Application of Time-Delay Estimation to Mixed Reality Multisensor Tracking

Manuel Huber[*], Michael Schlegel[*], Gudrun Klinker[*]

[*]Technische Universität München
Fakultät für Informatik
Boltzmannstraße 3
85748 Garching b. München, Germany
email: {huberma, schlegem, klinker}@in.tum.de

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 spatial tracking sensor fusion has already seen considerable interest, most results only deal with the integration of carefully arranged setups as opposed to dynamic sensor fusion setups.

A crucial prerequisite for correct sensor fusion is the temporal alignment of the tracking data from several sensors. Tracking sensors are typically encountered in Mixed Reality applications, are generally not synchronized. We present a general method to calibrate the temporal offset between different sensors by the Time Delay Estimation method which can be used to perform on-line temporal calibration. By applying Time Delay Estimation on the tracking data, we show that the temporal offset between generic Mixed Reality spatial tracking sensors can be calibrated.

To show the correctness and the feasibility of this approach, we have examined different variations of our method and evaluated various combinations of tracking sensors. We furthermore integrated this time synchronization method into our UBITRACK Mixed Reality tracking framework to provide facilities for calibration and real-time data alignment.

First presented at the 9th Workshop der GI-Fachgruppe VR/AR 2012, extended and revised for JVRB

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 a head-mounted display) require a high level of tracking accuracy in order to render convincing visualizations.

For example an inside-out square marker based tracking system offers sufficient tracking quality for an augmentation as long as a marker is visible in the camera image and the motion is slow enough. As the movement of the camera gets faster, the accuracy of the marker tracker decreases. An inertial orientation sensor on the other hand can handle fast movements, but generally suffers from drift.

In order to improve the overall accuracy, the concept of sensor fusion was early introduced to solve various tracking problems (e.g. [Wel96]). Here 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. Different concepts of competitive, complementary and cooperative fusion can be applied [DW88]. The most commonly used competitive methods to fuse data currently include Kalman-based filters [WB04] and particle filters [DFG01].

1.1 Requirements in Augmented Reality Applications

In order to correctly combine data from two tracking sensors, it is necessary to know the exact temporal relationship between data acquired from the different sources.

As a simple example consider an indirect tracking setup, where a mobile, local tracker (for example a camera mounted on the user's head mounted display) is tracking personal objects (for example interaction devices) in the immediate vicinity of the user. Furthermore, the local tracker itself may be tracked in a larger environment by permanently installed room-level trackers (for example a multi-camera outside-in system). To determine the position and orientation of the user's personal objects in the coordinate frame of the room-level tracker, it is necessary to compute the concatenation of the spatial relationships between the room and the local tracker and between the local tracker and the personal object (i.e., a simple matrix multiplication in this case.)

A concrete instance of this setup, for example, is the tracking of tools inside an occluded volume, like the AR-guided welding scenario inside a car white body, as discussed in [KSK08]. Here, a welding gun needs to be tracked in order to augment welding points on the car body. By introducing a mobile, indirect tracker, as illustrated in figure 1, the tool can be tracked even if the worker operates inside the car body.

For the computation of the indirect tracking to be correct, the concatenation may only be performed on tracking data that refers to the same moment in time. Otherwise any movement of the local tracker, for example, introduces additional tracking errors on the resulting pose. While advanced sensor fusion methods (for example [Wel96]) are able to handle asynchronously arriving data and data arriving at different sample rates, they rely on knowing the temporal relation between the tracking sensors remains.

This either requires (hardware-)synchronized sensors or suitable interpolation of the data (see [Pus06]), and thus knowledge of the exact relative temporal alignment of the sensors. We call sensors, for which this relationship is known, to be temporally calibrated.

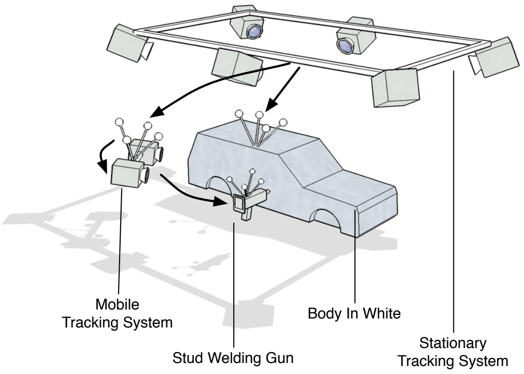


Figure 1: Indirect welding gun tracking setup (from [KSK08]).

Synchronization can be achieved either by hardware means or by software 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. Usually this trigger is derived from one of the sensor oscillators itself.

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 otherwise 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 need 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 spatial tracking sensors S_1 and S_2, which we assume to be rigidly connected.

Considering a motion event happening in the physical world at time t_0, two spatial tracking 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}. In general, these observation times of the same event differ, since every type of sensor requires a different amount of time for the internal signal processing. Further delay will be caused by the various communication stacks of the operating systems as well as network transport of the measurement data. We assume the measurements to arrive at the tracking framework at times t'_{S_1}, t'_{S_2} 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. A schematic of the mentioned points in time can be seen in figure 2.

The accumulated lags can be reduced to one single delay for each sensor $\Delta t_{S_1} = t'_{S_1} - t_0$ and $\Delta t_{S_2} = t'_{S_2} - t_0$. 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 in this context. To align the sensor data for sensor fusion purposes, it is sufficient to determine the temporal offset $\Delta t = t'_{S_1} - t'_{S_2}$.

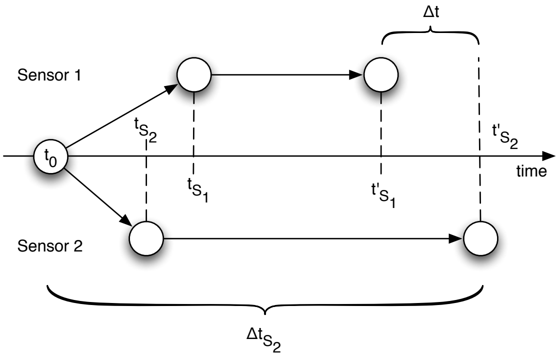


Figure 2: Schematic visualization of different points in time which are relevant for a temporal calibration

Note that especially for optical see-through Augmented Reality and related Mixed Reality modes, the overall latency of the system is also relevant, as ideally visualizations should coincide with real, physical events. While evidently important, this paper deals with the determination of the relative lag between tracking sensors to enable sensor fusion approaches.

In this paper we show how the temporal offset between different spatial tracking sensors, as used in many Virtual and Mixed Reality environments, can be calibrated by applying Time Delay Estimation on the sensor data of the spatial trackers. While the principle of Time Delay Estimation is already well established and used in many contexts, this paper focuses on the necessary steps required to apply this technique in the context of Augmented or Mixed Reality, as well as the resulting performance of the calibration and the improvements in tracking quality obtained by temporal calibration. In this sense this paper builds upon the large body of research on signal processing and delay estimation.

The data from two rigidly connected sensors measuring corresponding spatial relationships are compared by computing a similarity measure which determines the level of agreement between the two sensors. An example of this situation is given in figure 3. A time shift is then applied to one of the data series and the similarity measure recomputed, interpolating between measurements as needed. By applying various time shifts from a certain window, the best match can be determined which then is used as the calibrated value. We use normalized cross-correlation as the similarity measure.

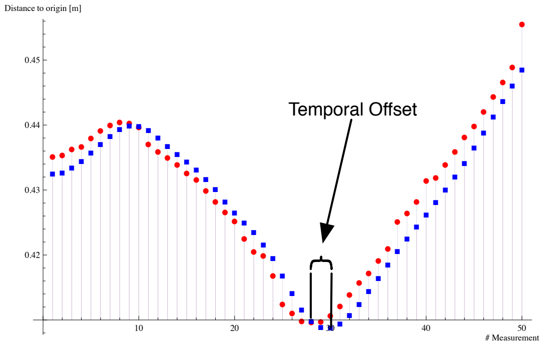


Figure 3: Data of two different senors; the effect of the temporal shift is discernible.

The presented method is applicable either as an off-line calibration step or as an on-line recalibration method which periodically adjusts the sensor calibration due to temporal drifts of one or both sensors. Such recalibration is especially relevant in distributed sensor fusion setups which involve computer communication or in long term applications. Over longer periods of time, the offset between two sensors may change due to temporal drift caused by crystal instabilities (for example due to thermal clock drift; See [PPH$^+$12] for an evaluation.) This also enables calibration of ad-hoc dynamic sensor fusion setups in highly flexible ubiquitous Augmented Reality tracking scenarios, where no a priori knowledge about the involved sensors is available. Discussions of this calibration method can also be found in [Sch11] and [Hub11].

