Temporal Calibration in Multisensor Tracking Setups

Manuel Huber, Michael Schlegel, Gudrun Klinker

Technische Universität München
Fakultät für Informatik
Boltzmannstraße 3
85748 Garching b. München, Germany

Abstract: Spatial tracking is one of the most challenging and important parts of mixed reality environments. Many applications, especially in the domain of Augmented Reality, rely on the fusion of several tracking systems in order to optimize the overall performance. While the topic of sensor fusion has already seen considerable interest, most results only deal with the integration of particular setups as opposed to dynamic sensor fusion setups. A crucial prerequisite for correct sensor fusion is the temporal alignment of the sensor signals, as sensors in general are not synchronized. We present a general method to calibrate the temporal offset between different sensors which can be used to perform on-line calibration. To show the correctness and the feasibility of this method, we evaluated various combinations of tracking sensors.

Keywords: sensor fusion, calibration, tracking, ubiquitous tracking, synchronization

1 Introduction

One of the most active topics in the research field of mixed and especially Augmented Reality is the area of determining the pose of the user and of physical objects with which the user interacts. This is generally referred to as the *tracking problem*. Especially applications involving head mounted augmentations (for example via HMD) require a high level of tracking accuracy in order to render convincing visualizations.

In order to improve the overall accuracy, the concept of sensor fusion was early introduced to solve various tracking problems. Sensor fusion aims to improve the data quality by measuring the same or related physical properties by multiple sensors and combining their data to ideally obtain an improved measurement. The most commonly used competitive methods to fuse data currently include Kalman-based filtering [WB04] and particle filters [DDFG01].

1.1 Temporal Calibration

In order to correctly combine data from two sensors, it is necessary to know the exact temporal relationship between data acquired from the different sources. For example a concatenation of poses may only be performed on data that refers to the same moment in time of the physical reality. We call such sensors to be synchronized or temporally calibrated.

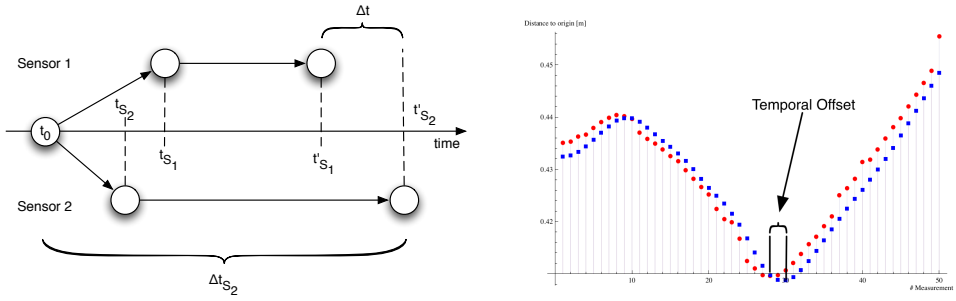
Synchronization can either be achieved by hardware means or by logical means on the sensor

data. For hardware synchronization the acquisition of sensor data is triggered by a central hardware clock (trigger signal), connected to all participating sensors. Synchronization in software on the other hand depends on correctly attaching timestamps to each sensor measurement. Such timestamps are preferably provided by the sensor itself or have to be generated by the tracking framework as soon as the measurement enters the system [Pus06]. In either case the timestamps of the different sensors have to be adjusted to refer to a common time base.

1.2 Relative Sensor Lag

The problem of adjusting timestamps between sensors arises for arbitrary numbers of sensors. For simplicity we state the problem for a single pair of sensors S_1 and S_2, but the principle can be extended to n sensors. If each sensor pair is calibrated accordingly, this results in $\binom{n}{2}$ calibrations.

