Optical Outside-In Tracking using Unmodified Mobile Phones

Daniel Pustka[*], Jan-Patrick Hülß, Jochen Willneff
Advanced Realtime Tracking GmbH

Frieder Pankratz[†], Manuel Huber[‡], Gudrun Klinker
Technische Universität München

Abstract

Marker-based optical outside-in tracking is a mature and robust technology used by many AR, VR and motion capture applications. However, in small environments the tracking cameras are often difficult to install. An example scenario are ergonomic studies in car manufacturing, where the motion of a worker needs to be tracked in small spaces such as the trunk of a car.

In this paper, we describe how to extend the tracking volume in small, cluttered environments using small and flexible wireless cameras in form of unmodified mobile phones that can quickly be installed. Since the mobile phones are not synchronized with the main tracking cameras, we describe several modifications to the tracking algorithms, such as inter-frame interpolation, the replacement of the least-squares adjustment by a Kalman filter and the integration of rolling-shutter compensation.

To support the quick setup of mobile phones while the tracking system is running, the system is extended by an on-line calibration technique that determines the extrinsic camera parameters without requiring a dedicated calibration step.

Index Terms: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 Introduction

Marker-based optical outside-in tracking systems are based on a simple and robustly working principle: targets consisting of rigid arrangements of multiple spherical or circular flat markers are observed by multiple synchronized cameras installed at known locations in the environment. Markers are either made of passive retro-reflective material, illuminated by a light source at the cameras or they consist of an active LED, often synchronized to the exposure time of the cameras. In order to prevent distraction of the user, infrared light is generally used. By using short exposure times and/or narrow band-pass filters, this results in images that are virtually black except for the markers, allowing for easy segmentation. Many commercial systems built on this principle exist, including our own products, and are frequently used in VR, AR and motion capture applications.

The work we describe here is motivated by scenarios in the automotive industry, where ergonomic studies need to be performed in small and cluttered spaces, such as the trunk of a car [15] or the cabin of a truck. Here, the installation of conventional tracking cameras is difficult, due to their size and cabling needs. Also, a more flexible setup is needed so that additional cameras can easily be installed and moved to provide optimal coverage of the tracking volume.

To allow an ad-hoc installation of additional cameras in small and cluttered areas of the tracking volume, such cameras have to be reasonably small and should require no or only light cabling. However, without cables, the exact synchronization of multiple cameras is difficult, and can only be realized by customized hardware making use of radio or optical data transmission. As an alternative to the hardware-based synchronization, tracking algorithms can be developed that explicitly integrate asynchronous measurements.

In the recent years, mobile phones have drawn a lot of attention in the AR community, as they integrate camera, location sensors, display, power supply, communication and increasingly powerful processing capabilities in a small form factor. Due to the large production quantities, they come at a price lower than most previous mobile AR setups. For the same reasons, mobile phones are ideal candidates for additional cameras to use in the small tracking spaces describe above. By activating the integrated video illumination, the use of retro-reflective markers is possible, as they appear noticeably brighter than the rest of the environment.

Approach

In this paper, we describe an optical outside-in tracking system that makes use of small wireless asynchronous cameras, which can be used to extend a conventional tracking system or to provide outside-in tracking using the asynchronous cameras alone. This system is realized using mobile phones.

In order to handle the the integration of unsynchronized cameras, tracking algorithms need to be developed that support asynchronous measurements, and a system architecture must be able to provide reliable timestamps of the individual frames. The approach described in this paper uses network clock synchronization, an inter-frame interpolation scheme and Kalman filter-based 6DoF tracking to overcome these problems.

Camera phones and other low-cost cameras have a rolling shutter that exposes the upper part of an image at a different time than the lower part, resulting in warped images of fast motions. As the difference between the upper and lower image edges can be as high as the inverse of the frame rate, this effect needs to be compensated by the tracking algorithms. The approach described here achieves this by integrating the rolling shutter compensation directly into the Kalman filter measurement equations.

To provide a very flexible camera system requiring minimal user intervention, it should be possible to move individual cameras without having to manually re-calibrate the whole setup. The system presented here provides such functionality by detecting wrongly calibrated cameras and starting an automatic re-calibration process based on observations of targets that are tracked by the remaining cameras.

Related Work

Optical marker-based outside-in tracking is a well studied field. Madritsch and Gervautz [7] describe an early two-camera tracking system using unsynchronized cameras, but simply treat them as synchronized. The system is able to detect active LEDs in an image and compute their 3D position. The system described by Dorfmüller [3] uses synchronized IR cameras and is able to distinguish 6D targets using the known 3D geometry. Pintaric and Kaufmann [10] present yet another system. The algorithm they describe to identify different targets is a simplified version of the one used in our setup. Commercial marker-based optical outside-in tracking systems include our own ART products and many others. Other manufacturers like NDI focus more on high-accuracy measurements rather than flexible camera setups.

[*] e-mail: daniel.pustka@ar-tracking.de
[†] e-mail: pankratz@in.tum.de
[‡] e-mail: huberma@in.tum.de

Other authors have already built asynchronous tracking systems. The classical approach towards integrating single asynchronous observations into an extended Kalman filter is SCAAT [17], developed for an inside-out tracking system. Mulder et al. [8] present a two-camera tracking system using unsynchronized FireWire cameras. Their approach to interpolate 2D measurements is similar to the one we use to initialize our Kalman filter-based tracking. The approach described by Rasmussen et al. [12] integrates single asynchronous measurements in a SCAAT-like fashion, but using cameras with a global shutter.

Rolling shutter compensation has been the subject of many earlier publications. Ait-Aider et al. [9] simultaneously estimate pose and velocity from a single camera using a measurement model similar to ours. The mobile phone version of PTAM [6] contains rolling shutter compensation based on the velocity of 2D features. Baker et al. [1] remove wobble in rolling shutter images resulting from high-frequency camera jitter and present an auto-calibration technique for the rolling shutter time.

Outline The paper is structured as follows: First, we describe the general principles of marker-based optical outside-in tracking. We then present the system architecture using mobile phones. In section 4, we describe in detail the Kalman filter-based asynchronous algorithm, followed by a description of the automatic recalibration of individual cameras. Section 6 presents an evaluation of our approach. Finally, the approach is evaluated in a motion capture scenario where the volume of a conventional tracking system is extended using mobile phones, and in an application shown at last year's ISMAR tracking competition.

2 Marker-based Optical Outside-In Tracking

Before we start with the asynchronous mobile phone tracking, we quickly introduce the steps that the standard synchronous IR tracking performs. The modifications for asynchronous tracking will be described in the next section.