Being able to precisely align arbitrary tracking sensors is an important benefit for Augmented Reality applications.

First, as already stated, the knowledge of the temporal relation of tracking sensors is imperative for correct fusion of the tracking data. By removing this hurdle to sensor fusion in generic Augmented Reality setups, more scenarios become available both for research and for actual deployment. This enables many advanced Mixed Reality tracking modalities, which otherwise would be limited to specific laboratory setups. Bringing greater tracking power at managed complexity to application designers helps to make new Augmented Reality interactions and applications more accessible. This fits well into the overall vision of the UBITRACK tracking framework (see also [Kei11]).

Second, being able to dynamically determine the temporal calibration of generic tracking sensors is essential for dynamic tracking sensor fusion ([PHBK06]), where the Augmented Reality system dynamically reacts to changes to the tracking setup and reconfigures the processing of the tracking data at runtime. Dynamic sensor fusion is itself a necessary requirement for the vision of Ubiquitous Augmented Reality, where AR interactions become prevalent in a Ubiquitous Computing sense (see [Hub11], [MSW$^+$03], [NIH01]).

A third area where synchronization of sensors is important, is the error evaluation of tracking setups. Here the data measured by a tracking system under test is compared to ground-truth data (or reference 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 where this kind of calibration was used is described in [GGV$^+$10].

2 Related Work

Several publications deal with the topic of sensor fusion. Examples of successful applications of sensor fusion for different applications include outdoor Augmented Reality [PT01, RD06] which describes the fusion of separate location and orientation sensors to derive a full pose.

The negative influence of lag on the usability of AR applications is generally agreed upon (see for example [AB94, Hol97, Wel96]). Nevertheless, temporal calibration considerations for Augmented or Mixed Reality setups are so far mostly limited to particular, specialized setups. Mostly components are either hardware synchronized or the lag between different sensors is tuned in software by experimental means. Drawbacks of these approaches are that common off-the-shelf hardware often lacks suitable hardware synchronization interfaces.

The introduction of the concept 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. Also the dynamic fusion of inertial tracking devices was successfully integrated [PK08].

One of the major remaining questions in this setup is how to dynamically account for sensor synchronization. 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. Also in [NBP$^+$07] and [HPK$^+$07] a general centrally coordinated peer-to-peer tracking architecture is proposed. With the further application of distributed computing to sensor fusion, the problem of clock synchronization is further emphasized. In these scenarios a method to temporally calibrate distributed sensors is of great importance.

In [ASB07] and [ASB04] 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 [BS08] the employed camera and inertial tracker are synchronized via a common clock source that triggers both sensors. Such a setup using hardware synchronization currently seems to be the most common case, but in general is prohibitive in ubiquitous tracking scenarios.

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 residual. The temporal offset which minimizes the geometrical residual is then taken as the temporal offset (see section 5.2 for a comparison with our approach). Also a related approach, which jointly estimates the position and the relative lag between sensors using a sequential Monte-Carlo approach was presented in [VMAR07].

In [GFG11] the authors presented a method for tracking in a wireless sensor network with unsynchronized sensors. Their approach is based on particle filters and aims to offer a favorable trade-off between tracking accuracy and low computational burden.

In [JLS97] an in-depth study of latencies occurring in an Augmented Reality setup 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 Augmented Reality system was built and used for manually adjustable prediction to determine the temporal offset between the sensors by visual means.

In [JK05] the effect on the resulting sensor fusion uncertainty of a Kalman filter of imprecisely known relative time delays between sensors and of uncertain sampling instants was investigated.

3 General Calibration 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 tracking 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, nor do they occur at the same frequency. It isn't even guaranteed that they occur at constant frequency. The only assumption is that they have reliable time stamps, even though taken with a, yet to be calibrated, temporal offset.

For each pair of signals a suitable similarity measure $\rho_{X,Y}$ can be computed that measures the mutual agreement of the signals. Since the measurements of different tracking sensors refer to the same physical observation, the signals show inherent similarities (even though they may have been taken from different vantage points.)

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 sensor is used since both measurements will be individually affected by noise and other kinds of tracking 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 Δt between the sensors. This is a well-known approach in signal processing [Car81] and can be written as

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

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

3.1 Various properties of Time Series

Time series of tracking data, as encountered in Mixed Reality spatial tracking settings, exhibit specific properties. These have to be considered before Time Delay Estimation can be applied.

Dimensionality Different spatial sensors can sense different kinds of physical activity. Accordingly, the degrees of freedom (DoF) of the measurements of the involved sensors and accordingly the number of dimensions required to represent these measurements vary. The most common types of tracking sensors measure 3D position, 3D orientation or both simultaneously (6 DoF). But also less common combinations exist, such as 3DoF position and orientation poses on a 2D surface [WHK$^+$10]. The calibration procedure has to take these different types of measurements and their respective representations into consideration.

Registration In order for the signals of two different tracking sensors in a Mixed Reality application to be comparable we assume that the spatial relationship between the two sensors is static and known and that the sensor measurements have been transformed into a common coordinate frame. This spatial relationship can usually be determined by spatial calibration. Note that since the accuracy of this spatial calibration also depends on the temporal alignment of the sensors, the spatial and temporal calibration should be used in an iterative or adaptive process. Also, since the temporal calibration process is assumed to be rather robust against spatial registration errors, the requirements on the accuracy of the spatial registration of the tracking sensors is rather low.

Signal-to-noise ratio The Signal-to-noise ratio (SNR) of a signal is defined as the ratio between the power of a signal and the power of the measurement noise, respectively the square of the individual amplitudes.

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2,$$

where P and A are power and amplitude of the respective components.

For tracking sensors this characteristic can be interpreted as 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. Note that in the remainder of the paper, we will not make quantitative statements about the SNR of specific spatial tracking signals. The SNR will only be used to distinguish between "high SNR" signals, which feature motions of large-scale motions, from signals with "low SNR" where the extents of the intended movement is within an order of magnitude of the spatial tracking sensor noise.

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. Also while the sample points are usually assumed to be distributed equidistantly in time, in practice this assumption does not always hold true. Also the implications of this are further discussed in section 4.3.

4 Calibration Procedure Details

As mentioned above, 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. To estimate the temporal offset between the tracking signals, we aim to compute a single, real-valued similarity value that characterizes the agreement of the signals for a specific time shift. By first reducing the multidimensional tracking data to a one dimensional real-valued time series, we can directly apply various, common similarity measures. While reducing the physical representational precision of the sensor signal, the time calibration primarily utilizes the shape of the motion. Similar to the situation in machine learning [Fod02], 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 (Segmentation, Dimensionality reduction, Interpolation, Time Delay Estimation and Aggregation) of our proposed calibration method in more detail.

4.1 Segmentation

The sensor data is assumed to be an endless stream of measurements. Thus, as a first step, this data is divided into small chunks of a fixed length (duration in time). This serves two specific purposes. Shorter chunks of data are easier to process, both in terms of speed and complexity.

Second any on-line estimation requires some sort of segmentation in order to produce results during the runtime of the procedure. Although it is possible to aggregate the calibration results of several chunks, global optimization strategies working on the complete history of the tracking data in general are not suitable for on-line synchronization.

The segmentation is performed by accumulating incoming samples in a separate buffer for each sensor. As soon as the required amount of sensor data has been acquired, the buffer is copied and processed by the calibration method. Note that if the sampling rates of the sensors differ, this condition will depend on the amount of data produced by the slower sensor.

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 if the sample rates of the sensors differ. In the former case the buffers are completely emptied between segments, whereas in the latter case the buffers are only partially cleared and shifted.

4.2 Dimensionality reduction

As previously discussed, the optimization is not performed on high dimensional tracking data, but rather on one dimensional signals. As mentioned, the critical aspect of this reduction is not to maintain the immediate relationship to any specific geometric interpretation. Rather the similarity of corresponding events as registered by different sensors should be immediately evident from the signal.

We will limit the discussion of the projection methods to the spatial tracking sensor S_1 and assume that the same projection is used for the data of the tracking sensor S_2. In this context, we search for a mapping $f : \mathbb{R}^n \mapsto \mathbb{R}$ that reduces the high dimensional spatial measurement data x_t from S_1 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 of the spatial tracking data 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 a reduced SNR of the projected signal if the movement of the tracking 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 of the frame of reference for 3D position measurements.

This method also works for incremental rotation measurements (as used for gyroscope integration, see also [PK08]). A static projection can be used if the measurements are represented as incremental rotation measurements. If the measurements are represented as a 3-element rotation velocity, the reduction can be obtained by computing the norm.

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 directions of the most significant movements in each segment, which can then be used as projection vectors for the particular segments.