Considering an event happening in the real world at time t_0, two sensors S_1 and S_2 sense this event as an analog physical input and convert it into digital representations at times t_{S_1} and t_{S_2}. Further delay will be caused by the various communication stacks of the operating system or by network transport of the measurement data. We assume the measurements to arrive at the tracking framework at times t'_{S_1}, t'_{S_2} (see figure 1(a)) and to be tagged with according timestamps at that point in time. As soon as a reliable timestamp is attached, all further processing delays can be managed by the tracking framework. For sensor fusion it is only necessary that all sensors are temporally aligned relative to each other; the offset to the unknown true point in time t_0 is not relevant. To align the sensor data, it is sufficient to determine the temporal offset $\Delta t = t'_{S_1} - t'_{S_2}$.



(a) (b)

Figure 1: (a) Schematic visualization of the relevant points in time; (b) Temporal offset in sensor data from two sensors.

In this paper we present a general method to calibrate the temporal offset between different sensors. This method can either be applied as a one-time calibration or it can be used as an online recalibration method to adapt to sensor setups that can change at runtime or to compensate for changing environmental influences (such as thermal clock drift or network delay). This also enables calibration of ad-hoc sensor fusion setups in highly flexible ubiquitous tracking scenarios, where no a priori knowledge about the involved sensors is available. Further discussions on this calibration method can also be found in [Sch11] and [Hub11].

Another area where synchronization of sensors is important, is the error evaluation of tracking setups where the data measured by a tracking system under test is compared to ground-truth data

from a second, more precise system. A temporal offset between the system under test and the reference data causes an additional tracking error which disturbs the actual experiment. Such a scenario is described in [GGV+10].

2 Related Work

The negative influence of lag on the usability of AR applications is generally agreed upon (see for example [AB94, Hol97, WB97]). Nevertheless, temporal calibrations considerations so far are mostly limited to particular hardware setups. Components are either synchronized by hardware clocks (for example [BS08]) or the lag between different sensors is tuned in software by experimental means.

The introduction of Ubiquitous Augmented Reality [NWB+04] has lead to the Ubiquitous Tracking problem and the need for dynamic sensor fusion to cope with the corresponding, highly dynamic scenarios. In [PHBK06] and [HPK+07] a general framework was introduced which aims to account for this by using a pattern based approach working on spatial relationship graphs. This framework so far accounts for unsynchronized sensors by utilizing a Push/Pull dataflow architecture [Pus06] which depends on the correctness of timestamps associated with sensor measurements.

In [ASB07] a sensor synchronization scheme is discussed for the application of calibrating inertial sensors and vision based tracking. Their approach relies on detecting abrupt movements in both the camera image as well as the inertial tracker.

In [LBMN09, SS04] a static temporal offset was calibrated in a static manner during the spatial calibration process. They shift one signal in time while constantly calculating the geometric residuum. The temporal offset which minimizes the geometrical residuum is then taken as the temporal offset (see section 4.2 for a comparison with our approach).

In [JLS97] an in-depth study of latencies occurring in an AR-System was conducted. In addition to the measurement of the overall latency they also performed a calibration of the offset of two sensors during setup time. Instead of using mathematical optimization methods, an AR-System built and used to manually adjustable prediction to determine the temporal offset between the used sensors by visual means.

3 Calibration Method

3.1 General Strategy

Let T_{S_1}, T_{S_2} be the sets of all timestamps $t \in T_{S_1} \cup T_{S_2}$ where measurements of either S_1 or S_2 respectively were taken. We define the two data time series $X = \{x_t : t \in T_{S_1}\}$ and $Y = \{y_t : t \in T_{S_2}\}$ as the actual sensor data x_t, y_t from sensors S_1 and S_2 respectively at the individual timestamps. Note that the timestamps at which S_1 and S_2 acquire data, are in general not the same and thus T_{S_1} and T_{S_2} may be distinct.

For each pair of signals a suitable similarity measure $\rho_{X,Y}$ can be computed that measures the mutual agreement of the signals. We assume the similarity measure to be normalized such that $0 \leq \rho_{X,Y} \leq 1$. We call the signals *orthogonal* if $\rho_{X,Y} = 0$ or *identical* if $\rho_{X,Y} = 1$. In general $\rho_{X,Y} \neq 1$ even if the same type of sensors is used since both measurements will be individually affected by noise and other kinds of errors.

