High-Speed and Robust Monocular Tracking

Henning Tjaden[1], Ulrich Schwanecke[1], Frédéric A. Stein[2] and Elmar Schömer[2]

[1]*Computer Science Department, RheinMain University of Applied Science, Unter den Eichen 5, Wiesbaden, Germany*

[2]*Institute of Computer Science, Johannes-Gutenberg University, Staudingerweg 9, Mainz, Germany*

Keywords: Optical Tracking, Monocular Pose Estimation System, Infrared LED, Camera, High-Speed, Calibration.

Abstract: In this paper, we present a system for high-speed robust monocular tracking (HSRM-Tracking) of active markers. The proposed algorithm robustly and accurately tracks multiple markers at full framerate of current high-speed cameras. For this, we have developed a novel, nearly co-planar marker pattern that can be identified without initialization or incremental tracking. The pattern also encodes a unique ID to identify different markers. The individual markers are calibrated semi-automatically, thus no time-consuming and error-prone manual measurement is needed. Finally we show that the minimal spatial structure of the marker can be used to robustly avoid pose ambiguities even at large distances to the camera. This allows us to measure the pose of each individual marker with high accuracy in a vast area.

1 INTRODUCTION AND BACKGROUND

Tracking of moving objects is a very important aspect in many fields of application such as robotics, automotive, sports, health, or virtual reality. Often it is the basis to automatically supervise, classify and optimize motion sequences such as in athletic training or medical therapy (Fitzgerald et al., 2007; Vito et al., 2014). Other applications can be found in industrial assembly, where optical tracking can be used to assist humans or robots in order to increase their productivity and reliability or enables them to work more autonomously (Ong and Nee, 2004; Zetu et al., 2000).

Over the past decades, a large number of different solutions have been developed to track the position and orientation of objects based on various technologies such as ultrasound, magnetism, inertial, or optical sensors (Welch and Foxlin, 2002). Most widely used are probably the optical tracking systems which can be divided into active and passive systems. Passive systems use just the visible light to detect high contrast artificial fiducials or natural structures. Active systems use additional light sources to facilitate the detection of specially designed fiducials. Thereby, they usually work with infrared light to minimize the influence of ambient lighting.

Active optical tracking systems can be divided further into systems where the additional light source is attached to the camera – also called *active camera systems* – and systems with light emitting devices such as light emitting diodes (LEDs) attached to the fiducials – also called *active tracker systems*. While *active camera systems* utilize very lightweight and keen targets, *active tracker systems* usually provide a larger total working volume and higher precision.

Optical tracking systems often determine the three-dimensional position of a ball-like fiducial based on two or more camera images and triangulation. While this can be done very efficiently utilizing epipolar constraints, it requires the fiducials always to be seen by (at least) two cameras. In some applications, this may not always be guaranteed and a system based on the image of just a single monocular camera is needed. In this work we therefore focus on tracking with a single monocular optical camera.

In its simplest form, optical tracking of the pose of a rigid object with a single camera is based on passive planar markers (Olson, 2011; Herout et al., 2013). The advantage of these passive systems is their low complexity. Users can simply print or even draw their own markers. But, even though the latest methods achieve higher robustness against fast movement by tracking arbitrary patterns based on feature descriptors (Wagner et al., 2008; Ozuysal et al., 2010), or using a parametrized model of the object under investigation (Schmaltz et al., 2012; Prisacariu and Reid, 2012), passive systems are still limited in distance range and stability under low lighting conditions or lack in terms of precision or high-speed performance.

Tjaden H., Schwanecke U., Stein F. and Schömer E..
High-Speed and Robust Monocular Tracking.
DOI: 10.5220/0005267104620471
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 462-471
ISBN: 978-989-758-091-8

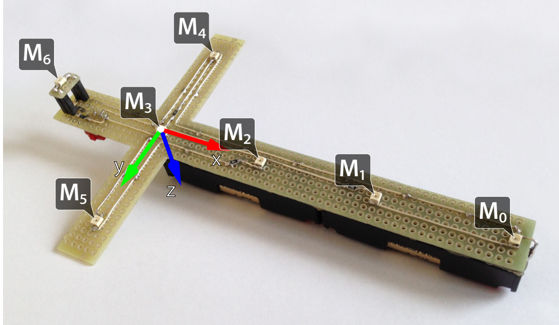


Figure 1: The proposed markers consist of seven LEDs arranged in a cross-shaped pattern, where $\mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_5$ define the right handed marker coordinate frame.

Active tracking systems overcome most of the previously mentioned limitations as they are less dependent on the given lighting conditions. One of the main tasks of active systems is the identification of the individual lights or retro-reflective tags each marker is composed of. (Naimark and Foxlin, 2005) e.g. presented a technique for encoding individual LED markers by amplitude modulation, used in an initialization step. After initialization the LEDs are tracked incrementally from frame to frame. Such techniques come with the downside of potential tracking losses, after which the system would have to get re-initialized.