2.1 Calibration

Before the actual 6D tracking takes place, three different calibration steps need to be performed:

Camera Calibration The purpose of the camera calibration is to determine the intrinsic camera parameters of each camera. The camera calibration is part of the manufacturing process and is derived from a set of convergent images of a reference body with accurately known 3D points. Providing the assigned image coordinates of the reference points the intrinsic camera parameters are then computed according to the lens model described in [18] and stored in the camera memory. During the tracking process each measured image coordinate can now be corrected for the influence of the intrinsic parameters of the camera. Figure 1 shows the image acquisition of the reference body with a mobile phone.

Room Calibration After the cameras are set up, the extrinsic camera parameters, i.e. the relative camera poses, need to be determined. The procedure requires the user to move a calibration wand, consisting of two retro-reflective markers, in front of the cameras. When enough data is collected, an initial room calibration is computed using epipolar geometry of camera pairs, and then refined by global bundle adjustment. Figure 2 shows a calibration wand and an angle tool angle tool for coordinate system definition.

Body Calibration The tracking system detects and uniquely identifies 6D tracking targets ("bodies") by the rigid arrangement of retroreflective markers. Therefore, each tracking target needs to be made known to the tracking system. This is done by moving the target within the tracking volume. The system then detects a set of 3D markers that have fixed relative distances and adds them to the target list. An example of a 6D target is shown in figure 3.

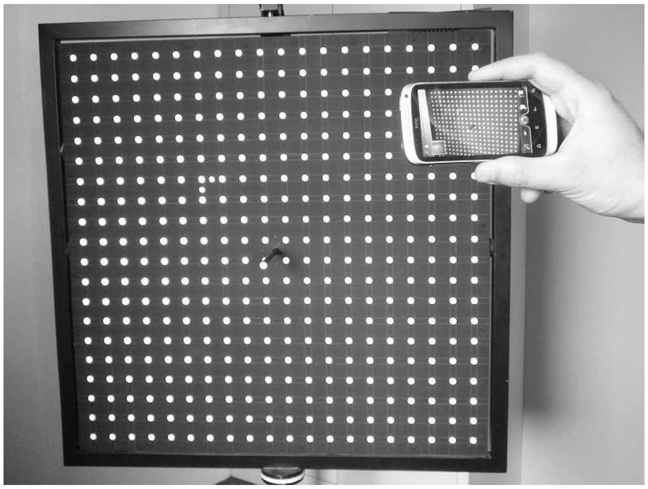

Figure 1: Image acquisition of the calibration board with a mobile phone

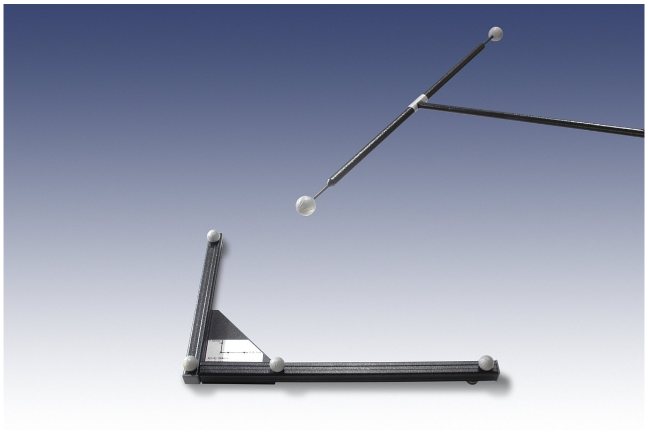

Figure 2: Room calibration set consisting of calibration wand and angle tool for coordinate system definition

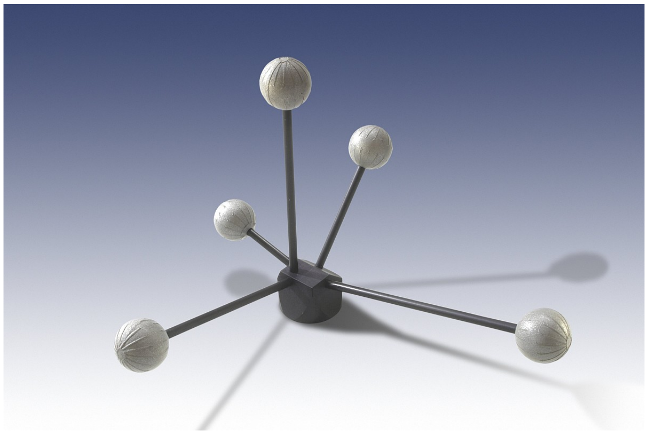

Figure 3: 6D tracking target with five retroreflective markers

2.2 Tracking

The tracking process at runtime is roughly divided into the following steps:

Image Segmentation Due to the retro-reflective markers, the active infrared illumination and the short exposure times, the images taken by the cameras are mostly black with bright white spots as the images of the markers. Therefore, simple thresholding and subpixel-accurate calculation of the segment centers determines the 2D marker positions in the image. This processing takes place inside the camera and the detected 2D marker locations are transmitted to the controller computer using UDP packages.

3D Marker Detection In order to identify 6D targets during the system runtime and during body calibration, the 3D markers in the tracking volume need to be detected. This is done by searching for epipolar correspondences in camera pairs and then eliminating ambiguous assignments. This approach is purely based on the 2D position of the markers in the image, as, unlike in feature-based markerless tracking, all measurements are similar and thus no reasonable feature descriptor can be computed from the image.

6D Target Detection In order to track 6D targets, their known geometry must be found in the set of detected 3D markers. The approach we use is an improved version of the one described in [10]. In short, it first computes the distances between all observed 3D markers and compares them to the distances of the known body geometries. After some filtering, the remaining distances are put in a graph on which maximal cliques are detected. The approach requires all sets of four markers to have unique distances, within some similarity threshold.

Prediction-based Tracking In order to save processing time and to make the tracking more robust, the system tries to track detected 3D markers and 6D targets from frame to frame without going through the detection steps described above. For this, the positions of targets and markers are predicted into the current frame and associated with the available measurements. Finally, a standard least-squares adjustment (see [11] for details) is run for all markers and targets. Only those measurements that did not match a prediction or where the adjustment failed are fed into the 2D interpolation algorithm.

3 Mobile Phone Setup

One of our goals in this research project was to use unmodified off-the-shelf mobile phones. To enable the application to run on a broad set of mobile phones, we used the Android operating system for development.

Mobile Phone Camera The Standard tracking cameras use infrared illumination and infrared pass filter to capture gray images containing only the retro-reflective markers. Applying the same concept to the mobile phone cameras would require hardware modifications, as most phones have good infrared blocking filters installed. However, early experiments have shown that the while LED video illumination, frequently found in mobile phones, has enough luminosity to easily detect retroreflective markers at a distance of up 3 meters. An example image shot through a phone camera is shown in figure 4.