The time-offset of the two sensor signals can be determined by consecutively shifting one signal by small offsets against the other signal until a maximum of agreement is reached. The shift value Δt for which this maximum is attained is identified as the temporal offset between the sensors. This is a well-known approach in signal processing [Car81] and can be written as

$$\Delta t = \text{argmax}_{\delta t}\{\rho_{X,Y^{(\delta t)}}\},$$

where $Y^{(\delta t)}$ is the signal Y shifted in time by δt.

3.2 Various properties of Time Series

Time series have a number of different properties which can have different impacts on the performance of the calibration procedure.

Registration In order for the signals of two different sensors to be comparable we assume that the static registration between the two sensors is known and that the sensor measurements have been transformed into a common coordinate frame. Note that since the temporal calibration process is rather robust against spatial registration errors, the requirements on the accuracy is rather low.

Signal-to-noise ratio For tracking sensors the Signal-to-noise ratio (SNR) describes the amount of movement present in the signal compared to the measurement noise of the sensor. Time series with large SNR usually feature large movements or fast velocities, whereas low SNR indicates little activity or very slow movement. Thus the noise characteristics of the sensor also determines the minimum movement required to produce a signal exhibiting sufficient SNR.

Sampling rate Another basic assumption about the tracking data is that each sensor acquires its corresponding measurement of the real world at periodic intervals in time. Each measurement is called a sample point of the sensor and the periodicity of these samples is called the sampling rate. Thus the sampling rate is a basic characteristic of the sensor that also determines the maximum temporal resolution of tracked movements.

3.3 Overview

The sensor data that is acquired by positional sensors usually features high (commonly 3 or 6 DoF) dimensionality. In the course of the proposed calibration method we reduce these measurements to one dimensional signals. The rationale behind this is that while reducing the physical

representational precision of the sensor signal, the time calibration primarily utilizes the shape of the sensor signals. Similar to the situation in machine learning, the reduction in dimensionality may actually enhance the performance of the time calibration by making the characteristics of the sensor movement more prevalent. This results in improved behavior in low SNR settings, as will be discussed later. Also computing the similarity on signals of reduced dimensionality is computationally faster, which enables real-time on-line calibration in the first place. Furthermore the adaptation of the calibration process to different pairs of sensors is simplified, even in cases where no immediate geometrical comparison of the sensor data is available. The following sections discuss the required steps of our proposed calibration method.

3.4 Segmentation

The sensor data is assumed to be an endless stream of measurements. Thus, as a first step, this data is divided into chunks of equal length (duration). This serves two specific purposes. Shorter chunks of data are easier to process, both in terms of speed and complexity. Also the significance of the comparison of very large chunks tends to decrease due to the increased existence of tracking outliers. Second any on-line estimation requires some sort of segmentation in order to produce results during the runtime of the procedure.

The two basic strategies are to either produce subsequent disjoint segments or to produce overlapping segments, where each new segment consists of a certain amount of old data with new data appended. Note that the number of samples in the corresponding chunks of two sensors may not be equal, especially samples rates of the sensors differ.

3.5 Dimensionality reduction

As previously discussed, the optimization is not perfomed on high dimensional tracking data, but rather on one dimensional, reduced signals.

We will limit the discussion of the projection methods to the sensor S_1 and assume that the same projection is used for the data of sensor S_2. In general we search for a mapping $f : \mathbb{R}^n \mapsto \mathbb{R}$ that reduces the high dimensional measurement data x_t to a 1 dimensional signal $\hat{x}_t = f(x_t)$ for all timestamps t. To achieve this goal a number of different approaches are feasible, which we classify as follows.

Static computations A simple, yet useful method to reduce the dimensionality is to use a static projection or computation for each measurement. One example of this class of reduction is just taking the x-component of a 3D position measurement by the projection $\hat{x}_t = w_x^T x_t$ with projection vector $w_x = (1,0,0)^T$. These kinds of projections suffer from reduced resulting SNR in cases where the movement of the sensor is mainly perpendicular to the projection vector.