Monocular tracking of purely co-planar markers suffers from pose ambiguities, while markers with spatial structure are likely to occlude themselves. (Faessler et al., 2014) recently presented a system, that can track a single marker composed of four or five LEDs at 90 Hz. Although these LEDs can be arranged arbitrarily, they should span a volume as large as possible to avoid pose ambiguities. The resulting self-occlusions are dealt with a time consuming combinatorial brute force approach, that is used to initialize and re-initialize the LED identification. Once initialized the LEDs are tracked incrementally in conjunction with motion prediction. The spatial positions of the marker LEDs are calibrated with a commercial multi-camera motion capturing system.

In this paper we present a high-speed and robust monocular (HSRM) tracking system that is superior to all other present systems regarding accuracy, tracking range and robustness, while still being low-cost on the hardware side. The system can estimate the pose of multiple markers, each of them composed of seven infrared LEDs, at frequencies far over 500 Hz, allowing robust tracking even of fast-moving objects. Our three main contributions are: First, a method to identify each individual LED of a marker solely based on 2D geometrical constraints not requiring any initialization or frame-to-frame tracking. Second, an empirical proof that the minimal 3D structure of our nearly co-planar marker is an optimal tradeoff between the occurrence of self-occlusions and the avoidance of pose ambiguities. Third, a semi-automatic marker calibration algorithm avoiding time-consuming and error-prone manual measurements and ensuring accurate tracking results.

2 MONOCULAR MARKER TRACKING

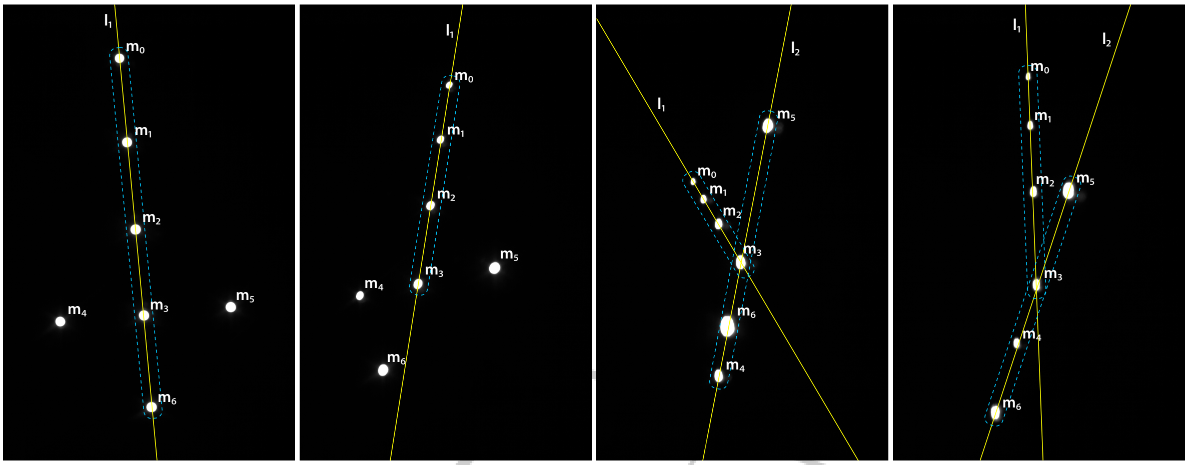
In the following we describe our marker pattern and its identification as well as the tracking of multiple markers in detail. We also show that our approach is robust to measurement noise and pose ambiguities and present an algorithm to semi-automatically calibrate our markers with high accuracy.

2.1 Preliminaries

The proposed markers consist of seven infrared LEDs connected to a small battery pack (Figure 1). For our prototype we chose SMD (surface-mounted device) LEDs which are perceptible from large viewing angles up to nearly $\pi/2$. We tested our system with a variety of monochrome USB 3.0 high-speed cameras, operating from 90 Hz to 500 Hz with resolutions up to 2048×2048 pixels. The cameras have pre-calibrated intrinsics and are equipped with an infrared filter to reduce the influence of ambient light.

The marker LEDs are arranged in a constrained geometrical *cross-shaped* pattern. The spatial LED positions are denoted relative to the marker coordinate frame and are referred to as $\mathbf{M}_i = (x_i, y_i, z_i)^\top, i = 0, \ldots, 6$. Their corresponding 2D projections into the camera image are denoted by $\mathbf{m}_i = (x_i, y_i)^\top, i = 0, \ldots, 6$. LED $\mathbf{M}_3 = (0,0,0)^\top$ defines the origin of the right handed marker coordinate frame, while the vectors $\overrightarrow{\mathbf{M}_3\mathbf{M}_2}$ and $\overrightarrow{\mathbf{M}_3\mathbf{M}_5}$ define the corresponding *x*- and *y*-axis, respectively. $\mathbf{M}_0, \ldots, \mathbf{M}_5$ lie in the *xy*-plane and $\mathbf{M}_6$ is slightly elevated for stabilization purposes as will be explained in section 2.2.3. In section 3 we show, that the exact spatial positions of all LEDs can be measured automatically (up to a scaling factor) in a pre-calibration step.

