of approximately 0.3 mm, future work will involve real-time implementation of the algorithm for robotic HIFU as well as investigation of 4D tracking approaches.

Acknowledgements This work is funded by the Australian National Health and Medical Research Council.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights No human or animal experiments were performed in this study.

Appendix 1: All tracking results

Landmark	Mean	SD	TE95th	Min	Max	Resolution
CIL-03_1	0.69	0.46	1.52	0.12	2.74	0.40
CIL-03_2	2.51	0.77	3.82	0.43	5.73	0.40
CIL-04_1	0.75	0.36	1.35	0.08	1.61	0.50
CIL-04_2	0.72	0.38	1.42	0.05	1.77	0.50
CIL-05_1	0.87	0.54	1.82	0.06	2.74	0.30
CIL-05_2	0.54	0.27	1.00	0.04	1.64	0.30
ETH-06-1_1	0.54	0.23	0.94	0.02	1.31	0.40
ETH-06-2_1	0.45	0.23	0.86	0.01	1.40	0.38
ETH-07-1_1	0.40	0.15	0.64	0.05	1.16	0.37
ETH-07-1_2	0.72	0.39	1.35	0.04	2.34	0.37
ETH-07-2_1	0.45	0.56	1.67	0.02	7.63	0.37
ETH-07-2_2	0.60	0.34	1.19	0.02	2.47	0.37
ETH-08-1_1	0.27	0.16	0.56	0.02	1.07	0.36
ETH-08-1_2	0.45	0.30	1.08	0.01	1.84	0.36
ETH-08-2_1	0.33	0.20	0.70	0.03	1.03	0.36
ETH-08-2_2	0.66	0.39	1.41	0.03	2.13	0.36
ETH-09-1_1	0.54	0.68	1.18	0.02	13.11	0.40
ETH-09-1_2	0.51	0.60	0.89	0.01	12.42	0.40
ETH-09-1_3	0.45	0.63	0.94	0.02	12.72	0.40
ETH-09-1_4	0.73	1.19	1.87	0.05	16.30	0.40
ETH-09-2_1	0.59	0.23	0.99	0.02	1.35	0.42
ETH-09-2_2	0.55	0.32	1.13	0.01	2.13	0.42
ETH-09-2_3	0.55	0.31	1.15	0.05	1.69	0.42
ETH-10-1_1	0.27	0.16	0.58	0.01	1.09	0.36
ETH-10-1_2	0.30	0.16	0.60	0.03	0.93	0.36
ETH-10-1_3	0.38	0.20	0.74	0.01	0.90	0.36
ETH-10-2_1	0.34	0.20	0.72	0.01	1.08	0.36
ETH-10-2_2	0.61	0.29	1.12	0.01	1.92	0.36
ETH-10-2_3	0.43	0.24	0.86	0.01	1.55	0.36
ETH-11-1_1	0.39	0.24	0.79	0.02	1.87	0.45
ETH-11-1_2	0.61	0.35	1.29	0.03	1.94	0.45
ETH-11-2_1	0.79	0.49	1.76	0.02	3.52	0.45
ETH-12-1_1	0.57	0.28	1.09	0.02	1.93	0.71
ETH-12-2_1	0.75	0.40	1.51	0.00	2.62	0.77
ETH-13-1_1	0.38	0.83	0.67	0.00	14.53	0.71
ETH-13-2_1	0.28	0.16	0.59	0.01	0.93	0.71
ICR-05_1	0.57	0.22	0.98	0.04	1.40	0.55