As the automatic exposure mechanism of the camera is always active, the environment is still visible. The retroreflective markers are bright spots in the image, but many interfering reflections are present, from, for example, rounded edges of tables, white areas or other reflecting surfaces. However, simple shape analysis can discard most these interferences.

Since Android version 2.2, the situation has slightly improved, as the camera API includes an exposure compensation parameter that allows us to reduce the exposure time of the images manually. Therefore, the contrast between the markers and the environment is increased and many interferences disappear. However, exact control over the camera exposure time still is not possible.

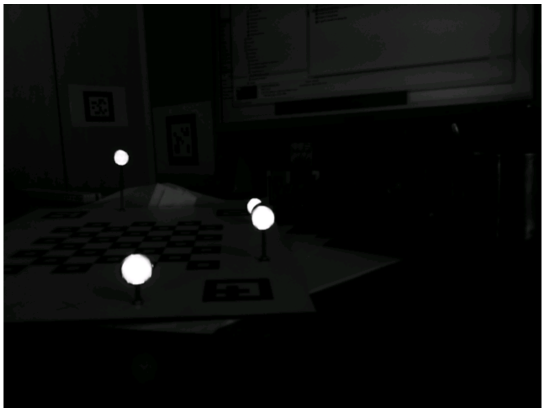

Figure 4: Markers in mobile phone image

Time synchronization In order to achieve a stable 6DoF tracking from asynchronously acquired camera images, the exact moment of the recording has to be determined by the means of a timestamp assigned to each image. To synchronize the clocks of multiple phones, we implemented a method based on the network protocol SNTP (RFC 4330). In the implemented procedure, each device synchronizes its own clock to a central selected master clock, the time server. Using an SNTP query, each device determines the current difference between its local clock and the central master clock. This is repeated at regular intervals to further compensate for frequency differences and frequency drift of the local clocks. To correct remaining measurement noise caused by network delays, we added a double-exponential smoothing method.

Tracking Software The Android application is split into a background service and a GUI application that is used to configure and control the tracking. The main tasks are handled by the background service, including time synchronization, camera access, 2D marker detection and network communication.

To reduce the latency when generating timestamps and to avoid unnecessary image copying, the image processing is done in native C++, using an unofficial camera API. The rest of the application uses the standard Java API provided by the Android SDK/NDK. While this does not conform to the standard Android API, the application should be able to run on every Android 2.2/2.3 device.

The mobile phones communicated with the central tracking server using UDP packets over WiFi. WiFi is supported by almost all mobile phones, and offers the highest transmission rate and lowest latency of all available communication channels.

To save computing power as well as electricity, we can put the mobile phone into stand-by mode while continuing the tracking. This extends the operation time of the mobile phones to about 2 hours, as the output on the display is not needed at runtime.

4 Asynchronous Tracking

In order to support asynchronous cameras, the standard tracking procedure, as described above, had to be substantially modified. Compared to the standard synchronous procedure, the biggest differences are the replacement of the least-squares adjustments for 3D marker positions and 6D target poses by Kalman filters and the addition of an interpolation module to generate synchronous 2D marker positions for the epipolar search. The basic structure of our asynchronous tracking is shown in figure 5. In brief, the tracking is composed of the following modules:

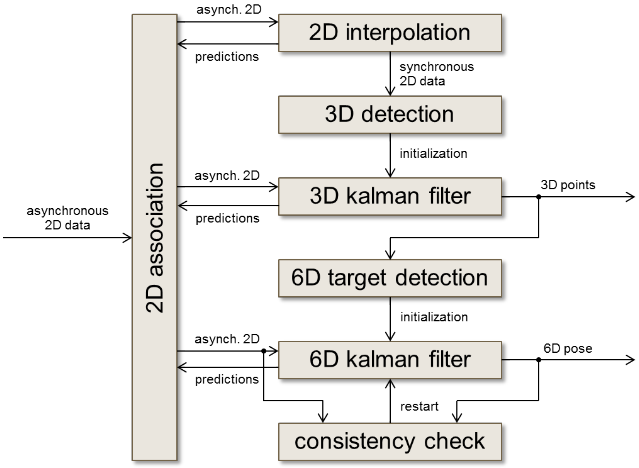

Figure 5: Modules of the asynchronous tracking process.

2D Association The 2D association module receives the asynchronous 2D measurements from a single camera as well as the predictions made by the Kalman filters and the 2D interpolation. Measurements that can be unambiguously assigned to predictions of tracked 3D markers or 6D targets are passed on to the respective Kalman filters. Measurements that cannot be assigned or correspond to measurements seen in the previous frames without belonging to a marker are given to the 2D interpolation.

2D Interpolation In order to allow the tracking system to detect 3D markers in images taken by unsynchronized cameras, a timestamp-based interpolation was implemented to artificially generate synchronous marker positions. For this purpose, 2D markers are tracked in subsequent images of each camera by linearly extrapolating the 2D positions in two subsequent frames and passing the result as additional predictions to the 2D association module in the following frame.

In regular intervals, synchronous frames are generated for all cameras by linearly interpolating the 2D positions of the markers between two frames. The resulting synchronized images are passed to the 3D marker detection. 2D tracking and interpolation only takes place for measurements that do not (yet) belong to 3D markers or 6D targets. In these cases, the Kalman filters produce more reliable predictions.

3D Detection In order to detect new 3D markers, the same algorithm is used as in the synchronized case, except that it is provided with interpolated measurements. For each detected 3D marker, a new 3D Kalman filter is initialized.

3D Kalman Filter For each 3D marker that has not yet been identified as belonging to a 6D target, an individual Kalman filter is instantiated. The Kalman filter directly integrates single asynchronous 2D measurements and estimates the current position, velocity and acceleration of the marker. This information is again used to generate predictions of the 2D marker positions. In regular intervals, the 3D Kalman filters generate synchronous estimates of all available 3D markers for detection of 6D targets. The 3D Kalman filter is a simplified version of the 6D Kalman filter described in section 4.2, without the rotational elements.

6D Target Detection The distance-based 6D target detection algorithm is the same as in the synchronous case. For each newly detected target, a new 6D Kalman filter is instantiated.

6D Kalman Filter Each 6D target is handled by a 6D Kalman filter, which integrates single 2D measurements into its internal state and generates predictions of its individual markers. The 6D Kalman filter is described in detail in section 4.2.

Consistency Check The 6D and 3D Kalman filters are regularly checked for consistency. This includes checks for the consistency with the measurements as well checks of the internal state. More details are described in section 4.4. When the consistency check fails, all measurement associations are removed and the target has to be detected again, starting from the 2D interpolation.

4.1 2D Association

Data association, i.e. the mapping of observed 2D features to known objects, is a crucial aspect of any tracking algorithm, as both missing or wrong associations may lead to tracking interruption. In the case of a marker-based outside-in tracking system, we need to associate the 2D measurements in the current frame with the 2D predictions from 6D targets, single 3D markers and 2D predictions generated by the interpolation.