We refer to the pose of a marker with respect to the camera coordinate frame as $P_M = [R_M|\mathbf{t}_M]$, with $R_M \in \mathbb{SO}(3)$ describing the rotation of the marker and $\mathbf{t}_M \in \mathbb{R}^3$ being the location of the origin of the marker coordinate frame in the camera coordinate frame. Perspective projection of a 3D point $\mathbf{M}$ including dehomogenisation is denoted by $\pi(\mathbf{M}) = (x/z, y/z)^\top$.



(a) Case L5. (b) Case L41. (c) Case L42a. (d) Case L42b.

Figure 2: Raw input video frames of a marker under various perspective transformations showing the different configurations distinguished during correspondence assignment. The dominant lines $\mathbf{l}_1$ and $\mathbf{l}_2$ are drawn and their participants are framed by a dashed line. (a) An example where five LED projections lie on $\mathbf{l}_1$. (b) An example where $\mathbf{l}_1$ has four participants. (c) An example where two dominant lines appear in the projection pattern. In this case $\mathbf{m}_6$ lies between $\mathbf{m}_4$ and $\mathbf{m}_5$ on $\mathbf{l}_2$. (d) The second case where two dominant lines appear. In this case $\mathbf{m}_6$ lies outside of $\mathbf{m}_4$ and $\mathbf{m}_5$ on $\mathbf{l}_2$.

2.2 Tracking Method

The pose of each marker is determined based on the known 3D positions $\mathbf{M}_i$ of the seven LEDs and their corresponding 2D projections $\mathbf{m}_i$. Therefore, in a first step we segment the projected LEDs in each video frame and determine their exact 2D positions as explained in detail in section 2.2.1. After the 2D projections are segmented they have to be assigned to the corresponding LEDs. Since all LED projections look alike they cannot be distinguished by simple local analysis. In section 2.2.2 we show how to determine the correct correspondences, while section 2.2.3 explains how to estimation the pose of the markers. Finally, in section 2.2.4 the extension to multi-marker tracking is presented.

2.2.1 LED Segmentation and 2D Localization

The LEDs will appear as bright blobs in the video frames. These regions are saturated to a great extent and therefore are significantly brighter than the rest of the scene. To reduce the influence of ambient light, we utilize an infrared filter, so that almost only the LEDs remain visible in the camera image. To segment the blobs, we use a simple binary thresholding, keeping only pixels with an intensity above a prescribed threshold s. The remaining pixels are grouped into connected regions Ω_i using a union-find algorithm.

In a next step we determine the 2D positions

$$\mathbf{m}'_i = \left(\frac{m_{10i}}{m_{00i}}, \frac{m_{01i}}{m_{00i}}\right)^\top \qquad (1)$$

of the LED projections by calculating the intensity centroid of each region Ω_i based on the moments

$$m_{pqi} = \sum_{(x,y)\in\Omega_i} x^p y^q \mathbf{I}(x,y)^2. \qquad (2)$$

Thereby, we used a quadratic weighting of the pixel intensities $\mathbf{I}(x,y)$ to reduce the influence of border pixels, that tend to flicker due to their lower intensities compared to pixels near the centroid. Experiments showed that (2) reduces jitter of the measurements similarly to applying a low-pass filter (e.g. Gaussian), while being computationally less expensive.

In a last step all determined LED projections $\mathbf{m}'_i$ are normalized and undistorted to ideal image coordinates $\mathbf{m}_i$ using the pre-calibrated camera intrinsics.

2.2.2 Determining the 2D/3D Correspondences

Our constrained *cross-shaped* marker pattern allows to determine the correspondences between 3D LED position $\mathbf{M}_i$ and ideal image coordinate $\mathbf{m}_i$ for each individual frame. Thereby, the only restriction is that all seven LEDs of a marker have to be visible in the respective frame.

Because $\mathbf{m}_0, \ldots, \mathbf{m}_3$ always must lie on a straight line, we start by investigating the dominant lines in the set of projections $\mathcal{M} = \{\mathbf{m}_0, \ldots, \mathbf{m}_6\}$. The first dominant line $\mathbf{l}_1$ can either have four participants (case L4, see (Fig. 2(b)), five participants (case L5, see Fig. 2(a)) or more than five participants. The latter case only occurs under very shallow viewing angles with low measuring accuracy and will therefore not be further considered.