ICR-05_2	0.77	0.48	1.54	0.03	3.37	0.55
ICR-06_1	0.65	0.34	1.26	0.03	2.13	0.55
ICR-06_2	1.24	0.98	2.03	0.10	10.48	0.55
ICR-07_1	0.45	0.25	0.91	0.02	1.34	0.49
ICR-07_2	0.77	0.46	1.60	0.02	2.45	0.49
ICR-08_1	1.22	2.83	6.84	0.01	16.81	0.50
ICR-08_2	0.41	0.24	0.84	0.02	1.31	0.50
ICR-08_3	0.40	0.21	0.78	0.02	1.15	0.50
ICR-09_1	4.55	5.31	18.26	0.13	19.08	0.57
ICR-09_2	1.57	0.45	2.18	0.01	2.53	0.57
ICR-10_1	0.77	0.28	1.23	0.04	1.55	0.49
ICR-10_2	0.45	0.26	0.97	0.02	1.38	0.49
MED-06-1_1	1.09	0.53	2.09	0.22	3.20	0.41
MED-06-1_2	0.71	0.30	1.30	0.09	1.65	0.41
MED-06-1_3	0.85	0.41	1.52	0.12	2.83	0.41
MED-06-1_4	0.36	0.43	0.67	0.01	6.50	0.41
MED-06-2_1	1.43	1.10	3.00	0.12	9.93	0.41
MED-06-2_2	0.43	0.26	0.92	0.02	1.41	0.41
MED-06-2_3	1.04	0.58	2.08	0.02	2.67	0.41
MED-07-1_1	0.85	0.50	1.86	0.03	3.03	0.41
MED-07-1_2	0.88	0.31	1.41	0.07	1.70	0.41
MED-07-1_3	1.02	0.43	1.66	0.11	1.89	0.41
MED-07-2_1	0.56	0.32	1.17	0.04	1.81	0.41
MED-07-2_2	0.54	0.27	1.02	0.03	1.33	0.41
MED-07-2_3	0.43	0.24	0.90	0.06	1.42	0.41
MED-07-3_1	3.16	1.79	6.57	0.11	7.79	0.41
MED-07-3_2	1.68	3.49	14.45	0.02	16.04	0.41
MED-07-3_3	0.76	0.37	1.34	0.02	1.84	0.41
MED-07-4_1	1.62	0.88	3.03	0.08	3.94	0.41
MED-07-4_2	0.65	0.31	1.17	0.07	1.63	0.41
MED-07-4_3	0.36	0.20	0.75	0.02	1.00	0.41
MED-07-4_4	0.29	0.16	0.58	0.02	0.84	0.41
MED-08-1_1	0.78	0.37	1.45	0.07	1.90	0.35
MED-08-1_2	1.05	0.37	1.65	0.18	1.98	0.35
MED-08-1_3	0.46	0.23	0.86	0.02	1.12	0.35
MED-08-2_1	0.92	0.32	1.45	0.16	1.80	0.35
MED-08-2_2	0.97	0.30	1.51	0.16	1.83	0.35
MED-08-2_3	2.08	1.19	4.09	0.23	5.35	0.35
MED-09_1	1.38	1.30	3.61	0.09	9.63	0.48
MED-10_1	1.19	0.82	3.13	0.10	5.58	0.45
MED-10_2	0.85	0.52	1.92	0.05	2.80	0.45
MED-11_1	2.18	1.09	4.13	0.13	4.97	0.45
MED-11_2	1.40	0.74	2.60	0.11	3.39	0.45
MED-12_1	0.94	0.42	1.65	0.08	2.35	0.48
MED-12_2	1.50	0.75	2.75	0.07	4.88	0.48
MED-13_1	0.45	0.26	0.94	0.02	1.25	0.27
MED-14_1	1.24	0.59	2.22	0.16	4.87	0.27
MED-14_2	0.89	1.08	1.59	0.01	9.97	0.27