Another kind of static computation is to use the Euclidean norm as a mapping function $\hat{x}_t = \sqrt{x_t^T x_t}$. This is equivalent to computing the distances to the origin for 3D position measurements.

This method also works for incremental rotation measurements (as used for gyroscope integration, see also [PK08]).

Adaptive computations A more advanced method of dimensionality reduction incorporates the influence of all measurements in the current segment and adapts the projection vector accordingly. Similar to the static case the projection can be defined as $\hat{x_t} = w_t^T x_t$, where w_t is now dynamically computed for each segment. It is still constant for each segment, and typically the same projection will be used for different sensors. To compute a projection vector the measurements of the current segment can for example be transformed using principal component analysis [Fod02]. This determines the direction of the most significant movement in each segment.

Pathological sensor data While there are many variations possible, experience shows that in practice the time offset calibration is usually robust against the choice of dimensionality reduction. We will shortly discuss the various possible pathological cases which render the dimensionality reduction ineffective and thus the temporal calibration useless.

The simplest pathological case is zero movement, thus no events happening in reality. In this case the signal of the two sensors consist only of the sensor-noise. This obviously leads to useless results and the inability to determine the relative lag.

Apart from this trivial case each dimensionality reduction method may exhibit specific cases, which can indubitably be constructed. For example the simple projection of the 3D sensor position onto one of the primary axes obviously produces unsuitable signals for sensor movements orthogonal that axis.

Yet, we argue that these cases are actually rare in practice, since they require very precise movements. To show this we conducted an experiment with two participants and calculated the relative lag between an infrared system and a coordinate measurement machine (CMM) using only these sensor movements. The task for the participants was to try to keep the sensors as steady as possible, without actually resting their arm on a table or similar. This approximates the trivial pathological case of zero movement.

Due to the physiological (normal) tremor of the hands of the participants, the signal-to-noise ratio of these measurements was already enough to successfully calculate the relative lag. Also the frequency of the physiological tremor (about $6-12\,\text{Hz}$, see [Lip71]) is well within the temporal resolution of both tracking systems. The relative lag was estimated as 32.3 ms with 1.5 ms standard deviation. On the other hand the signal of the reference experiment, where the sensors were fixed with a vice, produced no suitable results as expected.

Thus we argue that while pathological cases for the individual cases do exists, due to the nature of the human interaction these cases are scarcely encountered in practice. Also the use of adaptive projections can help to make such cases even more unlikely. Only in scenarios where the sensors are mounted on computer controlled actuators or robots, special care has to be taken to tune the dimensionality reduction to the data at hand.

3.6 Interpolation

A common assumption is that each signal is represented as measurements that are sampled at equidistant points in time. In practice the sample rate of a single sensor, as seen from the tracking

framework, may not be constant but is subject to jitter and clock noise. Furthermore the relationship between the sample points of the two signals is generally not known. Thus one sample point of the first signal does not directly correspond to one sample point of the second signal. It is thus necessary to create a common basis of sample points for both signals.

Using the originally assigned timestamps for each measurement of either sensor, we interpolate both signals using linear interpolation between the individual sampling points, resulting in continuous signals. These can then be resampled at common sampling points for both signals. Note that in practice it suffices to perform the actual interpolation on only one signal. The sampling points for the similarity computation can conveniently be chosen to coincide with the sampling points of one of the two sensors. If the sampling rates of the sensors differ, it is beneficial to interpolate the signal with the higher sampling rate to minimize interpolation errors.

3.7 Time Delay Estimation (TDE)

After these preprocessing steps we have two one-dimensional signals $\hat{X} = \{\hat{x}_t = f(x_t) : t \in T'\}$ and $\hat{Y} = \{\hat{y}_t = f(y_t) : t \in T'\}$ which have been interpolated and can be assumed to be continuous on the time domain T'.