Due to the structure of the marker there exist perspective transformations where $\mathbf{m}_6$ together with $\mathbf{m}_3$, $\mathbf{m}_4$ and $\mathbf{m}_5$ forms a second dominant line $\mathbf{l}_2$ (see Figs. 2(c) and 2(d)). Thus, case L4 is further separated into L41 (one line) and case L42 (two lines). These two cases can be differentiated based on regression lines to all thirty-five subsets of four points from $\mathcal{M}$. Therefore, the lines and their corresponding points are grouped by quality and angle, where quality is determined by the distance of the furthest outlier to the line. The cases L5, L41 and L42 now can be distinguished as follows, based on simple 2D geometric criteria:

L5: The two points that do not belong to $\mathbf{l}_1$ define a second line $\mathbf{l}_2$. The closest point to the intersection of $\mathbf{l}_1$ and $\mathbf{l}_2$ must be $\mathbf{m}_3$. The point that lies solely on one side of $\mathbf{l}_2$ is $\mathbf{m}_6$. The points $\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2$ are assigned descendingly according to their absolute distance to $\mathbf{l}_2$. Finally $\mathbf{m}_4$ and $\mathbf{m}_5$ are assigned based on their signed distance to the line $\mathbf{l}_{03}$ given by $\mathbf{m}_0, \mathbf{m}_3$.

L41: The point belonging to $\mathbf{l}_1$ furthest from the centroid of the three points not lying on $\mathbf{l}_1$ must be $\mathbf{m}_0$. Next, $\mathbf{m}_1, \mathbf{m}_2$ and $\mathbf{m}_3$ can be assigned ascendingly according to their distance to $\mathbf{m}_0$. One of the three remaining LEDs (either $\mathbf{m}_4$ or $\mathbf{m}_5$) will lie solely on one side of $\mathbf{l}_{03}$. It can be detected and assigned by calculating the three signed distances to $\mathbf{l}_{03}$. Assuming it is $\mathbf{m}_5$, we construct line $\mathbf{l}_{35}$ given by $\mathbf{m}_3, \mathbf{m}_5$ and assign $\mathbf{m}_4$ and $\mathbf{m}_6$ ascendingly to their distance to it.

L42: We first identify $\mathbf{l}_1$ by the fact that all points incident to $\mathbf{l}_1$ lie on the same side of line $\mathbf{l}_2$, except $\mathbf{m}_3$ which is the intersection of $\mathbf{l}_1$ and $\mathbf{l}_2$. Next, we can enumerate $\mathbf{m}_0$, $\mathbf{m}_1$ and $\mathbf{m}_2$ in descending order according to their distance to $\mathbf{m}_3$. Now the LED that lies solely on one side of $\mathbf{l}_1$ is either $\mathbf{m}_4$ or $\mathbf{m}_5$. Thereby the correct assignment is determined by the signed distance to $\mathbf{l}_1$ as in L41. Assuming that this assignment was $\mathbf{m}_5$, there are two possibilities (L42a and L42b) to assign $\mathbf{m}_6$ and $\mathbf{m}_4$, which can not be robustly distinguished by 2D criteria (see Figs. 2(c) and 2(d)). We therefore solve this ambiguity with regard to the 3D structure of the marker. This is done by calculating the two homographies H_{46} and H_{64} for both possible assignments from all six coplanar LEDs. These homographies allow us to calculate two pose estimations (Xu et al., 2009) called P_{H46} and P_{H64}. To determine the correct assignment, we calculate the average reprojection errors (see eqn. (3) in section 2.2.3) for both pose estimations using all seven LEDs and choose the assignment combination that yields the smaller error.

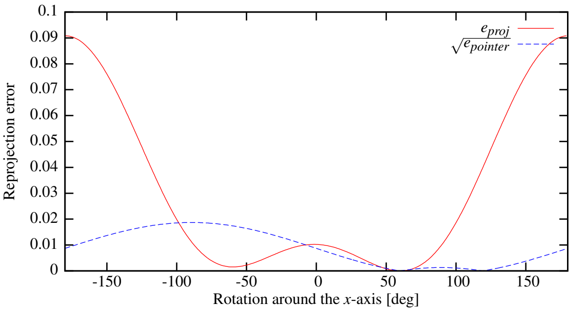


Figure 3: A synthetic experiment, demonstrating the effectiveness of our pointer LED. We rotated the model points around the x-axis and measure the reprojection errors e_{proj} and $e_{pointer}$ to a chosen *ground truth* angle $\alpha = 60°$ with $\mathbf{t}_M = (0,0,10)^\top$ in the image plane. Our model consists of five points: $\mathbf{M}_0 = (1,1,0)^\top$, $\mathbf{M}_1 = (1,-1,0)^\top$, $\mathbf{M}_2 = (-1,-1,0)^\top$, $\mathbf{M}_3 = (-1,1,0)^\top$, $\mathbf{M}_{pointer} = (0,0,-0.1)^\top$. For further explanation please refer to (Schweighofer and Pinz, 2006). We plot $\sqrt{e_{pointer}}$ for visualizing purposes.

