Real-time Kernel-based Multiple Target Tracking for Robotic Beating Heart Surgery

Gerhard Kurz[1], **Marcus Baum**[2], and **Uwe D. Hanebeck**[1]
[1]Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Karlsruhe Institute of Technology (KIT), Germany
[2]University of Connecticut, USA
gerhard.kurz@kit.edu, baum@engineer.uconn.edu, uwe.hanebeck@ieee.org

***Abstract*—Performing surgery on the beating heart has significant advantages for the patient compared to traditional heart surgery on the stopped heart. A remote-controlled robot can be used to automatically cancel out the movement of the beating heart. This necessitates precise tracking of the heart surface. For this purpose, we track 24 identical artificial markers placed on the heart. This creates a data association problem, because it is not known which measurement was obtained from which marker. To solve this problem, we apply a multiple target tracking method based on a symmetric kernel transformation. This method allows efficient handling of the data association problem even for a reasonably large number of targets. We demonstrate how to implement this method efficiently. The proposed approach is evaluated on in-vivo data of a real beating heart surgery performed on a porcine beating heart.**

***Index Terms*—symmetric measurement equation, data association, medical image processing**

I. Introduction

We consider a medical application of multiple target tracking. In 2001, Nakamura et al. suggested a remote-controlled robot could be employed to automatically cancel out the motion of the heart during beating heart surgery [1]. The suggested system would have significant advantages for the patient, because in conventional operations, the heart has to be stopped, which can cause various additional health risks. To create such a system, accurate tracking of the beating heart is crucial [2]. For this purpose, we place 24 closely spaced identical markers on the heart surface (see Fig. 1). We suggest the use of the Kernel-SME (symmetric measurement equation) method [3] to track multiple markers while considering data association uncertainties.

Simultaneous tracking of multiple targets is of interest in a variety of applications, such as air traffic surveillance, image processing, robotics, and biomedical research [4]. The primary challenge when tracking multiple targets is the association of measurements to the target from which they originate, i.e., the employed sensor gives a set of measurements but is not able to distinguish which target is responsible for a given measurement. This data association problem can be addressed with several different techniques.

On the one hand, there are solutions that explicitly enumerate all association hypotheses, such as the JPDAF (joint probabilistic data association filter) [5], MHT (multiple hypothesis tracker), and some random finite set (RFS) methods [6]. Unfortunately, enumeration of all association hypotheses becomes infeasible for a large number of targets, because their number, and thus the algorithmic complexity, is exponential in the number of targets. There are some approaches to deal with this complexity such as Monte Carlo methods [7]. On the other hand, there are implicit methods that avoid excplicit consideration of all association hypotheses, such as the PHD (probability hypothesis density) filter [8].

This paper is based on another implicit method, the SME approach [9], [10]. In contrast to the plain (C)PHD filter, the SME approach works with the entire joint target state vector. This has the advantage compared to the PHD filter that targets in the state vector remain identifiable across multiple time steps even though the association between state and measurement is not known. The idea of the SME approach is to apply a symmetric transformation to the measurement, i.e., the transformation is invariant to permutation of the targets. More specifically, we use a kernel-based symmetric transformation as introduced in [3].

Our contribution is the following. First of all, we demonstrate how the Kernel-SME method can be implemented efficiently to allow real-time tracking of more than 20 closely spaced targets (markers on the heart surface, in our case). Furthermore, we apply this multi-target tracking approach to real data for the first time and evaluate it in a medical setting. Finally, we extend the analytic solution given in [3] to allow for correlated targets, i.e., state covariance matrices that are not block-diagonal.