Compared to our standard IR cameras, mobile phones have both a lower frame rate and a lower accuracy. This makes the association more difficult, as higher distances between predicted and measured position need to be handled, resulting in more ambiguities in cluttered situations. Thus, the relatively simple strategy used in the existing tracking system, based on fixed association radii and strict ambiguity rejection, could not be used.

The new approach developed for the mobile phone tracking uses a much higher radius. It associates predictions with measurements whose rays pass a 50 mm sphere around the 3D location of the prediction or, in case of 2D predictions for interpolation, with measurements that have a distance of up to 30 pixels. Within this radius, the best matching measurement is selected. However, in order to avoid ambiguous assignments, the algorithm discards associations where a second measurement is closer than twice the distance of the selected one.

4.2 6D Kalman Filter Modeling

The task of the Kalman filter is to recursively compute the current state of a 6D target, including pose, velocity and acceleration, from single 2D measurements. In the following discussion, we assume that the reader has basic knowledge about the prediction-correction cycle of the filter.

Motion Model As in our earlier paper [11], where we describe non-linear least-squares adjustment, the extended Kalman filter (EKF) formulation is based on the following transformation from target to world coordinates:

$$x_w = \exp([\Delta r]_\times) R_{tw} x_t + p_{tw} \tag{1}$$

where $\exp([\Delta r]_\times)$ is an exponential map from 3-vector Δr to a rotation matrix that is computed from Δr using the Rodrigues formula. To simplify linearization, the filter assumes that Δr is zero at the beginning of each update. The full rotation matrix R_{tw} is stored outside the filter and updated according to $R_{tw,n+1} = \exp([\Delta r]_\times) R_{tw,n}$ after each filter update and Δr is again assumed to be zero. This formulation is similar to the one found in [13], except that the exponential map representation is only used for the rotation parts.

Our motion model is based on the assumption of constant acceleration in both translation and rotation. Consequently the state is composed of the following 6 elements, where each element itself is a 3-vector, resulting in 18 elements in total.

$$s = (p_{tw}, \dot{p_{tw}}, \ddot{p_{tw}}, \Delta r, \dot{\Delta r}, \ddot{\Delta r}) \tag{2}$$

The Kalman filter time updates are computed according to the following equations:

$$\begin{aligned} s_{n+1} &= A(\Delta t)\, s_n & (3) \\ P_{n+1} &= A(\Delta t)\, P_n A^T(\Delta t) + Q(q_p, q_r, \Delta t) & (4) \end{aligned}$$

with

$$\begin{aligned} A(\Delta t) &= \begin{bmatrix} A_{acc}(\Delta t) & 0 \\ 0 & A_{acc}(\Delta t) \end{bmatrix} \\ Q(q_p, q_r, \Delta t) &= \begin{bmatrix} q_p\, Q_{acc}(\Delta t) & 0 \\ 0 & q_r\, Q_{acc}(\Delta t) \end{bmatrix} \end{aligned}$$

As both translation and rotation increments are equally modeled assuming linear constant acceleration, the same sub-matrices A_{acc} and Q_{acc} can be used:

$$\begin{aligned} A_{acc}(\Delta t) &= \begin{bmatrix} I_3 & \Delta t\, I_3 & \frac{\Delta t^2}{2} I_3 \\ 0 & I_3 & \Delta t\, I_3 \\ 0 & 0 & I_3 \end{bmatrix} \\ Q_{acc}(\Delta t) &= \begin{bmatrix} \frac{\Delta t^5}{20} I_3 & \frac{\Delta t^4}{8} I_3 & \frac{\Delta t^3}{6} I_3 \\ \frac{\Delta t^4}{8} I_3 & \frac{\Delta t^3}{3} I_3 & \frac{\Delta t^2}{2} I_3 \\ \frac{\Delta t^3}{6} I_3 & \frac{\Delta t^2}{2} I_3 & \Delta t\, I_3 \end{bmatrix} \end{aligned}$$

Measurement Model At the measurement step, our Kalman filter directly integrates single 2D observations of a marker in a given camera. The projection process can be decomposed into the following transformations:

R_{tw}, p_{tw} is the 6D transformation from target to world coordinates as estimated by the Kalman filter

R_{wc}, p_{wc} is the 6D transformation from world to camera coordinates. This transformation was determined during the room calibration process.

pinhole(x_c) is the pinhole perspective transformation from camera coordinates into image coordinates. As we un-distort and normalize all image coordinates before tracking, this simply is a division by the negative z coordinate.

This results in the following measurement equation for the Kalman filter:

$$x_i = \text{pinhole}(\, R_{wc}\, (\exp([\Delta r]_\times)\, R_{tw}\, x_t + p_{tw}\,) + p_{wc}\,) \qquad (5)$$

where x_t is the 3D position of the observed marker in target coordinates, as determined by the body calibration and

$$\text{pinhole}(x) = -\frac{1}{x_3} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The required Jacobian matrix of the measurement equation wrt. the state vector s is computed using the chain rule, assuming $\Delta r = 0$:

$$H = \frac{\partial x_i}{\partial s} = J_{\text{pinhole}}(x_c)\, J_{wc}\, J_{tw} \qquad (6)$$

with

$$\begin{aligned} x_c &= R_{wc}\, (R_{tw}\, x_t + p_{tw}\,) + p_{wc} \\ J_{\text{pinhole}}(x) &= \frac{-1}{x_3^2} \begin{bmatrix} x_3 & 0 & -x_1 \\ 0 & x_3 & -x_2 \end{bmatrix} \\ J_{wc} &= R_{wc} \\ J_{tw} &= \begin{bmatrix} I_3 & 0_{3\times6} & [R_{tw}\, x_t]_\times & 0_{3\times6} \end{bmatrix} \end{aligned}$$

where x_c is the marker position transformed to camera coordinates and $[x]_\times$ denotes the skew-symmetric matrix.

To actually apply a single 2D measurement, we evaluate the measurement equation 5 using $\Delta r = 0$ and compute the Jacobian as in eq. 6. Then the usual EKF measurement update procedure as described in the literature (e.g. [4] or [16]) is applied.

Rolling Shutter Compensation The measurement model given above is suitable for cameras with global shutter. To compensate the rolling shutter used by the mobile phones in the measurement equation, we first computed the shutter time offset Δt_s of each marker relative to the mean exposure time of the image:

$$\Delta t_s = \frac{y - \frac{1}{2}(y_{\text{top}} + y_{\text{bottom}})}{y_{\text{bottom}} - y_{\text{top}}}\, t_{\text{shutter}} \qquad (7)$$

where y is the predicted y-coordinate of the measurement, y_{top} and y_{bottom} are the y-coordinates of the top and bottom of the image and t_{shutter} is a tunable constant that represents the time in which the shutter "rolls" from the top to the bottom (or vice-versa if negative).