2.2.3 Pose Estimation

Once all seven correspondences are known, the pose $P_M = [R_M|\mathbf{t}_M]$ of the marker in the associated frame can be determined. Thereby a first estimate P_H is calculated based on the homography H given by the six coplanar correspondences $(\mathbf{M}_i, \mathbf{m}_i), i = 0, \ldots, 5$ (Xu et al., 2009). Next, P_H is refined by iteratively minimizing the reprojection error in the image space

$$e_{proj}(P,n) = \sum_{i=0}^{n} \|\mathbf{m}_i - \pi(R\mathbf{M}_i + \mathbf{t})\|^2. \qquad (3)$$

Using a Levenberg-Marquardt solver (Levenberg, 1944; Marquardt, 1963) this results in the pose $P_1 = \arg\min e_{proj}(P,6)$. There exist ambiguities when estimating the pose from coplanar markers, which cause the pose to flip at large distances (Schweighofer and Pinz, 2006). This is because in general the reprojection error $e_{proj}(P,n)$ has two distinct local minima (see Fig. 3). Following (Schweighofer and Pinz, 2006) we take P_1 and calculate the corresponding second pose P_2 belonging to the other minimum of $e_{proj}(P,5)$ using only the six coplanar correspondences. Schweighofer and Pinz then choose the correct pose by comparing the reprojection errors for both solutions, assuming that the desired pose will yield a smaller error. Unfortunately, in practice this approach only yields the correct solution with a probability of about 0.75 to 0.95, depending on the angle between the optical axis and the z-axis of the marker and its distance to the camera. Inspired by (Yang et al., 2012) we choose the correct solution by utilizing the non coplanar LED $\mathbf{M}_6$ as a pointer to the correct pose. Thereby, the reprojection error of the non coplanar point $\mathbf{M}_{pointer}$ projected on the image point $\mathbf{m}_{pointer}$ yields a second error function

$$e_{pointer}(P) = \|\mathbf{m}_{pointer} - \pi(R\mathbf{M}_{pointer} + \mathbf{t})\|^2 \quad (4)$$

that is robust to measurement noise and suitable to distinguish the two poses (see Fig. 3). Synthetic experiments showed that $e_{pointer}(P)$ and $e_{proj}(P)$ share the sought minimum while being contrary at the delusive minimum of $e_{proj}(P)$. Thus, a robust pose estimation is given by

$$P_M = \underset{P\in\{P_1,P_2\}}{\arg\min}\ e_{pointer}(P). \quad (5)$$

Refining the pose P_M using all seven correspondences results in the final pose estimation $P_M = \arg\min\ e_{proj}(P,6)$.

2.2.4 Multi-marker Tracking

Our tracking algorithm can easily be extended to multi-marker tracking. The image processing steps are exactly the same as described in section 2.2.1 until the centeroids of all LED projections are determined. We then use the k-means algorithm (MacKay, 2002) to cluster the projections with an adapted initialization scheme. Thereby we set $k = \lceil n/7 \rceil$, where n is the number of visible LEDs in the respective frame. So, false positive detections e.g. caused by reflections of sunlight or markers that are not fully visible, in many cases will end up in a separate smaller cluster and can easily be filtered. In order to speed up and assure correct convergence of k-means, we initialize the cluster centers $\mathbf{b}_i$ by exploiting the constraint that each marker cluster must have exactly seven members. This is done by repeating the following procedure k times, i.e. for $i = 0,\ldots,k-1$ do

1. Randomly choose a point $\mathbf{m}_i$ from all unassigned LED projections.
2. Assign the six (or less, if not more are unassigned) closest to $\mathbf{m}_i$ LED projections to it.
3. Set the location of the current center $\mathbf{b}_i$ to the barycenter of $\mathbf{m}_i$ and its assigned LED projections.

After this initialization k-means usually only needs one or two iterations for convergence.

For each cluster (marker) the 3D-2D correspondences can be determined independently as described in section 2.2.2. We found that, although k-means is a rather simple algorithmic choice for separating the markers and likely to fail if the projections of the markers are too close together, it still works fine for many application scenarios. For example when tracking a human arm and there are two markers attached to the upper and the lower arm, critical configurations are unlikely to occur.