It is also possible to further smooth the projection behavior of this method, by calculating the projection vector as a moving average over a limited history of previous segments. This further increases the robustness against outliers in the spatial tracking data while reducing the speed of adaption.

Reduction incorporating feedback Another improvement of the dimensionality reduction is to consider the final significance of the Time Delay Estimation as determined using a particular projection. This is a kind of feedback situation, where the choice of the dimensionality reduction projection is optimized in order to maximize the significance of the final result. It is easy to see that this increases the computational cost of the reduction.

Another similar method, which represents a tradeoff between the generality of this approach and the required computational cost is to consider two different dimensionality reductions in parallel. The Time Delay Estimation is then performed on two different pairs of signals and the final decision between the two projections is delayed until the significance of both is known.

Pathological sensor data While there are many variations possible, experience shows that in practice the time offset calibration of spatial tracking data, as typically encountered in the context of Mixed Reality scenarios, is usually robust against the choice of dimensionality reduction for most use cases. We will shortly discuss the various possible pathological cases, which may be encountered in the context of spatial tracking, that render the dimensionality reduction of tracking data ineffective and thus the temporal calibration of spatial sensors useless.

The simplest pathological case is zero movement, thus no events happening in reality. In this case the signal of the two tracking sensors consist solely of the sensor-noise. Since the sensors are assumed to be operating independently, the resulting signals are consequently uncorrelated. 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

User	Relative lag	Std. deviation
1	32.4 ms	2.7 ms
2	32.1 ms	0.33 ms

Table 1: Results of calibration using physiological tremor

for sensor movements confined to a plane orthogonal to that axis.

Yet, we argue that, in the context of spatial tracking for Mixed Reality applications, 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 by a human operator.

Due to the physiological (normal) tremor of the hands of the participants, the signal-to-noise ratio of these measurements was already sufficient 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. This can be seen in table 1. 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 exist, due to the nature of the human interaction in Mixed Reality scenarios, these cases are scarcely encountered in practice. Also the use of adaptive projections or the power of choice between different 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. This experiment also raises the question on the general influences of human factors on the various parameters of sensor data and the resulting quality of the temporal calibration.

4.3 Interpolation

A common assumption in Mixed Reality tracking frameworks, 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. In Figure 4 the sample rate of the Faro CMM is depicted and the variance of the sample rate is discernible.

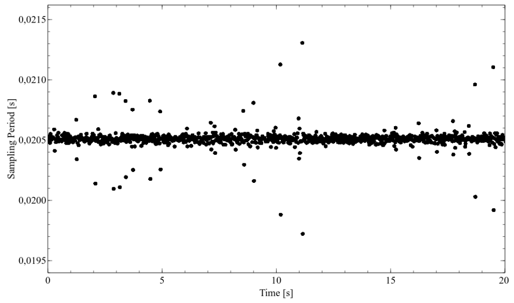


Figure 4: Sample frequency of the Faro CMM ($\mu = 48.8\,\text{Hz}$, $\sigma = 0.53\,\text{Hz}$)

Furthermore the relationship between the sample points of the two signals is generally not known. Thus one sample point of the first tracker does not directly correspond to one sample point of the second signal. Also, while the individual fluctuations in the update rates can usually simply be handled by assigning timestamps, these fluctuations complicate the computation of similarity measures and thus the comparability of the signals in general. 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. This results in direct one-to-one correspondences of samples in both signals by sampling from both signals at common timestamps. This is the foundation for the similarity computation.

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, which makes resampling for this sensor unnecessary. If the sampling rates of the sensors differ, it is beneficial to interpolate the signal with the higher sampling rate to minimize interpolation errors.

Also note that in our application, it is preferable to interpolate the one dimensional projected signal as opposed to the higher dimensional tracking data. First, the interpolation at this stage is more straightforward and more clearly defined. This may be harder for general tracking data (for example different interpretations of quaternion interpolation). Second, the computational complexity is also less due to the reduced number of dimensions.

This approach also implicitly handles tracking sensors with different update rates. As the tracking signal is interpolated between the actual sampling points of the individual sensors, the representation of the perceived motion by the trackers becomes independent of the concrete sampling rate of the devices and in fact of the actual sampling instances. The limiting factor on the acceptable disparity of the sensor update rates in the context of spatial Augmented Reality tracking is that the motions which are used for calibration need to stay below the temporal resolution of the tracking sensor with the slowest update rate. Otherwise significant motion events may only be observable in one tracking signal, which renders this particular motion ineffective for temporal calibration. Usually this is no concern in Augmented Reality tracking applications.

4.4 Time Delay Estimation (TDE)

After these preprocessing steps we have two one-dimensional signals $\widehat{X} = \{\widehat{x_t} = f(x_t) : t \in T'\}$ and $\widehat{Y} = \{\widehat{y_t} = f(y_t) : t \in T'\}$ 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 class of methods known as Time Delay Estimation (TDE). These methods are well known and understood in signal processing and are used for applications such as RADAR or SONAR (see for example [Car81]). Generally, the aim of Time Delay Estimation is to estimate the temporal offset of a specific pattern contained in a usually noisy signal. In many cases, this pattern is first transmitted by the system and later received as a reflection (e.g. SONAR/RADAR).

Our application differs, since we do not actively send out one instance of the pattern we are later looking for. We Rather treat the signal of one spatial tracking sensor as the search pattern we are looking for in the other tracking signal. The offset between these two pattern instances is the relative lag between the sensors.

As already mentioned to compute the Time Delay Estimation, we keep one tracking signal fixed and shift 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. Finally the timeshift which maximizes this measure expresses the time delay estimate.

Cross-Correlation One of the earliest and still most important similarity measures used for such setups is the normalized cross-correlation. The computation of the cross-correlation can be optimized by methods presented in [JS93] and others.

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

$$\rho_{X,Y} := \frac{Cov(x,y)}{\sigma_x \sigma_y}.$$

Calculating the similarity between the signals while shifting one in time results in a graph as exemplified in figure 5.

Note that there are also correlation approaches for vector valued time series (such as the canonical correlation; see for example [Joh97]). While this would eliminate the need to reduce the dimensionality of the tracking data, our initial evaluations showed that less expressive results at increased computational cost are obtained as compared to the approach using dimensionality reduction.

Resolution of the TDE For a grid-search approach as outlined above, 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. Furthermore the computation time obviously increases with decreasing step size, making very small steps unfeasible.

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]. Both the maximum correlation value and the rate of increase of the fitted parabola can be used as indicators on the significance of the time delay estimate on the individual segment.

Empirical studies showed that a temporal resolution of 1 ms is both feasible and useful, especially considering the update rates of commonly encountered spatial tracking sensors in the domain of Mixed Reality applications. These update rates typically range from 10 Hz up to 1 kHz. For more exact, but also more time consuming, computations step width up to 0.01 ms or an iterative refinement procedure can be used for the tracking data.

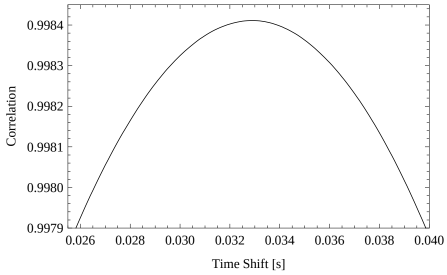


Figure 5: Graph of similarity (correlation) vs. timeshift

4.5 Aggregation

After the Time Delay Estimation has determined the relative lag of the two spatial trackers on one segment, multiple segments can be aggregated to identify meaningless results, reject outliers or perform smoothing of the lag calibration. Since the aim is to be able to perform calibration and correction at run-time, also the combination of segments needs to be performed adaptively.

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.

5 Evaluation

To validate the method described above we conducted a series of experiments involving different pairwise combinations of sensors. Figure 6 shows an exemplary path along which a sensor-pair was moved. We first describe the general hardware setup used as well as the individual experiments undertaken. We then demonstrate the effectiveness of this approach by an evaluation of registration errors in both unsynchronized and synchronized sensor fusion. All the evaluations were performed using the UBITRACK system.

The UBITRACK Mixed Reality tracking framework provides a wide range of native device drivers for various spatial tracking sensors, such as vision based, inertial or mechanical trackers. Multiple sensors may be connected into a single system sharing a common basis for timestamps, which are rigorously attached as soon as any measurement enters the system.

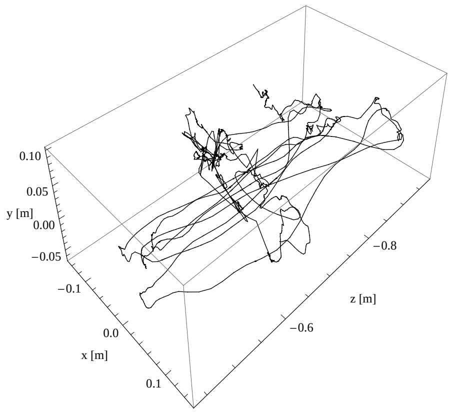


Figure 6: Sample movement used for calibration of the relative latency. (All axes in meters.)