II. System and Measurement Model

We assume that the number of targets is known a priori and consider N targets with n-dimensional state vectors $\underline{x}_k^1, \dots, \underline{x}_k^N \in \mathbb{R}^n$. The state vector of the system is given by $\underline{x}_k = [(\underline{x}_k^1)^T, \dots, (\underline{x}_k^N)^T]^T \in \mathbb{R}^{n \cdot N}$. In the considered application, we assume $n = 2$ and the state vector consists of the coordinates of all markers. The state evolves according to a linear system model

$$\underline{x}_{k+1} = \mathbf{A}\underline{x}_k + \underline{w}_k$$

with independent Gaussian process noise $\underline{w}_k$. The system input is implicitly included by state augmentation. The linear measurement model for the l-th target is given by

$$\underline{y}_k^{\pi_k(l)} = \mathbf{H}_k^l \underline{x}_k^l + \underline{v}_k^l \ ,$$

where $\underline{v}_k^l$ is independent Gaussian measurement noise and $\mathbf{H}_k^l$ is the measurement matrix (in our case, $\mathbf{H}_k^l$ is the identity matrix). In the considered application, the association is unknown because all of the markers on the heart surface are identical and closely spaced. In order to model the unknown association between targets and measurements, we use a permutation $\pi_k(\cdot)$ of $\{1, \ldots, N\}$, i.e., $\pi_k(\cdot)$ is a bijective map. We propose the use of a kernel-based symmetric transformation in order to deal with this data association uncertainty.

III. The Kernel-SME Method

In this section, we summarize the Kernel-SME method as introduced in [3].

A. Symmetric Transformation

A symmetric transformation is a function $S(\underline{y})$ that maps the measurement vector to some transformed vector such that $S(\underline{y}) = S(\pi(\underline{y}))$ for any permutation π. This means that $S(\cdot)$ is invariant under permutation. Practically useful transformation functions have the additional property to retain any information contained in $\underline{y}$ except the target association.

The approach of using a symmetric transformation has previously been suggested in [9] with symmetric polynomial functions. However, this transformation function does not perform well for a large number of targets or dimensions, because it involves polynomials of order N, which leads to strong nonlinearities. For this reason, we use the kernel transformation function as suggested in [3]. The idea of the kernel transformation is to transform the measurement to a space of functions, more precisely the space of unnormalized n-dimensional Gaussian mixtures. This is achieved by defining

$$S(\underline{y}) := F_{\underline{y}}, \quad F_{\underline{y}}(\underline{z}) = \sum\nolimits_{l=1}^{N} \mathcal{N}(\underline{z}; \underline{y}^i, \mathbf{\Gamma}) \tag{1}$$

for a certain kernel width $\mathbf{\Gamma}$. It is easy to verify that this transformation fulfills the criterion of symmetry, because addition is commutative, i.e., $\sum_{l=1}^{N} \mathcal{N}(\underline{z}; \pi(\underline{y})^i, \mathbf{\Gamma}) = \sum_{l=1}^{N} \mathcal{N}(\underline{z}; \underline{y}^i, \mathbf{\Gamma})$.

B. Test Vectors

Applying the kernel transformation results in an estimation problem where measurements are functions. We simplify this problem by choosing test vectors and thus, discretizing the problem. Two functions $f_1 : A \to B, f_2 : A \to B$ are identical if $f_1(a) = f_2(a),\ \forall a \in A$, i.e., the mapping is identical for all arguments. Since it is infeasible to evaluate functions at an infinite number of points, we choose a set of N_a test vectors $\underline{a}_k^1, \ldots, \underline{a}_k^{N_a}$. This simplifies the infinite-dimensional estimation problem to a finite-dimensional estimation problem.

In order to choose the test vectors, we use the procedure described in [3]. The state vector is propagated through the measurement equation and the symmetric transformation. Then, a sampling scheme is used on each of the Gaussian mixture components, to obtain $2n+1$ test vectors per target, for a total of $N \cdot (2n+1)$ test vectors.

C. Analytic Solution

An analytic solution for calculating the measurement mean and covariance as well as state and measurement cross-covariance is given in [3]. A MATLAB implementation of this algorithm can be found at [11]. However, this implementation is limited to block-diagonal covariance matrices and is not optimized for performance.