The identification of each individual marker (determined by the clusters) is based on the cross-ratio

$$CR(\mathbf{M}_0,\mathbf{M}_3;\mathbf{M}_2,\mathbf{M}_1) = \frac{\|\mathbf{M}_0-\mathbf{M}_2\|\cdot\|\mathbf{M}_3-\mathbf{M}_1\|}{\|\mathbf{M}_3-\mathbf{M}_2\|\cdot\|\mathbf{M}_0-\mathbf{M}_1\|} \quad (6)$$

of the collinear LEDs $\mathbf{M}_0,\ldots,\mathbf{M}_3$. By varying the positions of $\mathbf{M}_1$, $\mathbf{M}_2$ along the line given by $\mathbf{M}_0$, $\mathbf{M}_3$ new unique marker IDs can be constructed. All markers to track have to be calibrated beforehand, so that the correspondences between LED projection clusters and markers can be found by simply comparing the cross-ratio $CR(\mathbf{m}_0,\mathbf{m}_3;\mathbf{m}_2,\mathbf{m}_1)$ of each cluster with the cross-ratio of each marker.

3 MARKER CALIBRATION

Measuring the positions of the LEDs of a marker manually is error-prone, inconvenient and should therefore be reduced to a minimum. Automatic measuring of the spatial LED positions is convenient, achieves maximum tracking accuracy and can be done by only measuring the distance d_{03} between $\mathbf{M}_0$ and $\mathbf{M}_3$ manually as shown next. Thereby, d_{03} determines the overall scale of the corresponding marker.

Our calibration algorithm is based on a sequence of $n+1$ image frames I_j, $\quad j = 0,\ldots,n$ recorded while translating and rotating the marker arbitrarily in front of the camera at a rather short distance. For our experiments we record 500 frames within a period of 5 seconds, i.e. one frame each 10ms. For each individual frame I_j we then extract the set of normalized and assigned LED projections $\mathcal{M}_j$. Since the 3D structure of the marker is not yet known at this point, perspectives yielding case L42 (section 2.2.2) cannot be assigned correctly. These usually rarely occurring frames are omitted automatically. The complete calibration scheme is split into an initialization and a refinement step, as explained in the following.

3.1 Initialization

As illustrated in algorithm 3.1 we first select a coarse subset $\mathbb{M}$ containing only every o-th projection set $\mathcal{M}_j$. In our experiments we use a subsample offset $o = 50$, i.e. the respective frames were recorded with an offset of 500 ms which promotes wide baselines. In the next step we select all pairs $(\mathcal{M}_k,\mathcal{M}_{k'})_l \in \mathbb{M}\times\mathbb{M}, k < k'$ and calculate the relative camera pose $P_{C,l}$ from $\mathcal{M}_k$ to $\mathcal{M}_{k'}$ using the five-point algorithm (Nister, 2004). For each pair we then determine a set of spatial positions $\mathbf{M}_{i,l}$ via linear triangulation relative to $P_{C,l}$. To merge these sets of position estimates, they need to be transformed into a common marker coordinate system. This is done by applying a similarity transformation T_l that enforces

Algorithm 3.1: Marker Calibration.

```
Data: $d_{03}$, $\mathcal{M}_j$, $j = 0,\dots,n$, subsample offset $o$
Result: $\mathbf{M}_i$, $i = 0,\dots,6$
/* Initialization */
$\mathbb{M} \leftarrow \{\mathcal{M}_{j\cdot o} \mid j = 0,\dots,{}^{n}/_{o}\}$;
foreach $(\mathcal{M}_k, \mathcal{M}_{k'})_l \in \mathbb{M} \times \mathbb{M}, k < k'$ do
    $P_{C,l} \leftarrow$ rel. camera pose from $\mathcal{M}_k$ to $\mathcal{M}_{k'}$;
    Triangulate $\mathbf{M}_{i,l}$ using $P_{C,l}$, $i = 0,\dots,6$;
    $\mathbf{M}_{i,l} \leftarrow T_l \mathbf{M}_{i,l}$ with sim. $T_l$, $i = 0,\dots,6$;
end
$\mathbf{M}_i \leftarrow \left(x_i^{med}, y_i^{med}, z_i^{med}\right)$, $i = 0,\dots,6$;
/* Refinement */
$\mathcal{R} \leftarrow \{\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_4, \mathbf{M}_5, \mathbf{M}_6\}$;
repeat
    foreach projection set $\mathcal{M}_j$ do
        Calculate $P_{M,j}$;    // section 2.2.3
        $P_{C,j} \leftarrow [R_{M,j}^{\top} | -R_{M,j}^{\top} \mathbf{t}_{M,j}]$;
        foreach $\mathbf{m}_{i,j} \in \mathcal{M}_j \setminus \{\mathbf{m}_{0,j}, \mathbf{m}_{3,j}\}$ do
            Fill system $A_i \mathbf{x}_i = \mathbf{b}_i$ using $P_{C,j}$;
        end
    end
    $e \leftarrow 0$;
    foreach $\mathbf{M}_i \in \mathcal{R}$ do
        Solve $A_i \mathbf{x}_i = \mathbf{b}_i$;
        $e \leftarrow e + \|\mathbf{M}_i - \mathbf{x}_i\|$;
        $\mathbf{M}_i \leftarrow \mathbf{x}_i$;
    end
until $\Delta e < \varepsilon$;
```

$$T_l \mathbf{M}_{3,l} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \; T_l \mathbf{M}_{0,l} = \begin{pmatrix} d_{03} \\ 0 \\ 0 \end{pmatrix}, \; \frac{T_l \mathbf{M}_{5,l}}{\|T_l \mathbf{M}_{5,l}\|} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \quad (7)$$