5.1 Pairwise evaluation setup

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

- The *A.R.T. system*[1] 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 7(a).
- The *Faro Fusion*[2] 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 7(b).
- An inside-out *optical square marker tracker* (integrated into the UBITRACK framework; similar to [KB99]) 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 7(c).

For all of the following experiments, the following parameters were selected:

- Segmentation: Simple disjoint segments of 2 s length.
- Interpolation: Linear interpolation.
- Dimensionality reduction: Computation of the norm of the 3D-position vector.

[1] `http://www.ar-tracking.com`

[2] `http://www.faro.com`

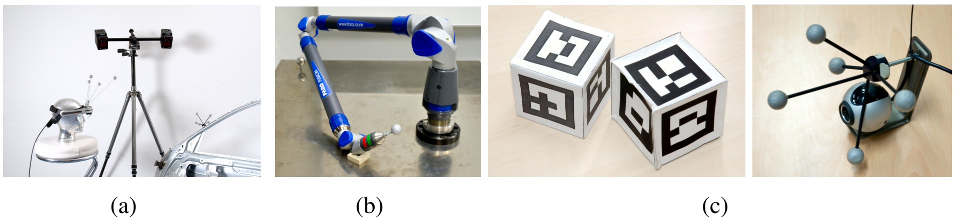


Figure 7: Sensors used for evaluation; (a) A.R.T. infrared tracker; (b) Faro CMM arm; (c) Square markers with webcam

- Time Delay Estimation: Normalized cross correlation with grid-search approach.
- Aggregation: Mean of the individual offsets of the segments.

For all experiments traces of 20 second length were recorded for each tracking sensor, yielding 10 segments per tracking sensor. Also each evaluation was performed two times to ensure the validity of the obtained results.

Setup 1 – A.R.T. vs. Square-Marker In the first setup we compute the relative lag between the A.R.T. outside-in tracking system and the inside-out square marker tracker. For this an A.R.T. marker body was mounted on top of a USB camera and a square marker was fixed relative to the A.R.T. cameras. The spatial relationship graph (SRG) describing this setup can be seen in figure 8.

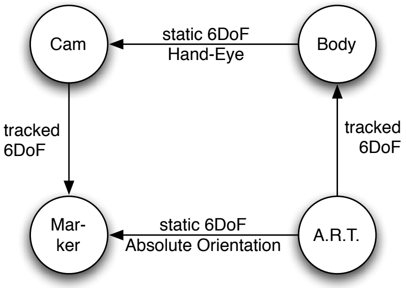


Figure 8: SRG describing the spatial relations of A.R.T. and square-marker tracker

The poses describing the relationship between the square marker and the A.R.T. cameras as well as the relationship between the USB camera and the mounted A.R.T. body, are static for the duration of the experiment and have both been registered in advance. Note that this setup combines an inside-out and an outside-in tracker and thus the actual sensors themselves are not rigidly connected. Nevertheless the inside-out sensor is rigidly connected to a tracked body from the outside-in sensor, which suffices to be able to transform the individually sensed motion into a common frame of reference. The same argument holds for all following setups, which combine inside-out and outside-in sensors.

While the pose of the square marker in the A.R.T. system was determined by solving the absolute orientation problem [Hor87], the relationship between the USB camera and the A.R.T. body was obtained by Hand-Eye-Calibration (e.g. [Dan99]). Using these registrations it is possible to transform the 6DoF poses of the A.R.T. system into the USB camera system. Figure 9 shows an example of the evaluation of one data set. Here the different correlation coefficients for two different 2s chunks of a 20s recording are plotted as well as a majority function (pointwise multiplication of the individual coefficients) over 10 chunks is shown for illustration.

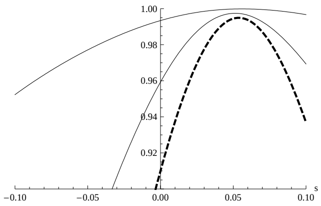


Figure 9: Correlation of 2 chunks and majority function (dashed)

Setup 2 – A.R.T. vs. Faro CMM In the second setup we synchronize the A.R.T. outside-in tracking system with the Faro CMM. The setup consisted of the Faro CMM inside the tracking range of the A.R.T. system, where a single A.R.T. marker ball was mounted on the tip of the Faro arm. The A.R.T. system in this scenario is only used for tracking the 3DoF position of the marker ball (as can be seen in figure 7(b)). The SRG describing this setup is shown in figure 10.

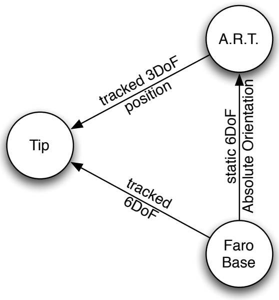


Figure 10: SRG describing the spatial relations of A.R.T. and Faro CMM

The tip of the Faro arm was registered to the center of the A.R.T. marker ball which is also the point tracked by the A.R.T. system. Thus this relationship does not show up as a static edge in the SRG. The relationship between the Faro base and the A.R.T. cameras had to be determined in order to transform the A.R.T. data into the Faro system. This calibration was again done by solving the corresponding absolute orientation problem.

In this setup one of the trackers only delivers 3DoF position information and thus only the position can be used for aligning the sensor data.

Setup 3 – Faro CMM vs. Square-Marker This setup is similar to the combination of the A.R.T. system with the square marker tracker. In this case the USB camera for marker tracking was mounted rigidly to the Faro head and a square marker was fixed relative to the base of the Faro base. The spatial relationship graph (SRG) describing this setup is omitted since it is similar to the SRG as seen in figure 8, with the Nodes “A.R.T.” and “Body” replaced by “Faro Base” and “Tip” respectively. Also the calibration procedure for this setup was similar, featuring an absolute orientation and a Hand-Eye-Calibration.

Results The results of the temporal calibrations are summarized in Table 2. The relatively large standard deviations in setups 1 and 3 stem mostly from pronounced temporal instabilities of the square marker tracker (see also section 7.3). This is especially evident when compared to the very precise results obtained for setup 2.

Transitivity Another test for the validity of the temporal calibration method is to examine the consistency of setups involving more than two spatial tracking sensors. When combining more than one pair of sensors the relative lag between the individual pairs has to obey transitivity. Ideally, the sum of relative lags respecting the temporal direction along a loop should be zero or a small residual error.

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 2: Results of temporal calibrations

This is visualized in figure 11 for the combination of the setups as described above. As can be seen the directed arrows of the temporal offset indeed form a loop and the values agree, considering the respective standard deviations. For a loop starting at the ART tracker, we get

$$\varepsilon = \Delta t_{ART,Faro} + \Delta t_{Faro,Marker} + \Delta t_{Marker,ART} \approx -7\,\text{ms},$$

which is well within the accuracy of the calibrations involving the square marker tracker.

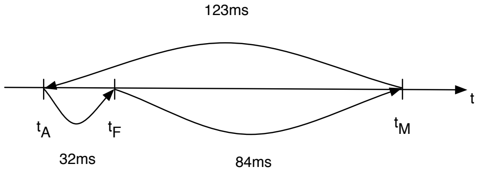


Figure 11: Transitivity of pairwise sensor offsets

5.2 Comparison with geometric error minimization

To further validate the correctness of our approach we compared the resulting relative lag of two tracking sensors with the time offset which minimizes the overall geometric error between the sensors. All other parameter of the calibration procedure were selected identical as in the pairwise evaluation above 5.1. This is similar to the static calibration approach used in [LBMN09, SS04]. As can be seen in table 3(a) both methods yield comparable results. This is expected since the correct relative lag also reduces the error between the registered coordinate frames of the sensors. The experiment in part (a) of 3 was performed using sensor data with significant SNR.

Method	Mean	Std. Dev.
(a) High SNR experiment		
Geometrical	32 ms	0.1 ms
Correlation	32 ms	0.1 ms
(b) Low SNR experiment		
Geometrical	35 ms	39 ms
Correlation	32 ms	0.3 ms

Table 3: Comparison with geometric error minimization

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 4.2). Table 3(b) shows the results. 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 relative lag estimation performs only slightly worse than before.

5.3 Error reduction

To illustrate the effectiveness of the temporal alignment for spatial tracking data as encountered in the context of Mixed Reality applications, 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 (Setup 2) also using the same calibration procedure parameters.

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 12(a) shows the error vector between measurements from the A.R.T. system and corresponding points for identical timestamps as measured by the Faro system during the movement. The root mean square (RMS) spatial error between the trackers 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.