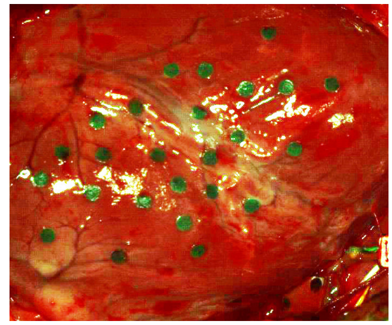

Fig. 1: Heart surface with indistinguishable markers.

The equation for the measurement covariance (23) in [3] is only valid for block-diagonal state covariance matrices, so we use a more general formula in this paper. The measurement covariance for test vectors i and j is given by

$$\mathbf{\Sigma}_k^{s_i s_j} = \left(\sum_{l=1}^{N} \sum_{m=1, m \neq l}^{N} \mathcal{N}\left(\begin{bmatrix} \underline{a}_k^i \\ \underline{a}_k^j \end{bmatrix}; \begin{bmatrix} \mathbf{H}_k^l \underline{\mu}_k^{x_l} \\ \mathbf{H}_k^m \underline{\mu}_k^{x_m} \end{bmatrix}, \mathbf{\Sigma}_k^{lm} \right) \right)$$
$$+ \mathcal{N}\left(\underline{a}_k^i; \underline{a}_k^j, 2\mathbf{\Gamma} \right) \sum\nolimits_{l=1}^{N} P_l^{0.5\mathbf{\Gamma}} \left(\frac{\underline{a}_k^i + \underline{a}_k^j}{2} \right) - \underline{\mu}_{k,i}^s \cdot \underline{\mu}_{k,j}^s \ ,$$

where

$$\mathbf{\Sigma}^{lm} = \begin{bmatrix} \mathbf{H}_k^l & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_k^m \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{k|k-1}^{x_l} & \mathbf{\Sigma}_{k|k-1}^{x_{lm}} \\ \mathbf{\Sigma}_{k|k-1}^{x_{ml}} & \mathbf{\Sigma}_{k|k-1}^{x_m} \end{bmatrix} \begin{bmatrix} \mathbf{H}_k^l & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_k^m \end{bmatrix}^T$$
$$+ \begin{bmatrix} \mathbf{\Sigma}_k^{v_l} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_k^{v_m} \end{bmatrix} + \begin{bmatrix} \mathbf{\Gamma} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma} \end{bmatrix} \ ,$$

and $P_l^{\mathbf{\Gamma}}(\underline{z}) = \mathcal{N}\left(\underline{z}; \mathbf{H}_k^l \underline{\mu}_k^{x_l}, \mathbf{H}_k^l \mathbf{\Sigma}_{k|k-1}^{x_k} (\mathbf{H}_k^l)^T + \mathbf{\Sigma}_k^{v_l} + \mathbf{\Gamma} \right)$. The derivation is similar to the derivation given in the appendix of [3] and relies on manipulation of normal densities and exploitation of the Kalman filter update formulas.

Based on the analytic formulas for measurement mean $\underline{\mu}_k^s$ and covariance $\mathbf{\Sigma}_k^{ss}$ as well as state and measurement cross-covariance $\mathbf{\Sigma}_k^{xs}$, we perform a Kalman filter update according to

$$\underline{\mu}_k^x = \underline{\mu}_{k|k-1}^x + \mathbf{\Sigma}_k^{xs} (\mathbf{\Sigma}_k^{ss})^{-1} (F_{\underline{y}_k}(\underline{z}_k)) - \underline{\mu}_k^s) \ ,$$
$$\mathbf{\Sigma}_k^x = \Sigma_{k|k-1}^x - \mathbf{\Sigma}_k^{xs} (\mathbf{\Sigma}_k^{ss})^{-1} (\mathbf{\Sigma}_k^{xs})^T \ ,$$

where $F_{\underline{y}_k}(\underline{z}_k))$ is the transformed measurement $\underline{y}_k$ with test vectors $\underline{z}_k$ according to (1).

