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

Maxim Makhinya and Orcun Goksel

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

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

1 Introduction

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

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

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

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

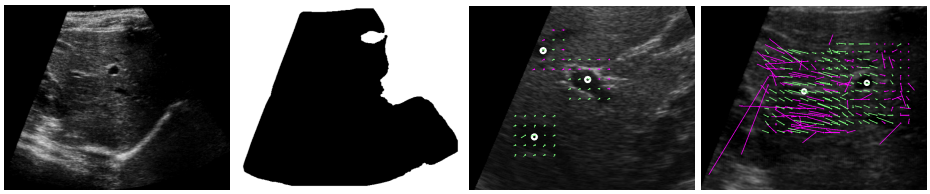
2 Methods

2.1 Motion Tracking

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

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

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



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

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

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

2.2 Vessel Size and Position Refinement

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

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

2.3 Template-based Reset

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

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

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

2.4 Motion Tracking Recovery

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

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

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

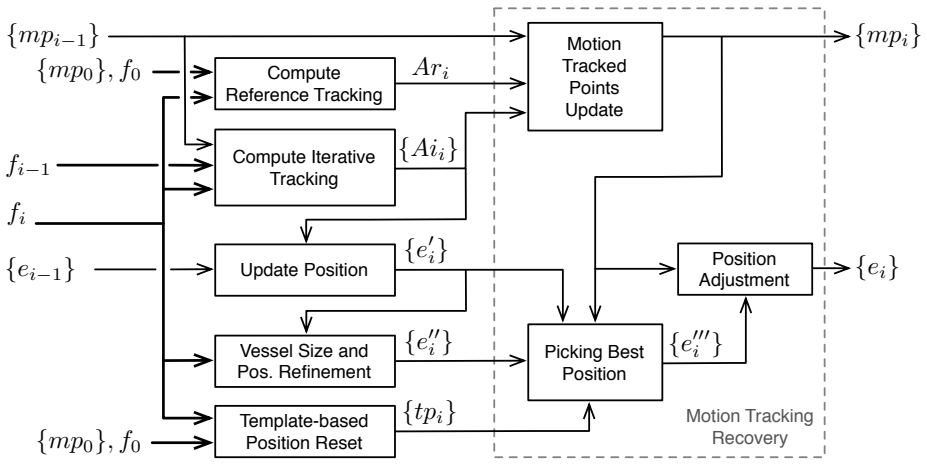


Fig. 2: Algorithm overview.

Algorithm 1 Motion Tracking Recovery

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

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

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

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

3 Results and Discussion

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

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

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

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

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

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

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

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

Individual Scores

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

Individual Scores

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

Average scores per set

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

Average scores for all POI

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

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

4 Conclusions

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

References

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