In the measurement equation, we shift the predicted measurements according to the target velocity and acceleration in the state:

$$\begin{aligned} x_i \;=\; & \text{pinhole}(\, R_{wc}\, (\exp([\Delta r + \Delta\dot{r}\, \Delta t_s + \frac{1}{2}\Delta\ddot{r}\, \Delta t_s^2]_\times)\, R_{tw}\, x_t\, + \\ & p_{tw} + \dot{p}_{tw}\, \Delta t_s + \frac{1}{2}\ddot{p}_{tw}\, \Delta t_s^2\,) + p_{wc}\,) \qquad (8) \end{aligned}$$

Assuming that the filter update does not significantly change Δt_s, we extend the Jacobian J_{tw} of the target-to-world transformation:

$$J_{tw} = \begin{bmatrix} I_3 & I_3\, \Delta t_s & I_3 \frac{1}{2} \Delta t_s^2 & J_{R_{tw}} & J_{R_{tw}}\, \Delta t_s & J_{R_{tw}} \frac{1}{2} \Delta t_s^2 \end{bmatrix}$$

where $J_{R_{tw}} = [R_{tw}\, x_t]_\times$. Although it was not the goal of our research, given enough measurements, this EKF formulation should in principle be able to estimated pose and velocity from a single frame, similar to the approach presented in [9].

4.3 EKF Integration

When a 6D target is detected, a new Kalman filter is instantiated and initialized with an initial pose computed from the 3D marker positions. Special treatment is implemented to allow the initialization from fast moving targets. In case the prediction-based tracking fails due to wrongly initialized velocities, but the interpolation-based approach is able to detect the target in two consecutive frames, the velocity of the target is computed and used together with the pose to initialize the filter.

When new measurements are received from a camera, all Kalman filters perform a time update step according to equations 3 and 4, and project all marker positions into the camera image.

Similar to the SCAAT [17] approach, associated measurements from the camera are integrated one-by-one into the filter using equations 5, 6 and the usual EKF measurement update equations. Although computationally more expensive than integrating all measurements of a camera in a single update, the approach has the advantage that with each measurement, the filter moves closer to the real state and the effects of the EKF linearization of the non-linear pose estimation problem are reduced.

In systems consisting of both mobile phones and conventional tracking cameras, measurements sometimes arrive out-of-order due to the higher delay of the mobile phones. To improve the response of the filter in cases where measurements arrive that are older than the current state, a number of old filter states is retained and the filter is put back into an old state before the measurements are applied. After the update, also the newer measurements are re-applied to the filter.

4.4 Consistency Check

In order to detect instabilities of the filter, the marker positions are again projected into the camera image after the measurement update and compared to the measured positions. Should the resulting residual error be higher than some threshold, the filter is discarded and tracking has to start from the 2D interpolation again.

A different cause of instability occurs when no or too few measurements are available over a certain time. This can be easily detected by analyzing the traces of the position and orientation parts of the covariance matrix. When the traces exceed certain pre-defined thresholds, the filter is discarded.

4.5 Kalman Filter Tuning

For the Kalman filter, the 2D measurement covariance matrix R and the process noise (q_p, q_r) need to be defined. This process is frequently called the "tuning" of the filter. In practice, R, q_p and q_r can have arbitrary scaling and only the ratios are important.

As we wanted to support mixed setups using both standard IR cameras and mobile phones, we started by optimizing the motion model parameters using a large database of recorded measurements from the standard cameras. These cameras already provide covariance matrices R of their measurements, which fix the scale of the problem. Applying the Kalman filter-based tracking to these measurements, the tracking quality was evaluated in terms of successfully tracked frames and pose jitter. On our measurement database, containing both slow and fast motions, we achieved the best overall results with the parameters

$$q_p = 5000^2 \frac{\text{mm}^2}{s^5} \quad \text{and} \quad q_r = 200^2 \frac{\text{rad}^2}{s^5}$$

(the units have to be read as "squared acceleration per second"). The ratio between q_p and q_r is particularly important. When, for instance, q_r is chosen too high, all measurement errors will be factored into the rotation, resulting in high jitter in the rotation.

In the last step, we determined the measurement noise for the mobile phones. For this, we built a mixed setup with phones and standard cameras and tuned the covariance matrices of the 2D measurements such that the tracking results of both the mixed system and a system consisting of only phones were satisfying. The resulting standard deviations of the mobile phones are approximately a factor of 8 higher than those of the standard cameras.

4.6 Asynchronous Room and Body Calibration

Adapting the room and body calibration algorithms to the asynchronous camera system is straightforward: Lacking a truly asynchronous algorithm for estimating fundamental matrices, we run the standard synchronous room calibration algorithm on interpolated 2D measurements. The body calibration uses a Kalman filter-based asynchronous tracking of 3D points and feeds the current state of the 3D markers in regular intervals into the existing algorithm.

5 Online Recalibration

In highly flexible camera setups, it is undesirable to perform the room calibration whenever a camera is moved. Therefore, a procedure to determine the pose of changed cameras during the tracking was developed. This procedure works in the background during the normal tracking process with no interruption of the measurements. The recalibration splits into three parts: identification of cameras with an incorrect pose, collection of calibration data and camera pose calculation.

Identifying Uncalibrated Cameras All cameras are tested permanently, whether their calibrated pose is correct. A camera is treated as uncalibrated, if the camera does not contribute measurements to the tracking for some time t_{no}, but can identify a target for some time t_{target}. Both parameters are adjustable and set to $t_{no} = 1$ s and $t_{target} = 160$ ms. To identify targets in a single camera, we use an P-n-P algorithm similar to SoftPOSIT [2].

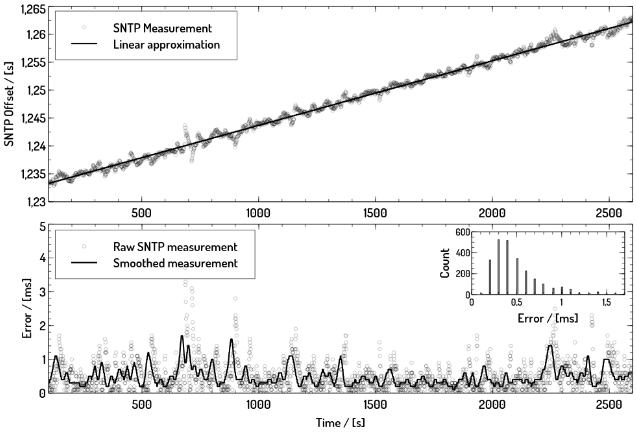

Figure 6: Clock stability

Data Collection and Progress For a camera with incorrect pose, 2D observations of 3D markers are stored with their corresponding 3D position, determined from the 6D pose of a target tracked by the remaining calibrated cameras. These correspondences are recorded for a time t_{data} until some minimum coverage c_{cover} of the image is reached. Both parameters are adjustable and set to $t_{data} = 10$ s and $c_{cover} = 30$ %. A large coverage of the camera image increases the accuracy of the recalibrated camera pose, but requires more time. A progess indicator for the re-calibration is computed based on these parameters and shown to the user.