IV. Real-time Implementation

In order to perform real-time processing of more than 20 targets, we had to do a variety of optimizations. Because the double summation in the equation above is required for every entry of the measurement covariance matrix, this approach scales with $\mathcal{O}(N^4)$ (see Fig. 2). Because the calculation of the measurement mean is in $\mathcal{O}(N^2)$ and the calculation of the cross-covariance between state and measurement is in $\mathcal{O}(N^3)$, the computation of the measurement covariance is the most costly operation for a large number of targets N. It is worth

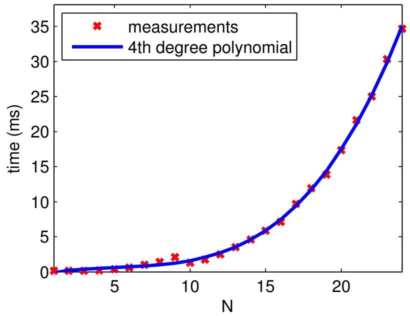

Fig. 2: Time for analytic solution depending on number of markers N. We fitted a fourth degree polynomial to the data to illustrate the fact that the computational effort is in $\mathcal{O}(N^4)$.

mentioning that the formula for the measurement covariance in the case of uncorrelated targets as given in [3] can be evaluated in $\mathcal{O}(N^3)$, but we need to consider correlation in our application. Therefore, we focus on optimization of this computation when trying to achieve real-time performance. We assume that the number of dimensions is $n = 2$, and the measurement matrices $\mathbf{H}_k^l$ are all identity matrices. These assumptions hold in our particular application, but it is easy to drop them if necessary.

When implementing the calculation of the measurement covariance matrix, we take advantage of the following properties. A covariance matrix is obviously symmetric, so it is sufficient to calculate $\mathbf{\Sigma}_k^{s_i s_j}$ for $j \geq i$ and obtain the remaining entries according to $\mathbf{\Sigma}_k^{s_i s_j} = \mathbf{\Sigma}_k^{s_j s_i}$. Furthermore, $\mathbf{\Sigma}^{lm} \in \mathbb{R}^{4\times 4}$ is independent of i and j and can thus be calculated beforehand, along with its inverse. To evaluate the normal distribution, the normalization constant needs to be computed, which can be done in advance as well, because it only depends on the determinant of $\mathbf{\Sigma}^{lm}$.

We implemented the analytic moment calculation as a MATLAB MEX-file in C++. For matrix operations, we used the highly optimized Eigen library [12], which uses closed-form expressions for 4×4 inverse and determinant. The exponential function occurring in the normal distributions was computed using AVX2 with fmath [13]. Furthermore, we parallelized the loop over i with OpenMP [14], because the computations of different entries of the measurement covariance are mutually independent. This results in almost linear speedup with the number of available CPU cores.

V. Evaluation

We evaluated the proposed Kernel-SME method on real data from a beating heart surgery. The experiments were conducted on a porcine heart at Heidelberg University Hospital. A set of indistinguishable markers was placed on the beating heart (see Fig. 1) and the heart was recorded with a PIKE F-210C Camera at a resolution of 1920×1080 and 30 frames per second. The target positions were obtained from the images using a color-based segmentation algorithm and calculating the center of mass for each connected area of appropriate size. This can sometimes result in clutter or missing measurements, but we

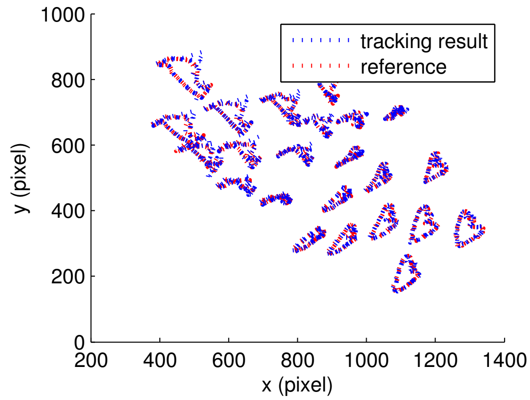

Fig. 3: Tracked trajectories and reference.

did not explicitly model these effects. We also recorded arterial blood pressure at a frequency of 1 kHz and downsampled it to 30 Hz. The pressure data is used as an input to the system model whereas the camera data is used as a measurement.