The actual estimation of the relative lag of these two signals can be performed by a method also known the time delay estimation (TDE). This method is well known and understood in signal processing and is used for applications such as RADAR or SONAR (for example [Car81]). Generally, the aim of this method is to estimate the temporal offset of a specific pattern contained in a usually noisy signal. In our application the offset between these two pattern instances is the relative lag between the sensors. As already mentioned the general procedure of the time delay estimation keeps one signal fixed and shifts the second in time relative to the first. For each possible time shift a similarity measure is computed, which is maximized over all possible shifts.

Cross-Correlation One of the earliest and still most important similarity measures used for such setups is the normalized cross-correlation.

The textbook definition of the normalized correlation coefficient (also called Pearsons' correlation) is

$$\rho_{X,Y} := \frac{E[XY]}{E[X]E[Y]}.$$

Resolution of the TDE The resolution is mainly determined by the step-size of the timeshifts performed, while the determination of the maximum value becomes increasingly less well-defined with decreasing step-size. A common approach is to fit a parabola to the similarity measurements and calculate the time delay estimation as the vertex of this parabola [BH81]. Our experiments suggest 1ms as both a feasible and useful resolution.

3.8 Aggregation

After the time delay estimation has determined the relative lag of the two signals on one segment, multiple segments can be aggregated to identify meaningless results, reject outliers or perform smoothing of the lag calibration. Suitable approaches to aggregate multiple estimates at runtime are the simple moving average, the weighted moving average using the significance parameters as discussed above as weights or the moving median.

4 Evaluation

To validate the method described above we conducted a series of experiments involving different combinations of sensors. We first describe the general hardware setup used as well as the individual experiments undertaken. Finally we demonstrate the effectiveness of this approach by an evaluation of registration errors in both unsynchronized and synchronized sensor fusion.

4.1 Pairwise evaluation setup

The following list summarizes the available hardware used for the experiments in this study.

- The *A.R.T. system* is an optical, infrared outside-in tracking system based on retro-reflective ball markers. Either 6DoF poses for rigid marker constellations or 3DoF positions for single balls can be obtained. The sample tracking setup can be seen in figure 2(a).
- The *Faro Fusion* coordinate measurement machine (CMM) is a high precision measurement device. It produces 6DoF measurements of the position and orientation of the tip of the arm. A picture of the used Faro CMM can be seen in figure 2(b).
- An inside-out *optical square marker tracker* which can track the 6DoF pose of a printed square-marker pattern using an off-the-shelf webcam. This combination can be seen in figure 2(c).

(a) (b) (c)

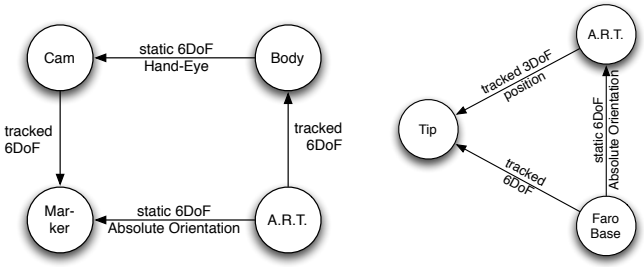
Figure 2: Sensors used for evaluation; (a) A.R.T. infrared tracker; (b) Faro CMM arm

All three combinations of these sensor systems have been evaluated using suitable setups and calibrations. The respective spatial relationships of the three setups are visualized in figure 3. For the combination of Faro CMM vs. Square-Marker tracker the graph has been omitted since it is

Setup	Mean	Std. Dev.
(1) A.R.T. vs. Square Marker	123 ms	65 ms
(2) A.R.T. vs. Faro	32 ms	0.1 ms
(3) Faro vs. Square Marker	84 ms	30 ms

Table 1: Results of temporal calibrations

similar to the A.R.T. vs. Square-Marker tracker case (figure 3(a)), with the Nodes "A.R.T." and "Body" replaced by "Faro Base" and "Tip" respectively. The results of the temporal calibrations are summarized in Table 1. The relatively large standard deviations in setups 1 and 3 stem mostly from pronounced temporal instabilities of the square marker tracker. This is especially evident when compared to the very precise results obtained for setup 2.