Pose Calculation The recorded correspondences between 3D positions and observations in the camera image are used in an iterative least-squares adjustment procedure to determine the camera pose. To increase the stability of the system, the new pose is only used if it differs significantly from the old pose.

The described algorithm allows the recalibration of moved cameras with respect to the remaining system. Any number of cameras can recalibrate simultaneously as long as at least two cameras remain calibrated and have an overlapping field of view.

6 Evaluation

6.1 Time Synchronization

For evaluation, a dedicated local time server was set up and synchronized using NTP with 6 international time standards, all equipped with local primary clocks. Before each evaluation, the time server was running for at least a week without interruptions to achieve final stability compared to the reference clocks.

Clock Stability The first evaluation considers the stability and predictability of the local clocks of mobile devices. To rule out environmental influences, the device under test is shielded from light and is kept at a constant temperature by active cooling. The frequency differences of the tested mobile phones relative to the master clock were in the order of magnitude of $10\mu s/s$ (figure 6). After the correction of the SNTP measurements, by a linear clock model, the residual error can be regarded as noise. This justifies the smoothing of the SNTP measurements.

Effect of environmental influences If the mobile phone is not cooled as in the previous evaluation, a significant rise in temperature of the device during operation of the tracking software is noticeable. These temperature changes affect the stability of the clock (see figure 7). In both the cold and in the final warm phase, the clock can be approximated by a linear function. However there is a significant difference in the slopes (frequency differences to the master clock) of about $10\mu s/s$ in a cold state to about $40\mu s/s$ in

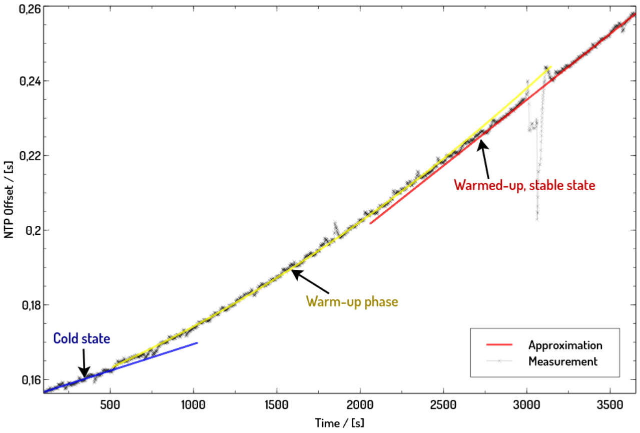

Figure 7: Change in the clock frequency by warming up

the warm state. During the warm-up phase a non-linear change in slope occurs. This frequency drift would require further parameters to describe this behavior. The warm-up phase here lasts about 2000 seconds. Since temperature changes in mobile phones can not be ruled out, a periodic update of the estimated frequency difference is necessary. To obtain linear approximations of sufficient quality an interval of not more than 5% of the expected phase drift (i.e. about 100sec.) seems useful.

Impact of network load Since the network is not used exclusively for SNTP synchronization, the effect of network loading has to be considered. For this purpose, a series of measurements has been systematically disturbed by generating additional network load. After 1000 seconds, the network was saturated by a typical bulk transfer. This increased the overall latency to about 30ms. Here it is apparent that the estimated time difference exhibits an error in the order of the network latency. This is consistent with the SNTP protocol, since its ability to adapt to the specific network characteristics is limited. Nevertheless, while this error in general is no longer tolerable a rough trend with correct slope can still be seen in this area. As long as the network is not saturated, the presented procedure achieved a sufficient degree of synchronicity between the local clock of a mobile phone and the central master clock (mean error of 0.29ms, standard deviation of 0.36ms).

6.2 Static Camera Accuracy

The results of the intrinsic camera calibration allow an estimation of the accuracy potential of mobile phone cameras in static situations. After bundle adjustment, the mean image residual is between 0.05 and 0.1 pixels at VGA resolution. Assuming this order of accuracy for the image coordinate measurements should lead to submillimeter accuracy for a single 3D point in typical camera setup within a hypothetical tracking volume of one cubic meter. For 6D poses, which are calculated from a higher number of observations than used for 3D positioning, the expected accuracy of the tracking results should be even higher. This accuracy is sufficient for the purpose of this project and should enable successful 3D- and 6D tracking as long as suitable observations are provided.

6.3 Dynamic accuracy

To test the dynamic accuracy of the asynchronous tracking system, we compared it to a standard ART tracking system, consisting of six ARTrack2 cameras running at 60Hz. Four cameras where mounted on the ceiling in a half circle. The remaining two cameras where on the floor, further back in the room, looking at center of the tracking volume defined by the other four cameras. The asynchronous tracking system used four LG P990 mobile phones mounted near the standard tracking cameras at the ceiling, about 2.5m from the center of the tracking volume. The two tracking systems where registered into a common coordinate system. We calculated the difference in the pose of a typical target observed by the two tracking systems and used this value as the error in translation and rotation. Before performing the measurements, the time delay between the two tracking systems was estimated using the techniques described in [5, 14] to reduce the error introduced through temporal misalignments.

Set	Speed in m/s	σ in m/s	Pos. error in *mm*	σ in *mm*
Turn table 1	0.3182	0.0394	3.1854	2.2120
User 1	0.7515	0.4029	1.5350	1.9973
User 2	0.7579	0.6191	4.0662	6.6933
User 3	1.0670	0.8600	19.8502	26.9888

Table 1: Evaluation dynamic accuracy, position

Set	Speed in deg/s	σ in deg/s	Rot. error in *deg*	σ in *deg*
Turn table 1	50.3265	6.4837	2.3050	16.3334
User 1	23.2638	25.7330	0.6132	0.7942
User 2	38.7500	31.5500	6.4680	29.6140
User 3	54.3541	41.8806	11.1535	28.0446

Table 2: Evaluation dynamic accuracy, rotation

We performed three different sets of measurements. In the first set, the target was mounted on top of a turntable, about 30cm from the point of rotation. The target should have moved with constant velocity, but because of mechanical imprecisions the speed varied from $0.2m/s$ to $0.4m/s$. In the second set we had a user perform arbitrary movements with the target in the tracking volume. In this set we recorded three movements of the user, where we asked him to increase the speed of his movements with each recording. The results are shown in table 1 and 2. In the third set we used the turn table again, but this time the speed was constantly increased from $0.15m/s$ to $1.3m/s$. The results from this evaluation are shown in figure 8. As expected, the error increases with the increased speed in translation and rotation of the target.

