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 and 1.98 ± 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

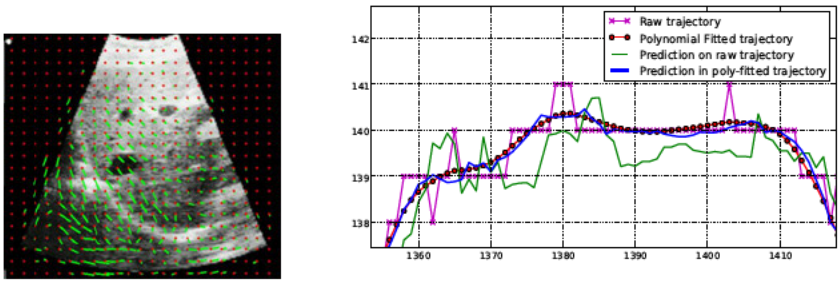
2.1 Datasets

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.