(a) (b)

Figure 3: Spatial relationship graphs describing the tracking setup between (a) A.R.T. and Square-Marker tracker and (b) A.R.T. tracker and Faro CMM

Transitivity Another test for the validity of the temporal calibration method is to examine the consistency of setups involving more than two sensors. When combining more than one pair of sensors the relative lag between the individual pairs has to obey transitivity. This is visualized and validated in figure 4 for the combination of the setups as described above.

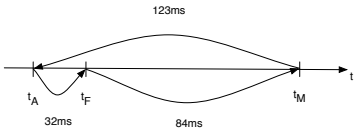


Figure 4: Transitivity of pairwise sensor offsets

4.2 Comparison with geometric error minimization

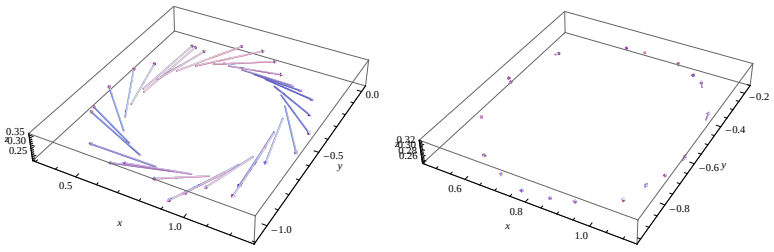
To further validate the correctness of our approach we compared the resulting relative lag of two sensors with the time offset which minimizes the overall geometric error between the sensors. This is similar to the static calibration approach used in [LBMN09, SS04]. The relative lag was similarly estimated by both approaches as 32 ms with 0.1 ms standard deviation. This is expected

since the correct relative lag also reduces the error between the registered coordinate frames of the sensors. This experiment was performed using sensor data with significant SNR. In cases with low SNR, the time delay estimation exhibits more robust behavior than the geometric minimization. We repeated the experiment with the data from the physiological hand tremor experiment (see section 3.5). Here the estimated values for the relative lag were 35 ms (geometric minimization) and 32 ms (correlation) with 39 ms and 0.3 ms respective standard deviation. This illustrates that under these circumstances, the result of the geometric optimization still falls in the same range as before but with massively increased uncertainty. On the other hand, the correlation-based time delay estimation performs only slightly worse than before.

4.3 Error reduction

To illustrate the effectiveness of the temporal alignment, we analyzed the resulting spatial registration error between two different trackers in both the unsynchronized and the synchronized case. The same setup from the A.R.T. vs. Faro CMM case is being used for this analysis. For this data set the tip of the Faro CMM was moved in a simple circle with moderate speed of about 1 m/s (determined afterwards). The 3DoF position of the tip was recorded both by the Faro system and by the A.R.T. system, which was additionally transformed into the Faro coordinate frame. Figure 5(a) shows the error vector between measurements from the A.R.T. system and corresponding points measured by the Faro system during the movement. The root mean square (RMS) error in this case is 32.1 mm. From the direction of the vectors the movement of the marker ball is clearly visible as a systematic misregistration. This indicates a distinctive lag between the two sensor systems.

Figure 5(b) shows the same error vectors after the timestamps were corrected according to the determined temporal calibration. In this plot the direction of the error vectors no longer corresponds to the direction of the movement and the RMS has been reduced to 1.6 mm. The remaining errors mostly stem from calibration errors and sensor noise.



(a) (b)

Figure 5: Error vector between measurements from A.R.T. and the Faro system during movement; (a) without temporal alignment; (b) with temporal alignment; (magnification factor 10)

5 Summary and Future Work

We have presented a method to automatically determine the temporal offset between two tracking sensors, by optimizing a similarity measure between the different sensors' data over a range of

temporal shifts. This solution is especially important for using the data for sensor fusion. We presented an evaluation supporting the feasibility and correctness of the approach was demonstrated as well as the importance of the correction for a registration error analysis. We furthermore integrated this method into the Ubitrack tracking framework and the trackman tracking management tool. A further venue of inquiry is to apply the temporal calibration method to unregistered sensor data and to evaluate other similarity measures.

