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 $\boldsymbol{x_t^d}$.

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

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.

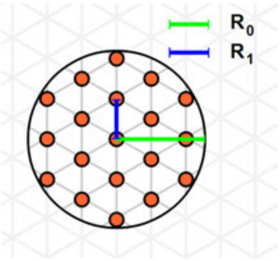
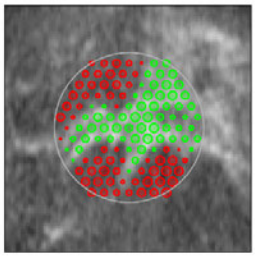


Fig. 1. Initialization: (Left) Within radius R_0 of a given initial position node points on a local triangular grid with grid constant R_1 are chosen. (Right) Sample initialization of point weights in a first frame: Area indicates value and color encodes sign (red: negative, green: positive) of the effective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

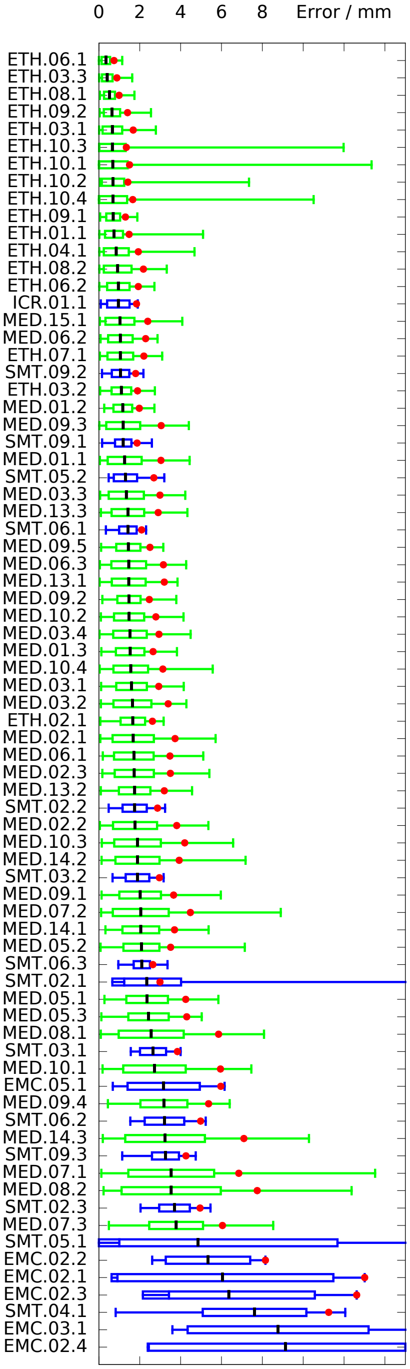
Performance Considerations For each frame computation time is constant, as the amount of computations needed is fixed. Most of the computation is spent for transforming positions for each sample and each point in the observation model. Main contribution of computation time of tracking is given by

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

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.

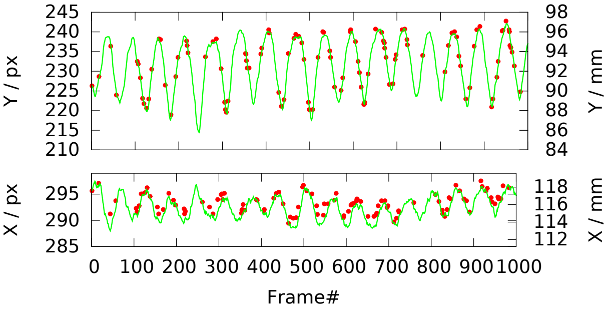


Fig. 3. Sample run on training case ETH-05_2: $S_1^{Tr} = 1$, $R_0 = 13$,$R_1 = 2$. Tracking result (green) and ground truth (red dots).

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.

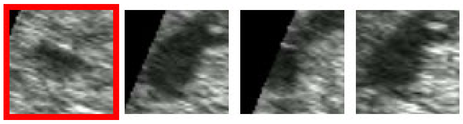


Fig. 4. Sample images of a difficult training sequence (ETH-04_3) in which the target changes the original shape (red) and repeatedly leaves the field of view.

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.

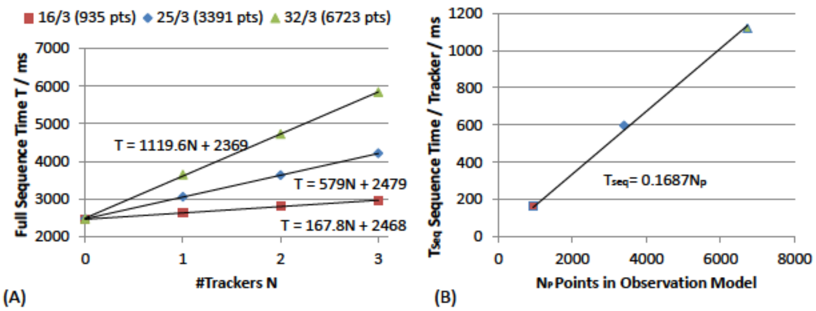


Fig. 5. Performance evaluation on 3D ultrasound data (SMT-02): (A) Time needed to track a 50 volume sequence using 0 to 3 trackers with $N_S = 200$, $F_T = 1$ and differently sized observation models ($R_0 = 16{,}25{,}32$, $R_1 = 3$). The time offset of approximately 2350ms is related to volume loading 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