The temporal offset in this experiment was, as before, determined to be 32 ms with 0.1 ms standard deviation calculated over all segments. Figure 12(b) shows the same error vectors (here magnified by a factor 10) 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 error has been reduced to 1.6 mm. The remaining errors mostly stem from spatial calibration errors and sensor noise.

6 Integration

The method for temporal calibration and alignment presented thus far was integrated into the UBITRACK tracking framework. This allows for setup of complex and dynamic sensor fusion scenarios which benefit from formal and automated reasoning about tracking environments.

The sensor calibration is integrated as a separate calibration method pattern, similar to Hand-Eye-Calibration or absolute orientation. This pattern is usually instantiated from tracking management tools such as trackman [Kei11] for calibration of setups prior to actual user interaction.

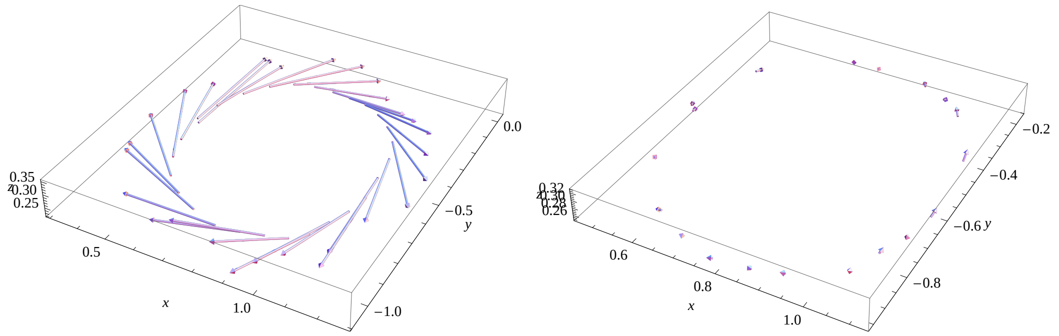
In an actual application of sensor fusion, the data from each sensor can be synchronized by a timestamp correcting component. The component is configured by calibration data obtained in the previous step and modifies the data stream accordingly. This allows for online correction for lag between different sensors.

While this method produces very accurate results it increases the overall latency of the system [JLS97]. Instead of holding the data it is also possible to perform prediction as proposed in [AB94, JLS97].

7 Monte-Carlo evaluation

As illustrated by the explanations in the previous sections, the design space for time calibration procedures is large. Due to the large number of possible approaches to each of the individual steps, the combinatorial number of possible implementations grows quickly beyond what can be reasonably evaluated in an integrated approach.

To facilitate careful and isolated evaluation of the various components we have also developed a suitable statistical simulation framework. Influences of various tracking data characteristics, for example the signal to noise ratio, can be investigated. Furthermore the impact of different approaches for the various components on the overall calibration quality can be directly



(a)

(b)

Figure 12: Error vector between measurements from A.R.T. and the Faro system during movement; (a) without temporal alignment; (b) with temporal alignment; (all axes in meters; magnification factor 10 for figure (b))

compared.

This section will give an overview of the framework and demonstrate its usefulness on an evaluation of different dimensionality reduction schemes.

In the area of Time Delay Estimation, simulations have already been used to evaluate the performance and robustness of methods. For example Fertner and Sjölund [FS86] have evaluated five different correlation functions by superimposing a sensor signal with two independent noise signals and subsequently shifting one by a fixed amount in time.

Motivation There are two main applications of the statistical simulation in this context.

An immediate application is the analysis of the calibration quality (relative lag in this case) as a function of the input parameters. The input parameters can model various adverse effects on the sensor signals which can interfere with the computations. Thus the robustness and performance of the calibration method can be evaluated when subjected to various disturbances.

Possible input parameters may include sensor noise, sensor update rates and jitter as well as registration errors or the relative lag itself.

A second application is to compare the relative performance of the different temporal calibration procedures, when substituting individual components. By keeping the input parameters fixed across various experiments, the relative performance of different implementations of the various stages can be compared. Thus individual subsystems of the procedure can be evaluated individually and the impact of changes on the overall performance can be assessed. This enables "one-factor-at-a-time" evaluation of the temporal calibration method.

7.1 Monte-Carlo model

The basis of the statistical framework is a Monte-Carlo model for simulation of the time calibration process. This model has seen many uses in randomized tests and statistical modeling, both in sensor technology and other areas. In [CHS04] and [CH06] an adaption of Monte-Carlo simulation to general metrological problems is given, and our implementation is based on this description. It is also similar to the methods employed by [FS86] and [ZA05].

Basic approach Monte-Carlo simulation is basically a technique for propagating the probability density function of input quantities X through a (complex) model $f(x)$. The model can be arbitrarily complex, since the propagation process is performed numerically rather than analytically. The result of each simulation are samples y from the dependent probability distribution of the output quantity Y. From these samples various statistical information can be derived and the behavior of the model under the input parameters can be inferred.

Simulation overview The general simulation process is divided into three phases [CH06]:

1. In a setup phase a suitable statistical model and the related probability distributions of the input parameters are determined. This is usually performed by obtaining an appropriate number of random samples from a real source and estimating their associated statistical properties. Also the number of experiments in the simulation is fixed.

2. The computation step, the simulation is performed for each random sample of the setup

phase. The result is thus a vector of random samples of the output distribution.

3. In the post-processing phase the simulation results are gathered and a statistical analysis is performed.

Figure 13 shows an overview of the Monte-Carlo process.

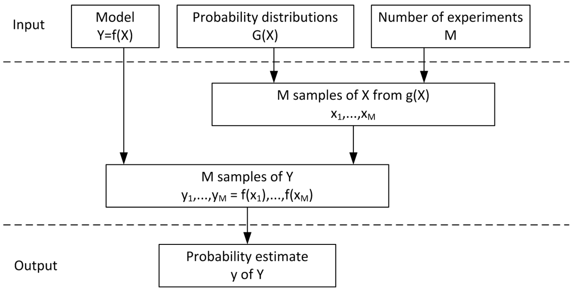


Figure 13: Monte-Carlo simulation process (simplified from [CHS04])

Setup phase In the setup phase the input parameters for the simulation are determined. This includes defining the computational model f, the input random variables X and determining the associated random distributions of the variables. As mentioned above there are various possibilities for defining relevant input parameters such as sensor noise or misregistrations.

Due to its numerical nature, a direct advantage of the Monte-Carlo method is that the computation model can be as complex as desired, as long as it is computable. In our case the computational model is the same computation as the already described temporal calibration process using suitable sensor signals which were perturbed according to the input random variable. This also enables comparison of the performance of various parts of the calibration process, by accordingly modifying the computation. Thus each simulation takes a randomly perturbed sensor signal as input and derives a corresponding relative lag as output, which is the output random variable in this context.

Also the number of sample values from distribution and thus the number of simulations is fixed. Depending on the overall “shape” of the probability distributions of the input parameters, this influences the uncertainty of the resulting estimate. To achieve reasonable results, as a general rule sample sizes in the range 10^5 to 10^6 should generally be used, although this might prove unfeasible depending on the complexity of the computation model. Also an adaptive approach can determine the number of required samples during the simulation. For more details see [CHS04] and [CH06].

Simulation and post processing phase The actual simulation consists of drawing the predetermined number of samples from the input distribution and computing the dependent model values.

The individual results of the simulations can be used to infer statistical properties of the output random variable Y. The most widely used characteristics are the mean and the variance. These can be approximated from these sampling using suitable estimators, such as the sample mean $\mu \approx \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and the unbiased sample variance $\sigma^2 \approx s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$.

7.2 Application to temporal calibration for Mixed Reality spatial tracking

To adapt the general Monte-Carlo simulation process to the simulation of relative lag estimation for spatial tracking in Mixed Reality setups, several peculiarities have to be noted. The most important aspect is the generation of the concrete input sensor signals for the calibration process. Also the careful selection of the input parameters and their distributions is important for attaining meaningful simulation results.

Input signal generation The temporal calibration as described above always operates on two time series of spatial sensor data and tries to determine the relative lag between these two. To generate suitable input data which reflects the random input parameter samples as drawn in the simulation phase, the following procedure was implemented.

First a library of reference data was produced. Different motions of actual AR users were captured and processed to reduce the influences of the concrete sensor. This approach has the benefit of keeping the simulation input close to actual AR use cases. Figure 14 shows the procedure to clean up the captured sensor data as reference data. The data is first filtered to eliminate outliers and to reduce the amount of noise in the signal. In order to simulate different sampling rates, the captured data is also smoothly interpolated, thus resulting in a smooth, continuous signal with no outliers.