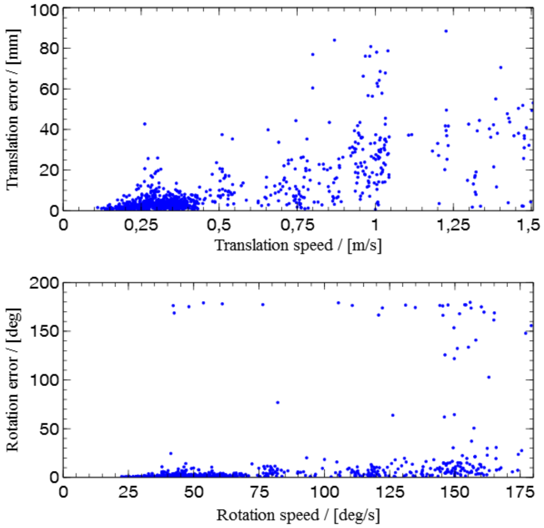

Figure 8: Dynamic error with increasing speed

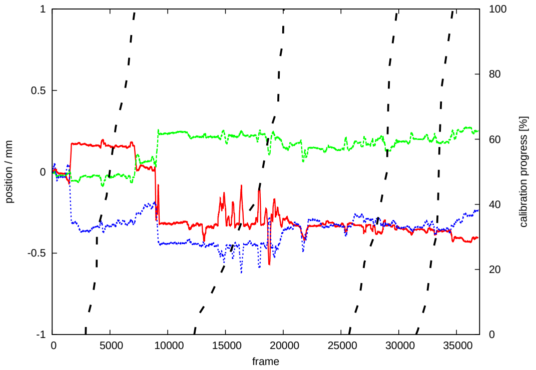

Figure 9: Position of a unmoved target during four successive recalibrations. The continuous lines are the X,Y,Z coordinate of the position and dashed lines indicate the progress of the recalibration.

6.4 Online Recalibration

In order to evaluate the online recalibration from section 5, we compared the accuracy of the recalibrated camera setup compared to the standard room calibration using our normal IR tracking cameras. A measure for the accuracy of a camera pose is the residual between re-projected points of 3D markers and observed image points. These residuals were calculated for eight different camera constellations, with one recalibrated camera in each setup. The mean residual over all setups was 0.061 ± 0.005 pixels for the recalibrated cameras. Compared to 0.059 ± 0.003 pixes for the other cameras, the accuracy is the same within statistical errors.

In a second evaluation, we measured how the pose of an unmoved 6D target changes during an online recalibration. Figure 9 shows the 3D position of a target in a four camera system. Each camera is recalibrated once and the calibration progress is also shown in figure 9. Occlusions cause the spikes in the deviations of the position. Ignoring the spikes, a shift of the position is observed whenever a camera moves (e.g near frame 1500) or finishes recalibration (e.g. near frame 7000). The shifts are caused by the changed observation situation. The changes for a moved camera (which does not contribute to the tracking any longer) or recalibrated camera (which contributes again to the tracking) are similar and below 0.3 mm in a random direction. Thus, even after several recalibration processes, the reference coordinate system shows no significant changes.

7 Applications

The usefulness of our approach is demonstrated in two scenarios. In the first scenario, we present and extension of a conventional tracking system to cover a small and occluded part of the tracking volume. The second scenario describes an outside-in tracking using mobile phones alone.

7.1 Motion Capture in Small Spaces

In order to evaluate the extension of the tracking volume in small and cluttered spaces, we set up a mock-up scenario of an assembly task inside the trunk or engine compartment of a car. For this, we used a large wooden box with a half-opened lid and filled the interior with additional obstacles (figure 10). Six mobile phones (four LGP 990, two HTC Desire) were installed inside the box and the outside tracking volume was covered by six ARTTrack3 cameras attached to the walls of the room. For the motion capture task, we attached 17 rigid targets to a person. The 6DoF tracking data of these targets was then fed into our ART-Human software, which performs skeleton calibration, inverse kinematics and simple visualization.

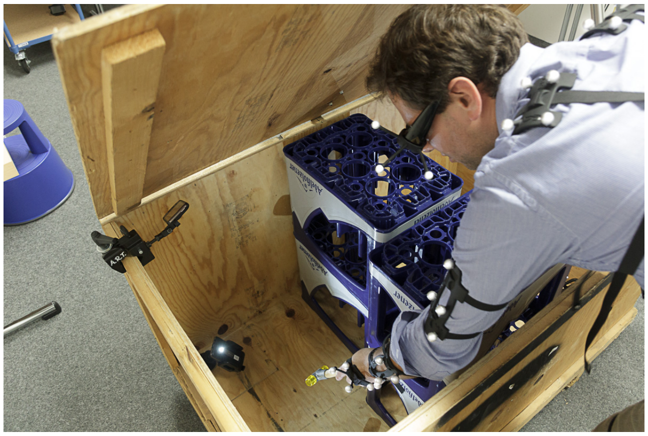

Figure 10: Motion capture of a prototypical assembly task in a box, with mobile phones installed inside. The rest of the tracking volume is covered by conventional IR tracking cameras installed in the room.

For the evaluation, we asked the person to perform a typical assembly motion at the bottom of the box. In the video accompanying this paper, it can be seen how a tracking system using only the external cameras fails to track the motion of the arms inside the box, whereas the motion capture continues when the phone cameras are added.

7.2 ISMAR Tracking Competition

Using the asynchronous tracking system, an augmented reality application was developed for the ISMAR Tracking-Competition 2011. In this competition, a room is equipped with crash-markers and several picking areas containing objects of various size. The participants are supplied with the 3D positions of the crash-markers in a reference coordinate system. During an initial setup phase, the participants use these reference positions to register their tracking system to the reference coordinate system. In the 30 minute contest run, the participants receive a file containing 3D positions of some objects in the reference coordinate system. The task is to successfully identify these objects.

Our tracking system was only using seven mobile phones (four LGP 990, three HTC Desire) and no standard tracking cameras. The participant was wearing a video-see-through HMD with a target as shown in figure 11. Since the volume of the room exceeded the the tracking volume of seven mobile phones by far, a dynamic tracking and registration process was used. The online recalibration allowed the repositioning of single cameras to move the current tracking volume through the room.