References

[AB94] Ronald Azuma and Gary Bishop. Improving Static and Dynamic Registration in an Optical See-through HMD. In *SIGGRAPH*, pages 197–204, 1994.

[ASB07] M. Aron, G. Simon, and M.O. Berger. Use of Inertial Sensors to Support Video Tracking. *Computer Animation and Virtual Worlds*, 18(1):57–68, 2007.

[BH81] R. Boucher and J. Hassab. Analysis of Discrete Implementation of Generalized Cross Correlator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(3):609 – 611, jun. 1981.

[BS08] G. Bleser and D. Stricker. Advanced Tracking through Efficient Image Processing and Visual–Inertial Sensor Fusion. *Computers and Graphics*, 2008.

[Car81] GC Carter. Time Delay Estimation for Passive Sonar Signal Processing. *IEEE Transactions on Acoustics Speech and Signal Processing*, 29(3):463–470, 1981.

[DDFG01] Arnaud Doucet, Nando De Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practice*. 2001.

[Fod02] I.K. Fodor. A Survey of Dimension Reduction Techniques. *US DOE Office of Scientific and Technical Information*, 2002.

[GGV+10] Lukas Gruber, Steffen Gauglitz, Jonathan Ventura, Stefanie Zollmann, Manuel Huber, Michael Schlegel, Gudrun Klinker, Dieter Schmalstieg, and Tobias Höllerer. The City of Sights: Design, Construction and Measurement of an Augmented Reality Stage Set. In *Proceedings of the 9th International Symposium on Mixed and Augmented Reality (ISMAR)*, October 2010.

[Hol97] Richard L. Holloway. Registration Error Analysis for Augmented Reality. *Presence*, 6(4):413–432, 1997.

[HPK+07] Manuel Huber, Daniel Pustka, Peter Keitler, Echtler. Florian, and Gudrun Klinker. A System Architecture for Ubiquitous Tracking Environments. In *Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR)*, November 2007.

[Hub11] Manuel Huber. *Parasitic Tracking for Ubiquitous Augmented Reality*. PhD thesis, Technische Universität München, 2011.

[JLS97] M.C. Jacobs, M.A. Livingston, and Andrei State. Managing Latency in Complex Augmented Reality Systems. In *Proceedings of the Symposium on Interactive 3D Graphics*. ACM, 1997.

[LBMN09] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab. A Dataset and Evaluation Methodology for Template-Based Tracking Algorithms. In *Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 145–151. IEEE Computer Society, 2009.

[Lip71] Olof Lippold. Physiological Tremor. *Scientific American*, 224:65–73, March 1971.

[NWB+04] J. Newman, M. Wagner, M. Bauer, A. MacWilliams, T. Pintaric, D. Beyer, D. Pustka, F. Strasser, D. Schmalstieg, and G. Klinker. Ubiquitous Tracking for Augmented Reality. In *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04)*, 2004.

[PHBK06] Daniel Pustka, Manuel Huber, Martin Bauer, and Gudrun Klinker. Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems. In *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06)*, October 2006.

[PK08] Daniel Pustka and Gudrun Klinker. Dynamic Gyroscope Fusion in Ubiquitous Tracking Environments. In *Proceedings of the 7th International Symposium on Mixed and Augmented Reality (ISMAR)*, 2008.

[Pus06] Daniel Pustka. Construction of Data Flow Networks for Tracking in Augmented Reality Applications. In *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, Koblenz, Germany, September 2006.

[Sch11] Michael Schlegel. *Zeitkalibrierung in Augmented Reality Anwendungen*. PhD thesis, Technische Universität München, 2011.

[SS04] B. Schwald and H. Seibert. Registration Tasks for a Hybrid Tracking System for Medical Augmented Reality. *Journal of WSCG*, 12:411–418, 2004.

[WB97] Greg Welch and Gary Bishop. SCAAT: Incremental Tracking with Incomplete Information. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 333–344. ACM SIGGRAPH, Addison Wesley, August 1997.

[WB04] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical report, 2004.