References

1. Illing RO, Kennedy JE, Wu F, ter Haar GR, Protheroe AS, Friend PJ, Gleeson FV, Cranston DW, Phillips RR, Middleton MR (2005) The safety and feasibility of extracorporeal high-intensity focused ultrasound (HIFU) for the treatment of liver and kidney tumours in a Western population. Br J Cancer 93(8):890–5
2. Chauhan S (2008) Image guided robotic systems for focal ultrasound based surgical applications. In: Medical robotics. I-Tech Education and Publishing, pp 29–40
3. Timmerman R, Kavanagh B (2005) Stereotactic body radiation therapy. Curr Probl Cancer 29(3):120–157

4. Timmerman R (2010) Stereotactic body radiation therapy for inoperable early stage lung cancer. JAMA 303(11):1070
5. Abhilash RH, Chauhan S (2012) Respiration-induced movement correlation for synchronous noninvasive renal cancer surgery. IEEE Trans Ultrason Ferroelectr Freq Control 59(7):1478–86
6. Gierga DP, Brewer J, Sharp GC, Betke M, Willett CG, Chen GTY (2005) The correlation between internal and external markers for abdominal tumors: implications for respiratory gating. Int J Radiat Oncol 61(5):1551–1558
7. Pepin EW, Wu H, Zhang Y, Lord B (2011) Correlation and prediction uncertainties in the CyberKnife Synchrony respiratory tracking system. Med Phys 38(7):4036–4044
8. Crijns SPM, Raaymakers BW, Lagendijk JJW (2012) Proof of concept of MRI-guided tracked radiation delivery: tracking one-dimensional motion. Phys Med Biol 57(23):7863–7872
9. Menten MJ, Fast MF, Nill S, Kamerling CP, McDonald F, Oelfke U (2016) Lung stereotactic body radiotherapy with an MR-linac—Quantifying the impact of the magnetic field and real-time tumor tracking. Radiother Oncol 119(3):461–466
10. Kim Y (2015) Advances in MR image-guided high-intensity focused ultrasound therapy. Int J Hyperth 31(3):225–232
11. De Luca V, Benz T, Kondo S, König L, Lübke D, Rothlübbers S, Somphone O, Allaire S, Bell ML, Chung DYF, Cifor A (2015) The 2014 liver ultrasound tracking benchmark. Phys Med Biol 60(14):5571
12. O'Shea T, Bamber J, Harris E (2014) Liver feature motion estimation in long high frame rate 2D ultrasound sequences. In: Challenge on liver ultrasound tracking, pp 5–12
13. Somphone O, Allaire S, Mory B, Dufour C (2014) Live feature tracking in ultrasound liver sequences with sparse demons. In: MICCAI workshop: challenge on liver ultrasound tracking, pp 53–60
14. Luebke D, Grozea C (2014) MICCAI CLUST 2014—Bayesian real-time liver feature ultrasound tracking. In: MICCAI workshop: challenge on Liver Ultrasound Tracking, pp 37–44
15. Benz T, Kowarschik M, Navab N (2014) Kernel-based tracking in ultrasound sequences of liver. In: Proceedings of challenge on liver ultrasound tracking, MICCAI workshop, pp 21–28
16. Kondo S (2014) Liver ultrasound tracking using long-term and short-term template matching. In: MICCAI workshop: challenge on liver ultrasound tracking, pp 13–20
17. König L, Kipshagen T, Rühaak J (2014) A non-linear image registration scheme for real-time liver ultrasound tracking using normalized gradient fields. In: MICCAI workshop: challenge on liver ultrasound tracking, pp 29–36
18. Rothlübbers S, Schwaab J, Jenne J, Günther M (2014) MICCAI CLUST 2014—Bayesian real-time liver feature ultrasound tracking. In: MICCAI workshop: challenge on Liver Ultrasound Tracking, pp 45–52
19. Nouri D, Rothberg A (2015) Liver ultrasound tracking using a learned distance metric. In: MICCAI workshop: challenge on liver ultrasound tracking, pp 5–12
20. Kondo S (2015) Liver ultrasound tracking using kernelized correlation filter with adaptive window size selection. In: MICCAI workshop: challenge on liver ultrasound tracking, pp 13–19
21. Makhinya M, Goksel O (2015) Motion tracking in 2D ultrasound using vessel models and robust optic-flow. In: MICCAI workshop: challenge on liver ultrasound tracking, pp 20–27
22. Hallack A, Papiez W B, Cifor A, Gooding MJ, Schnabel J A (2015) Robust liver ultrasound tracking using dense distinctive image features. In: Proceedings MICCAI workshop: challenge on liver ultrasound tracking, pp 28–35
23. Ozkan E, Tanner C, Kastelic M, Mattausch O, Makhinya M, Goksel O (2017) Robust motion tracking in liver from 2D ultrasound images using supporters. Int J Comput Assist Radiol Surg 12(6):941–950
24. Golemati S, Sassano A, Lever MJ, Bharath AA, Dhanjil S, Nicolaides AN (2003) Carotid artery wall motion estimated from b-mode ultrasound using region tracking and block matching. Ultrasound Med Biol 29(3):387–399
25. Li M, Kambhamettu C, Stone M (2005) Automatic contour tracking in ultrasound images. Clin Linguist Phon 19(6–7):545–554
26. Zhang X, Günther M, Bongers A (2010) Real-time organ tracking in ultrasound imaging using active contours and conditional density propagation, pp 286–294
27. De Luca V, Tschannen M, Székely G, Tanner C (2013) A learning-based approach for fast and robust vessel tracking in long ultrasound sequences, pp 518–525
28. Yeung Fai, Levinson SF, Fu Dongshan, Parker KJ (1998) Feature-adaptive motion tracking of ultrasound image sequences using a deformable mesh. IEEE Trans Med Imaging 17(6):945–956
29. Abolmaesumi P, Salcudean SE, Zhu Wen-Hong, Sirouspour MR, DiMaio SP (2002) Image-guided control of a robot for medical ultrasound. IEEE Trans Robot Autom 18(1):11–23
30. Ortmaier T, Vitrani M-A, Morel G, Pinault S Robust real-time instrument tracking in ultrasound images for visual servoing. In: Proceedings of the 2005 IEEE international conference on robotics and automation, pp 2167–2172
31. Farnebäck Gunnar (2003) Two-frame motion estimation based on polynomial expansion. In: Scandinavian conference on image analysis, pp 363–370
32. Idzenga T, Gaburov E, Vermin W, Menssen J, De Korte C (2014) Fast 2-D ultrasound strain imaging: the benefits of using a GPU. IEEE Trans Ultrason Ferroelectr Freq Control 61(1):207–213
33. Castaño-Díez D, Moser D, Schoenegger A, Pruggnaller S, Frangakis AS (2008) Performance evaluation of image processing algorithms on the GPU. J Struct Biol 164(1):153–160