For the purpose of evaluation, we consider $N = 24$ targets, which are to be tracked in $n = 2$ dimensions. Because all markers are identical, data association is not trivial in this application. In order to truly evaluate the multi-target tracking capabilities of the proposed method, we do not use any kind of gating to reduce the number of association hypotheses.

We obtained the reference trajectories semi-manually with a simple nearest-neighbor based tracking algorithm that was carefully tuned to avoid incorrect associations for the considered time period[1]. Still, some of the markers were missing in a few frames and their reference position was determined by interpolation. The system model was assumed to be linear and the time-invariant system matrix $\mathbf{A}$ was obtained from 200 time steps of the reference by solving a linear least squares problem. The same data was then used to estimate the process noise covariance matrix $\mathbf{\Sigma}_k^w$. We manually chose $\mathbf{\Sigma}_k^v = \mathbf{I}_{n\cdot N}$.

To evaluate the quality of the results, we ran the algorithm for 500 time steps, which corresponds to a video sequence of approximately 17 s. The state vector was initialized with the measurements of the first frame and we chose the kernel size $\mathbf{\Gamma} = \mathrm{diag}(1000, 1000)$. The resulting target trajectories in comparison with the reference are depicted in Fig. 3.

For any marker l and any time step k, we consider the Euclidean distance between the estimate $\underline{x}_l$ and the true position. The mean across all markers and time steps is 0.9550 pixels and the median is 0.3297 pixels, i.e., we achieve subpixel accuracy and the error has a similar magnitude to the segmentation accuracy.

In order to evaluate the run-time of the proposed algorithm, we used a computer with an Intel Core i7-4770 quad core CPU running at 3.4 GHz, 16 GB RAM and MATLAB 2013b 64bit along with the Microsoft Visual Studio 2012 compiler. We measured the duration of the analytic calculation of the moments for different numbers of targets and give the results

[1]This algorithm is not applicable in practice because it can not even recover from a single incorrect association.

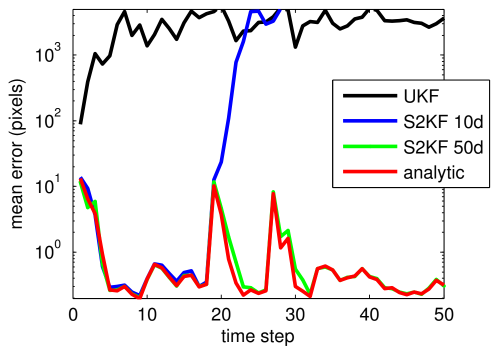

Fig. 4: Comparison of filters: UKF, S2KF with $10d$ and $50d$ samples, as well as the analytic solution. The vertical axis is logarithmic and gives the mean error over all tracked targets.

in Fig. 2. It can be seen, that the performance degrades with the fourth power of the number of targets, as is to be expected. The computation time for our considered scenario with 24 targets is around 35 ms with all optimizations. Because our algorithm is highly parallelizable, this time can easily be lowered further by employing more CPU cores or even implementing the algorithm on a GPU. For comparison, the MATLAB implementation without any optimizations takes 977 s on the same hardware.

Furthermore, we compared the analytic solution to a measurement update with sample-based filters, namely the UKF [15] and the S2KF [16]. The advantage of sample-based filters is the fact that the algorithmic complexity scales with $\mathcal{O}(N^3)$ rather than $\mathcal{O}(N^4)$ if the number of samples is linear in N. As we have a state of dimension $n \cdot N$ and non-additive noise of dimension $n \cdot N$, the augmented state vector is of dimension $d = 2n \cdot N$. The UKF has a fixed number of $2d + 1$ samples, whereas the number of samples used by the S^2KF can freely be chosen. We use $10d$ and $50d$ samples. The results are depicted in Fig. 4. As can be seen, the UKF is unable to handle this scenario and diverges immediately. The S^2KF also diverges at some point, if the number of samples is too small. As is to be expected, the analytic solution performs best, but for a large number of samples the S^2KF gives similar results to the analytic solution.