Once transformed into the marker coordinate system, we calculate the median values x_i^{med}, y_i^{med} and z_i^{med} of all $\mathbf{M}_{i,l}$ for each dimension independently. The final position estimates of the initialization are then given by the points $\mathbf{M}_i = \left(x_i^{med}, y_i^{med}, z_i^{med}\right)$.

3.2 Refinement

In order the increase the accuracy and reliability of our tracking system the spatial position estimates coming from the initialization step can be further improved by a subsequent refinement step. Experiments showed that the standard deviation of repeated initial calibration increases with the distance between marker and camera. In order to overcome this problem and ensure exact calibration results, we apply an iterative refinement step that robustly converges to the desired result.

The basic idea of the refinement step (see Algorithm 3.1) is to interleave and decouple the refinement of the camera poses and the refinement of the spatial LED positions, similar to (Lakemond et al., 2013). During refinement, each camera and each LED position is treated independently. We use all of the previously recorded projection sets $\mathcal{M}_j, j = 0,\dots,n$ for the refinement step. The overall scaling is preserved by using $\mathbf{M}_3 = (0,0,0)^{\top}$ and $\mathbf{M}_0 = (d_{03},0,0)^{\top}$ as fixed points that remain unaffected.

In each iteration we first calculate all marker poses $P_{M,j}$ with the method described in section 2.2.3 using every $\mathcal{M}_j$ and the current $\mathbf{M}_i$. Thereby the corresponding camera poses can be derived as $P_{C,j} = [R_{M,j}^{\top} | -R_{M,j}^{\top} \mathbf{t}_{M,j}] = [R_{C,j} | \mathbf{c}_j]$. Afterwards, all these updated camera poses are used to re-triangulate the LEDs. We therefore determine the intersection points $\mathbf{x}_i$ of the projection rays from each camera center $\mathbf{c}_j$ through each $\mathbf{m}_{i,j}$ such that

$$(\mathbf{c}_j - \mathbf{x}_i) \times \mathbf{v}_{i,j} = 0 \quad (8)$$

with $\mathbf{v}_{i,j} = R_{C,j}[\mathbf{m}_{i,j}^{\top}, 1]^{\top}$. For each LED this can be formulated as a least squares problem. Thereby, based on eqn. (8) a linear system $A_i \mathbf{x}_i = \mathbf{b}_i$ has to be solved. The overall error per iteration is given by $e = \sum \|\mathbf{M}_i - \mathbf{x}_i\|$. The algorithm stops if the change Δe of the error is smaller than a prescribed bound ε. In our experiments we set $\varepsilon = 0.0001$.

4 EVALUATION

In this section we demonstrate the capabilities and limitations of *HSRM*-Tracking. We start by presenting a brief runtime performance analysis followed by a detailed discussion of the reliability and repeatability of the automatic calibration method as well as the measurement accuracy of our pose estimation method. We conclude our evaluation by demonstrating that the robustness to pose ambiguities and rapid motion of our method outperforms any other present monocular tracking system.

4.1 Performance Analysis

For our experiments we used a two megapixel USB 3.0 camera[1] with a fixed-focus 9 mm lens and the prototype marker seen in Figure 1. The quantity d_{03} = 114.2 mm, i.e the scale of the marker was measured manually using a caliper. The automatically

[1] XIMEA xiQ MQ022MG-CM. See: www.ximea.com

calibrated spatial positions of its LEDs are shown in Table 1. The calibration thus yields a cross-ratio of $CR(\mathbf{M}_0,\mathbf{M}_3;\mathbf{M}_2,\mathbf{M}_1) \approx 3.99$ for the marker.

Table 1: Spatial LED positions and standard deviations of the calibration process for our prototype marker.