To achieve representative results, it is further necessary that the library is large enough and contains a diverse selection of motion captures. The motions have to cover a large spectrum of diverse scenarios

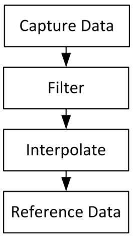


Figure 14: Reference Data generation

and should include both slow and fast examples. Also typical “calibration movements” exercising large, sudden movements in many dimensions, can be useful to determine the baseline performance of various procedures.

To generate an input instance from the reference data and a concrete sampling of the input parameter distribution, the reference data is duplicated and the copy is subjected to specific perturbations according to the input parameter. This could for example describe a certain amount of noise or misregistration added to the signal copy. The copied signal is furthermore shifted in time by a fixed and known amount. This is the a priori known relative lag Δt that the calibration procedure will try to recover from the input signals. Figure 15 shows the process to derive both input signals from a chosen reference and the random input parameter sample.

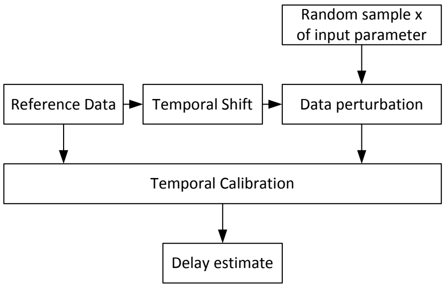


Figure 15: Simulation input generation

The relative lag estimate $\widehat{\Delta t}$ is compared to the known true value and the error is computed. The final output parameter of the simulation is thus the difference between the estimated relative lag and the known true value $\varepsilon_{\Delta t} = |\Delta t - \widehat{\Delta t}|$, rather than the temporal calibration value. This facilitates comparisons between different setups.

7.3 Simulation parameters

In the context of temporal calibration, there are various influences on the sensor signals that can determine the overall performance of the procedure. The characteristics of the these input parameters, in our case, are determined experimentally.

Noise One of the most important and prevalent influences on sensor data is the sensor noise. The most common models for noise is Gaussian noise, which can be represented by an n-dimensional additive random variable with Gaussian probability distribution $X \sim N_n(\mu, \Sigma)$ with mean μ and covariance matrix Σ. Furthermore the mean can be assumed to be 0, since any non-zero offset would model a systematic error rather than noise. Thus to accurately describe the sensor noise input parameter in the context of Monte-Carlo simulation, the associated covariance matrix has to be determined.

To experimentally estimate the noise characteristics of a tracker, it has to be fixated and the sensor data should be sampled for a reasonably long period. The captured data has to be corrected for any systematic offset, effectively moving the barycenter to the origin of the reference frame. Special care should be taken to remove sensor drift from spatial sensors susceptible to such behavior (e.g. inertial sensors).

We determined the noise behavior of the previously described A.R.T. and the Faro CMM systems. For this experiment, the Faro Tip with a mounted A.R.T. marker was rigidly fixed in a vice on a tripod. This setup can be seen in figure 16.

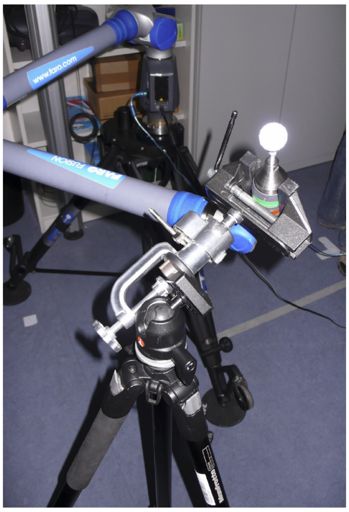


Figure 16: Fixed Faro tip with A.R.T. marker ball

From a 42 s long capture, the following covariance

	Mean	Std. Dev.
A.R.T.	16.7 ms	0.0804 ms
Faro CMM	20.5 ms	0.0596 ms
Sq. Marker	70.9 ms	17.5 ms

Table 4: Mean sample periods and standard deviations

matrices for the two systems were constructed.

$$\Sigma_{Faro} = \begin{pmatrix} 7.9\times10^{-11} & -6.0\times10^{-11} & 1.4\times10^{-11} \\ -6.0\times10^{-11} & 3.0\times10^{-10} & -6.5\times10^{-11} \\ 1.4\times10^{-11} & -6.5\times10^{-11} & 2.4\times10^{-10} \end{pmatrix}$$

$$\Sigma_{ART} = \begin{pmatrix} 6.3\times10^{-10} & -3.0\times10^{-10} & -2.8\times10^{-10} \\ -3.0\times10^{-10} & 2.0\times10^{-9} & -2.8\times10^{-9} \\ -2.8\times10^{-10} & -2.8\times10^{-9} & 6.1\times10^{-9} \end{pmatrix}$$

For illustration purposes, the corresponding RMS errors for the two covariance matrices were computed ([Sch06]) as $\mathrm{RMS}_{\mathrm{Faro}} = 0.025\,\mathrm{mm}$ and $\mathrm{RMS}_{\mathrm{ART}} = 0.094\,\mathrm{mm}$. These values are well within the expected range of these devices.

The noise behavior of the ART system was also the topic of previous experiments ([BSP$^+$06], [Sch06], [Kei11]) and similar covariance matrices can be found there.

Clock jitter and drift Of utmost importance for temporal calibration of spatial tracking sensors in the context of Mixed Reality applications is the temporal behavior and the temporal stability of the tracking sensors themselves. To determine these characteristics, 40 s long data segments were captured with each sensor. The difference between two consecutive timestamps of the received data can be computed and the resulting update rates are analyzed with regard to mean value and stability.

Figure 17 shows the behavior of the three previously discussed trackers. The results are summarized in table 4. Of special interest here is the rather irregular behavior of the square marker tracker, which in part can explain the large deviations in table 2 for results involving this sensor.

Spatial misregistration We previously assumed that the spatial tracking sensors are spatially registered to measure the movement in the same frame of reference. We furthermore stated that the temporal calibration is rather robust against moderate spatial misregistrations, as long as the overall "shape" of the perceived motion stays similar. Also approaches such as the canonical correlation (for example [Joh97]) can produce similarity measures that are invariant against affine transformations of the input signals.

To evaluate the actual impact of misregistrations and the performance of such approaches, the spatial divergence of the signals can be used as an input parameter to the simulation. Various models for registration errors are possible in such a setup, including simple offsets, spatial transformations or distortions.

An application of the general Monte-Carlo method to estimate the propagation of such calibration uncertainties in typical AR applications were also investigated by [Kei11].

Relative lag The relative lag between the two input signals can be used as a simulation parameter itself. By varying the offset by which the signal copy is shifted, the procedure can be evaluated against different ranges of temporal asynchronicity. A robust procedure is expected to operate independently of the concrete temporal offset, as long as it is in its search range.

7.4 Evaluation of dimensionality reduction

As a comprehensive example of the statistical evaluation, we compared a norm based dimensionality reduction method with a PCA based one.

As described earlier, the norm based dimensionality reduction maps each position vector of the sensor measurement to the distance from the origin in its coordinate frame. The second dimensionality reduction first computes the principal axes of each data segment and projects the vectors along the longest axis. The other parameters of the calibration procedure were chosen as in 5.1. The motivation for this approach is to maximize the significance of the projected signal.

Input setup The evaluation was performed using two different sets of reference data A and B of 4 s length. They are visualized in figure 18.

The first set A was derived from typical calibration movement featuring fast and large movements involving many changes of direction. The second set B was obtained from the experiment measuring the physiological hand tremor (see section 4.2). It also features many changes of direction, but the overall extent of the movement is rather small (millimeter range).

For this simulation, noise superimposed on the copy of the reference signal was chosen to be the input parameter. For each reference set the copy was shifted by

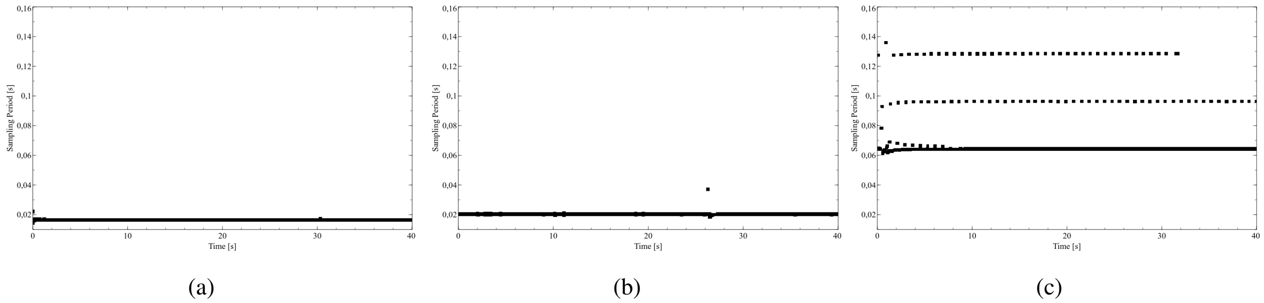


Figure 17: Sample periods for different sensors; (a) A.R.T. infrared tracker; (b) Faro CMM arm; (c) Square marker tracker

	Norm		PCA	
	Mean	Std. Dev.	Mean	Std. Dev.
Path *A*	≈ 0 ms	0.016 ms	≈ 0 ms	0.017 ms
Path *B*	0.029 ms	1.0 ms	0.014 ms	0.96 ms

Table 5: Mean error and standard deviation for norm and PCA based relative lag estimation

$\Delta t = 32.12$ ms and noise according to $X \sim N_3(\mathbf{0}, \Sigma_{\text{ART}})$ was added to the signal. The number of samples M for each simulation was set to $M = 10000$.