VI. Conclusion

We have applied the Kernel-SME method for multiple target tracking from [3] to marker tracking on the heart surface. It can be implemented in a way that is suitable for real-time applications. The proposed method has been evaluated on a real-world medical dataset. It was shown to be possible to reliably track 24 unlabeled targets in real-time. Accurate tracking of indistinguishable markers on the heart surface is a significant step towards robotic beating heart surgery.

Future work may include a closer investigation of the influence of the chosen kernel width and an improved algorithm for selecting the test vectors. Furthermore, we plan to extend the proposed method to 3D tracking of markers on the heart surface. This can be achieved by performing triangulation with a stereo camera system or with the help of depth sensors such as time-of-flight cameras or the Microsoft Kinect. It may also be possible to exploit the quasi-periodic nature of the problem.

Acknowledgment

This work was partially supported by the German Research Foundation (DFG) within the Research Training Groups RTG 1126 "*Intelligent Surgery - Development of new computer-based methods for the future working environment in visceral surgery*". We would like to thank Peter Hegedüs and Gábor Szabó for their contributions to the medical experiment. Furthermore, we would like to thank Geneviève Foley for her work on analyzing the medical data.

References

[1] Y. Nakamura, K. Kishi, and H. Kawakami, "Heartbeat Synchronization for Robotic Cardiac Surgery," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)*, Seoul, Korea, May 2001, pp. 2014–2019.

[2] E. Bogatyrenko, P. Pompey, and U. D. Hanebeck, "Efficient Physics-Based Tracking of Heart Surface Motion for Beating Heart Surgery Robotic Systems," *International Journal of Computer Assisted Radiology and Surgery (IJCARS 2010)*, vol. 6, no. 3, pp. 387–399, Aug. 2010.

[3] M. Baum and U. D. Hanebeck, "The Kernel-SME Filter for Multiple Target Tracking," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[4] G. Pulford, "Taxonomy of multiple target tracking methods," *Radar, Sonar and Navigation, IEE Proceedings*, vol. 152, no. 5, pp. 291–304, 2005.

[5] Y. Bar-Shalom, F. Daum, and J. Huang, "The Probabilistic Data Association Filter," *IEEE Control Systems*, vol. 29, no. 6, pp. 82–100, 2009.

[6] B.-T. Vo and B.-N. Vo, "Labeled Random Finite Sets and Multi-Object Conjugate Priors," *Signal Processing, IEEE Transactions on*, vol. 61, no. 13, pp. 3460–3475, July 2013.

[7] S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for Multi-Target Tracking," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 481–497, Mar. 2009.

[8] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[9] E. Kamen, "Multiple Target Tracking based on Symmetric Measurement Equations," in *American Control Conference, 1989*, 1989, pp. 2690–2695.

[10] M. Baum, B. Noack, F. Beutler, D. Itte, and U. D. Hanebeck, "Optimal Gaussian Filtering for Polynomial Systems Applied to Association-free Multi-Target Tracking," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[11] M. Baum, "Kernel-SME Filter Matlab Implementation," CloudRunner Web Page, Apr. 2013. [Online]. Available: http://www.cloudrunner.eu/algorithm/117/kernel-sme-filter/version/1/

[12] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2013.

[13] M. Shigeo, "Fast approximate float function fmath," 2013. [Online]. Available: https://github.com/herumi/fmath

[14] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE Computational Science Engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[15] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[16] J. Steinbring and U. D. Hanebeck, "S2KF: The Smart Sampling Kalman Filter," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.