For the initial registration and in cases where all cameras had to be moved, an online registration process was developed. Using the camera of the HMD, the 2D positions of the reference crash-markers where tracked in the camera image. On the poses provided by the tracking system and the 2D positions in the images, a stereo by motion approach was used to calculate the 3D positions of the reference crash-markers within the tracking coordinate system. The positional error of the reconstructed 3D reference crash-markers was about 2cm. By aligning the reconstructed positions of the reference crash-markers with their reference positions using an absolute orientation algorithm, the registration of the tracking system was performed. Once an initial registration is available, the system is capable of detecting whenever a reference crash-marker becomes visible within the camera image and automatically starts

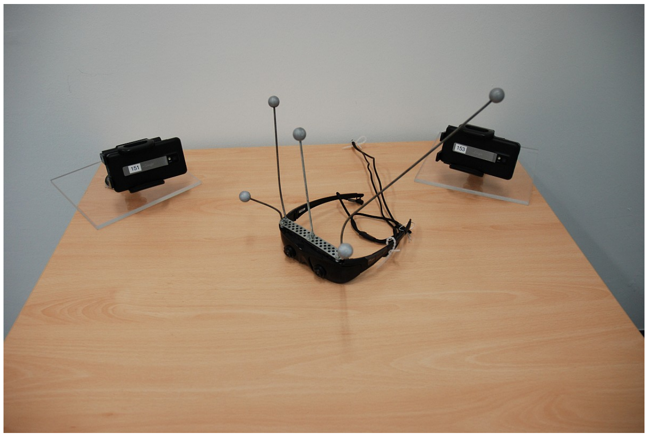

Figure 11: HMD with tree target and two mobile phone

the detection of the crash-marker. Any additional information is used to further improve 3D positions of the reference markers, thus improving the the registration of the tracking system. Since the ISMAR tracking competition was organized by authors of this paper, the participation was without rating.

8 Conclusion

In this paper we have shown that it is possible to build an optical marker-based outside-in system consisting solely of unmodified off-the-shelf mobile phones. The system delivers tracking results accurate enough for many applications. While our implementation used a separate PC for 6DoF processing, it would be possible to integrate this into the mobile phone software and distribute the whole system as an “app”.

The biggest obstacle towards making the mobile phone-based tracking system really robust is that, using the publicly available APIs, the devices automatically control the exposure time based on the observed image brightness. As the requirements of marker-based tracking (mostly black image with few white spots) contradict the normal photography requirements, the resulting images are not optimal and long exposure times with blurry images and reduced frame rate can be observed in dark environments. Despite our efforts to achieve a stable clock synchronization, the system still sometimes suffers from incorrect timestamps, leading to jitter in the calculated poses and sometimes even instability of the filter. Part of this is again caused by the lack of control over the phone cameras.

In addition to the scenarios motivated in this paper, there are other uses of asynchronous outside-in tracking algorithms. For instance, the exposure times of several IR tracking cameras could be systematically delayed to realize a tracking system that has a multiple of the tracking frequency of a single camera.

Acknowledgements

This work was supported by 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).

References

[1] S. Baker, E. P. Bennett, S. B. Kang, and R. Szeliski. Removing Rolling Shutter Wobble. In *CVPR*, pages 2392–2399, 2010.

[2] P. David, D. Dementhon, R. Duraiswami, and H. Samet. Softposit: Simultaneous pose and correspondence determination. *Int. J. Comput. Vision*, 59(3):259–284, Sept. 2004.

[3] K. Dorfmüller. An Optical Tracking System for VR/AR-Applications. In *Virtual Environments ’99, Proceedings of the Virtual Environments Conference and fifth Eurographics Workshop*, pages 33–42. Springer Verlag, 1999.

[4] M. Grewal and A. Andrews. *Kalman Filtering: Theory and Practice using MATLAB*. Wiley-IEEE Press, 3rd edition, 2008.

[5] M. Huber, M. Schlegel, and G. Klinker. Temporal Calibration in Multisensor Tracking Setups. In *8th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, Orlando, USA, October 2009.

[6] G. Klein and D. Murray. Parallel Tracking and Mapping on a Camera Phone. In *Proc. Eigth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’09)*, Orlando, October 2009.

[7] F. Madritsch and M. Gervautz. CCD-Camera Based Optical Beacon Tracking for Virtual and Augmented Reality. In *Computer Graphics Forum (Proc. Eurographics 96)*, volume 15, pages 207–216, 1996.

[8] J. D. Mulder, J. Jansen, and A. van Rhijn. An Affordable Optical Head Tracking System for Desktop VR/AR Systems. In *Proceedings of the Workshop on Virtual Environments 2003*, EGVE ’03, pages 215–223, New York, NY, USA, 2003. ACM.

[9] J. M. L. Omar Ait-Aider, Nicolas Andreff and P. Martinet. Simultaneous Object Pose and Velocity Computation Using a Single View from a Rolling Shutter Camera. In H. B. Aleš Leonardis and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3952 of *LNCS*, pages 56–68. Springer, 2006.

[10] T. Pintaric and H. Kaufmann. Affordable Infrared-Optical Pose Tracking for Virtual and Augmented Reality. In *Proceedings of IEEE VR Workshop on Trends and Issues in Tracking for Virtual Environments*, March 2007.

[11] D. Pustka, J. Willneff, O. Wenisch, P. Lükewille, K. Achatz, P. Keitler, and G. Klinker. Determining the Point of Minimum Error for 6DOF Pose Uncertainty Representation. In *Proceedings of ISMAR 2010*, October 2010.

[12] N. Rasmussen, M. Strring, T. Moeslund, and E. Granum. *Real-Time Tracking for Virtual Environments using SCAAT Kalman Filtering and Unsynchronized Cameras*, pages 333–341. Institute for Systems and Technologies of Information, Control and Communication, 2006.

[13] G. Reitmayr and T. W. Drummond. Going out: Robust Tracking for Outdoor Augmented Reality. In *Proc. ISMAR 2006*, pages 109–118, Santa Barbara, CA, USA, October 22–25 2006. IEEE and ACM, IEEE CS.

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

[15] S. Steck, R. Ehler, A. Hildebrand, L. Fritzsche, and A. Mosig. Mixed und Virtual Reality-Methoden zur Unterstützung digitaler Ergonomieabsicherungen in der frühen Produktionsplanung. In *Fachtagung Virtual Reality und Augmented Reality zum Planen, Testen und Betreiben technischer Systeme*, 2008.

[16] G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[17] G. Welch and G. Bishop. SCAAT: incremental tracking with incomplete information. In *SIGGRAPH*, pages 333–344, 1997.

[18] J. Willneff and O. Wenisch. The Calibration of Wide Angle Lens Cameras using Perspective and Non-Perspective Projections in the Context of Real-Time Tracking Systems. In *Proceedings of SPIE Optical Metrology, Videometrics, Range Imaging, and Applications XI*, May 2011.