Simulation The simulation procedure was performed twice involving different dimensionality reduction components. The first simulation was performed using the static norm computation, whereas the second simulation used the adaptive PCA based projection.

Result and interpretation For each simulation the mean and standard deviation of the relative lag error as compared to the a priori known true value is computed. The results of the four trials (two path with two dimensionality reductions each) are shown in table 5.

Overall the quality of the simulated calibrations is still beyond any results achievable in reality, which indicates that the signal perturbation is not yet realistic. Nevertheless this first result already implies that for low SNR scenarios, the norm based dimensionality reduction may be less robust than the PCA based dimensionality reduction, whereas in high SNR scenarios both perform equally well.

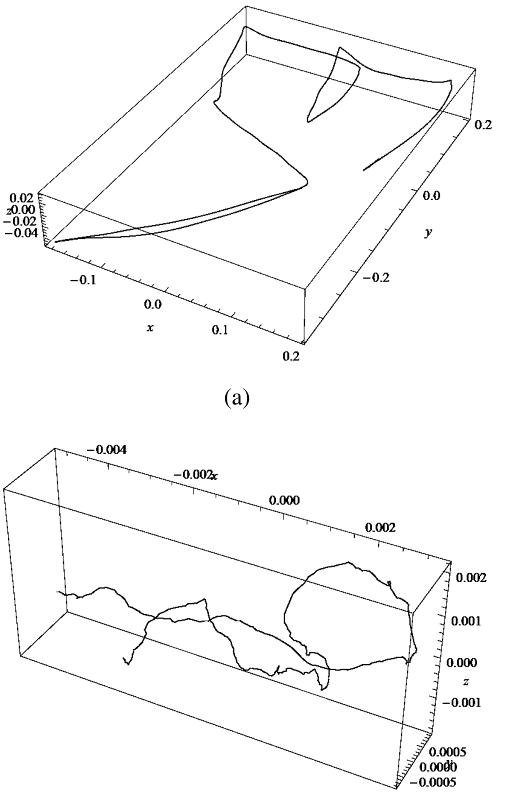
8 Summary and Future Work

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. Furthermore the importance of the correction for a registration error analysis was demonstrated. We discussed the feasibility of the method as an online recalibration method integrated into the UBITRACK framework, as well as the importance of such a method in the context of ubiquitous tracking. We presented a statistical simulation framework that enables both evaluation of the influence of various sensor and signal parameters on the calibration quality and enables "one-factor-at-a-time" evaluation of the various procedure components. Future work will, as already mentioned, focus on the influence of the user on the quality of the measured data and the resulting consequences on the calibration.

A further venue of inquiry is to apply the temporal calibration method to unregistered sensor data. By comparing signals derived from the direct measurements of each sensors, such as the instantaneous velocity derived from the tracked position, sensors for with the spatial relationship is not known could be compared. Future work will investigate approaches to compare sensor signals without known spatial registration.

9 Acknowledgments

This work was partially supported by the German BMBF project AVILUS (grant 01M09001V) and the Bundesministerium für Bildung und Forschung, KMU innovativ Verbundprojekt Asyntra: Entwicklung eines nicht periodischen, asynchronen Trackingsystems mit Kameras unterschiedlicher Eigenschaften (FKZ: 01 I S09034A/B).



(a)

(b)

Figure 18: Two paths for dimensionality reduction evaluation; (a) Distinct "calibration" movement; (b) Physiological hand tremor (all axes in meters.)

References

[AB94] Ronald Azuma and Gary Bishop, *Improving Static and Dynamic Registration in an Optical See-through HMD*, SIGGRAPH '94 Proceedings of the 21st annual conference on Computer graphics and interactive techniques, 1994, DOI 10.1145/192161.192199, pp. 197–204, ISBN 0-89791-667-0.

[ASB04] Michael Aron, Gilles Simon, and Marie-Odile Berger, *Handling Uncertain Sensor Data in Vision-Based Camera Tracking*, Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004, DOI 10.1109/ISMAR.2004.33, pp. 58–67, ISBN 0-7695-2191-6.

[ASB07] Michael Aron, Gilles Simon, and Marie-Odile Berger, *Use of Inertial Sensors to Support Video Tracking*, Computer Animation and Virtual Worlds **18** (2007), no. 1, 57–68, ISSN 1546-427X, DOI 10.1002/cav.161.

[BH81] Ronald E. Boucher and Joseph C. Hassab, *Analysis of Discrete Implementation of Generalized Cross Correlator*, IEEE Transactions on Acoustics, Speech and Signal Processing **29** (1981), no. 3, 609 – 611, ISSN 0096-3518, DOI 10.1109/TASSP.1981.1163623.

[BS08] Gabriele Bleser and Didier Stricker, *Advanced Tracking through Efficient Image Processing and Visual–Inertial Sensor Fusion*, Computers and Graphics **33** (2008), no. 1, 59–72, ISSN 0097-8493, DOI 10.1016/j.cag.2008.11.004.

[BSP$^+$06] Martin Bauer, Michael Schlegel, Daniel Pustka, Nassir Navab, and Gudrun Klinker, *Predicting and estimating the accuracy of n-occular optical tracking systems*, IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006. ISMAR 2006., 2006, DOI 10.1109/ISMAR.2006.297793, pp. 43–51, ISBN 1-4244-0650-1.

[Car81] G. Clifford Carter, *Time Delay Estimation for Passive Sonar Signal Processing*, IEEE Transactions on Acoustics Speech and Signal Processing **29** (1981), no. 3, 463–470, ISSN 0096-3518, DOI 10.1109/TASSP.1981.1163560.

[CH06] M. G. Cox and P. M. Harris, *Software Support for Metrology Best Practice Guide No. 6: Uncertainty Evaluation*, Tech. report, National Physical Laboratory. Reino Unido, 2006.

[CHS04] M. G. Cox, P. M. Harris, and I. M. Smith, *Software specifications for uncertainty evaluation*, Tech. Report NPL Report DEM-ES-010, National Physical Laboratory, 2004.

[Dan99] K. Daniilidis, *Hand-Eye Calibration Using Dual Quaternions*, The International Journal of Robotics Research **18** (1999), no. 3, 286–298, ISSN 0278-3649, DOI 10.1177/02783649922066213.

[DFG01] Arnaud Doucet, Nando De Freitas, and Neil Gordon (eds.), *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001, ISBN 978-1-4419-2887-0.

[DW88] Hugh F. Durrant-Whyte, *Sensor Models and Multisensor Integration*, International Journal of Robotics Research **7** (1988), no. 6, 97–113, ISSN 0278-3649, DOI 10.1177/027836498800700608.

Citation
Manuel Huber, Michael Schlegel and Gudrun Klinker, *Temporal Calibration in Multisensor Tracking Setups*, Journal of Virtual Reality and Broadcasting, 11(2014), no. 3, March 2014, urn:nbn:de:0009-6-38778, ISSN 1860-2037.

[Fod02] Imola K. Fodor, *A Survey of Dimension Reduction Techniques*, Tech. Report UCRL-ID-148494, Lawrence Livermore National Laboratory, 2002.

[FS86] Antoni Fertner and Anders Sjolund, *Comparison of various time delay estimation methods by computer simulation*, IEEE Transactions on Acoustics, Speech and Signal Processing **34** (1986), no. 5, 1329–1330, ISSN 0096-3518, DOI 10.1109/TASSP.1986.1164930.

[GFG11] Ángel F. García-Fernández and Jesús Grajal, *Asynchronous particle filter for tracking using non-synchronous sensor networks*, Signal Processing **91** (2011), no. 10, 2304–2313, ISSN 0165-1684, DOI 10.1016/j.sigpro.2011.04.013.

[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*, Proceedings of the 9th International Symposium on Mixed and Augmented Reality (ISMAR), 2010, DOI 10.1109/ISMAR.2010.5643564, pp. 157–163, ISBN 978-1-4244-9343-2.

[Hol97] Richard L. Holloway, *Registration Error Analysis for Augmented Reality*, Presence **6** (1997), no. 4, 413–432.

[Hor87] Berthold K. P. Horn, *Closed Form Solutions of Absolute Orientation Using Unit Quaternions*, Journal of the Optical Society of America A **4** (1987), no. 4, 629–642, ISSN 1084-7529, DOI 10.1364/JOSAA.4.000629.

[HPK+07] Manuel Huber, Daniel Pustka, Peter Keitler, Florian Echtler, and Gudrun Klinker, *A System Architecture for Ubiquitous Tracking Environments*, Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR), 2007, DOI 10.1109/ISMAR.2007.4538849, pp. 1–4, ISBN 978-1-4244-1749-0.

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

[JK05] Simon J. Julier and Jeffrey K.Uhlmann, *Fusion of time delayed measurements with uncertain time delays*, Proceedings of the 2005 American Control Conference, vol. 6, IEEE, 2005, DOI 10.1109/ACC.2005.1470607, pp. 4028–4033, ISBN 0-7803-9098-9.

[JLS97] Marco C. Jacobs, Mark A. Livingston, and Andrei State, *Managing Latency in Complex Augmented Reality Systems*, Proceedings of the Symposium on Interactive 3D Graphics, ACM, 1997, DOI 10.1145/253284.253306, pp. 49–54, ISBN 0-89791-884-3.

[Joh97] Björn Johansson, *Multidimensional Signal Recognition, Invariant to Affine Transformation and Time-Shift, Using Canonical Correlation*, Tech. Report LiTH-ISY-EX-1825, Linköping University, SE-581 83 Linköping, Sweden, 1997.

[JS93] Giovanni Jacovitti and Gaetano Scarano, *Discrete Time Techniques for Time Delay Estimation*, IEEE Transactions on Signal Processing **41** (1993), no. 2, 525–533, ISSN 1053-587X, DOI 10.1109/78.193195.

[KB99] Hirokazu Kato and Mark Billinghurst, *Marker tracking and HMD calibration for a video-based augmented reality conferencing system*, Proc. 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), IEEE, 1999, DOI 10.1109/IWAR.1999.803809, pp. 85–94, ISBN 0-7695-0359-4.

[Kei11] Peter Keitler, *Management of Tracking And Tracking Accuracy in Industrial Augmented Reality Environments*, Ph.D. thesis, Technische Universität München, 2011.

[KSK08] Peter Keitler, Michael Schlegel, and Gudrun Klinker, *Indirect Tracking to Reduce Occlusion Problems*, Advances in Visual Computing, Fourth International Symposium, ISVC 2008 Las Vegas, USA, December 1-3 (Berlin), vol. 2, Lecture Notes in Computer Science, no. 5359, Springer, 2008, DOI 10.1007/978-3-540-89646-3_22, pp. 224–235, ISBN 978-3-540-89645-6.

[LBMN09] Sebastian Lieberknecht, Selim Benhimane, Peter Meier, and Nassir Navab, *A Dataset and Evaluation Methodology for Template-Based Tracking Algorithms*, Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality, IEEE Computer Society, 2009, 10.1109/ISMAR.2009.5336487, pp. 145–151, ISBN 978-1-4244-5390-0.

[Lip71] Olof Lippold, *Physiological Tremor*, Scientific American **224** (1971), no. 3, 65–73, ISSN 0036-8733.

[MSW+03] Asa MacWilliams, Christian Sandor, Martin Wagner, Martin Bauer, Gudrun Klinker, and Bernd Brügge, *Herding Sheep: Live System Development for Distributed Augmented Reality*, Proceedings of the 2nd International Symposium on Mixed and Augmented Reality (ISMAR'03), 2003, DOI 10.1109/ISMAR.2003.1240695, pp. 123–132, ISBN 0-7695-2006-5.

[NBP+07] Joseph Newman, Alexander Bornik, Daniel Pustka, Florian Echtler, Manuel Huber, Dieter Schmalstieg, and Gudrun Klinker, *Tracking for Distributed Mixed Reality Environments*, Proc. IEEE VR 2007 Workshop on 'Trends and Issues in Tracking for Virtual Environments' (Aachen, Germany), Shaker Verlag, 2007, ISBN 978-3-8322-5967-9.

[NIH01] Joseph Newman, David Ingram, and Andy Hopper, *Augmented reality in a wide area sentient environment*, IEEE and ACM International Symposium on Augmented Reality (ISAR'01), IEEE, 2001, DOI 10.1109/ISAR.2001.970517, pp. 77–86, ISBN 0769513751.

[NWB+04] Joseph Newman, Martin Wagner, Martin Bauer, Asa MacWilliams, Thomas Pintaric, Dagmar Beyer, Daniel Pustka, Franz Strasser, Dieter Schmalstieg, and Gudrun Klinker, *Ubiquitous Tracking for Augmented Reality*, Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04), 2004, DOI 10.1109/ISMAR.2004.62, pp. 192–201, ISBN 0-7695-2191-6.

[PHBK06] Daniel Pustka, Manuel Huber, Martin Bauer, and Gudrun Klinker, *Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems*, Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06), 2006, DOI 10.1109/ISMAR.2006.297799, pp. 88–97, ISBN 1-4244-0650-1.

[PK08] Daniel Pustka and Gudrun Klinker, *Dynamic Gyroscope Fusion in Ubiquitous Tracking Environments*, Proceedings of the 7th International Symposium on Mixed and Augmented Reality (ISMAR), 2008, DOI 10.1109/ISMAR.2008.4637317, pp. 13–20, ISBN 978-1-4244-2840-3.

[PPH+12] Daniel Pustka, Frieder Pankratz, Jan-Patrick Hülß, Jochen Willneff, Manuel Huber, and Gudrun Klinker, *Optical Outside-In Tracking using Unmodified Mobile Phones*, 11th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2012), 2012, DOI 10.1109/ISMAR.2012.6402542, ISBN 978-1-4673-4660-3.

[PT01] Wayne Piekarski and Bruce H. Thomas, *Tinmith-Metro: New Outdoor Techniques for Creating City Models with an Augmented Reality Wearable Computer*, Proceedings of the 5th International Symposium on Wearable Computers, 2001, DOI 10.1109/ISWC.2001.962093, pp. 31–38, ISBN 0-7695-1318-2.

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

[RD06] Gerhard Reitmayr and Tom W. Drummond, *Going out: Robust Model-Based Tracking for Outdoor Augmented Reality*, IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006. ISMAR 2006, 2006, DOI 10.1109/ISMAR.2006.297801, pp. 109–118, ISBN 1-4244-0650-1.

[Sch06] Michael Schlegel, *Predicting the Accuracy of Optical Tracking Systems*, Master's thesis, TU München, Munich, Germany, 2006.

[Sch11] Michael Schlegel, *Zeitkalibrierung in Augmented Reality Anwendungen [Time calibration in Augmented Reality applications]*, Ph.D. thesis, Technische Universität München, 2011.

[SS04] Bernd Schwald and Helmut Seibert, *Registration Tasks for a Hybrid Tracking System for Medical Augmented Reality*, Journal of WSCG **12** (2004), no. 1-3, 411–418, ISSN 1213-6972.

[VMAR07] Mahesh Vemula, Joaquín Míguez, and Antonio Artés-Rodriguéz, *A sequential monte carlo method for target tracking in an asynchronous wireless sensor network*, 4th Workshop on Positioning, Navigation and Communication, 2007. WPNC'07, 2007, DOI 10.1109/WPNC.2007.353612, pp. 49–54, ISBN 1-4244-0871-7.

[WB04] Greg Welch and Gary Bishop, *An Introduction to the Kalman filter*, Tech. report, University of North Carolina at Chapel Hill, Department of Computer Science, 2004, TR 95-041.

[Wel96] Gregory Francis Welch, *SCAAT: Incremental Tracking with Incomplete Information*, Ph.D. thesis, University of North Carolina at Chapel Hill, 1996, TR96-051.

[WHK+10] Christian Waechter, Manuel Huber, Peter Keitler, Michael Schlegel, Daniel Pustka, and Gudrun Klinker, *A Multisensor Platform for Wide-Area Tracking*, 9th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2010), 2010, DOI 10.1109/ISMAR.2010.5643604, pp. 275–276, ISBN 978-1-4244-9343-2.

[ZA05] Yushi Zhang and Waleed H. Abdulla, *A comparative study of time-delay estimation techniques using microphone arrays*, Tech. report, Department of Electrical and Computer Engineering, The University of Auckland, 2005, School of Engineering Report 619.