[mm]	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_4$	$\mathbf{M}_5$	$\mathbf{M}_6$
x	$75.91 \pm .02$	$37.91 \pm .01$	$-0.14 \pm .05$	$0.28 \pm .05$	$-38.29 \pm .03$
y	$-0.12 \pm .01$	$-0.04 \pm .01$	$-37.97 \pm .07$	$38.15 \pm .08$	$0.4 \pm .01$
z	$0.14 \pm .01$	$0.24 \pm .01$	$0.33 \pm .03$	$-0.03 \pm .02$	$-11.21 \pm .03$

We tested a C++ implementation of our system on a commodity quad core CPU @ 2.6 GHz. Thereby the whole tracking process is performed in a single thread. Figure 4 shows a plot of the computation times subdivided into the combined 2D image processing steps and the marker pose estimation including correspondence assignment. With an average computation time of about 1 ms per frame we can easily track the pose at full framerate (170 fps) of our camera even with multiple markers. This enables tracking a single marker at frequencies of more than 1000 Hz given a suitable camera. Each additional marker would only increase the runtime by approximately 0.6 ms if not processed in parallel.

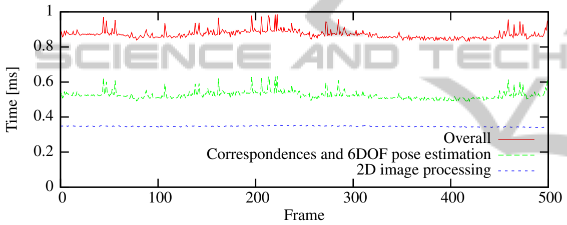


Figure 4: Computation times of the optical tracking method. All measurements were averaged over 100 runs using a pre-recorded sequence of 500 frames at full 2048×1088 resolution. In this sequence a single marker was held in hand and translated and rotated arbitrarily in front of the camera.

Table 2: Timings of the image processing step.

[px]	640 × 480	1280 × 1024	2048 × 1088	2048 × 2048
[ms]	≈ 0.1	≈ 0.3	≈ 0.4	≈ 0.9

The image processing mostly depends on camera resolution (see Table 2) and is nearly independent of the number of visible markers. Our implementation is optimized using SSE2 instructions, especially speeding up the LED region grouping step.

4.2 Calibration Reliability

The reliability of our marker calibration method can be determined by its repeatability when calibrating a marker multiple times. Since the spatial positions of the LEDs remain the same for each calibration, the standard deviation of the calibrated spatial positions should ideally be zero.

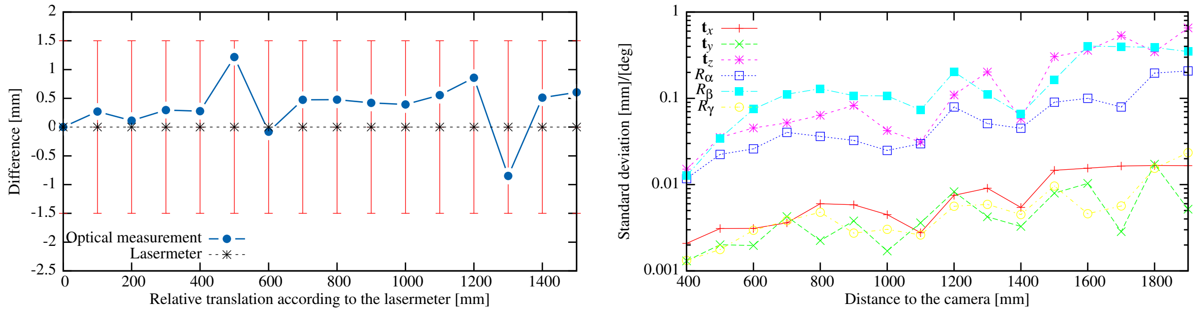
In practice the spatial positions after the initialization are already accurate to several tenths of a millimeter. For experimental evaluation of our refinement method we have compared it to an established implementation of full sparse bundle adjustment (SBA) (Lourakis and Argyros, 2009). We can show that our method reduces the standard deviation of the calibration initialization about an order of magnitude while SBA only reduces it by a factor of 2 to 3. To further analyze this difference, we have conducted experiments where we randomly distorted the initial estimates of $\mathbf{M}_i$ by ±5 mm for each calibration. Our refinement method was still able to robustly reduce the standard deviation to several hundredths of a millimeter while SBA varies about an order of magnitude more. This shows the reliability of our refinement method even for poor initial estimations.

4.3 Measurement Accuracy

We first analyze the translational accuracy of our pose estimation. Thereby, we are particularly interested in the depth accuracy since this is the most critical part in monocular tracking. We fixed the camera and the marker on a straight rail of 2 meters length and set up a lasermeter[2] next to the camera casting its beam onto the marker. The camera was fixed at one end of the rail and the marker was oriented parallel to the image plane. Starting at 400 mm distance we moved the marker in 100 mm steps according to the lasermeter towards the other end of the rail. At each position p we recorded 500 optical measurements and calculated the average marker translation vector $\bar{\mathbf{t}}_{M,p}$ as the mean of all measurements. Afterwards, we calculated the measured relative translation as $d_M = \|\bar{\mathbf{t}}_{M,p} - \bar{\mathbf{t}}_{M,0}\|_2$ for every position and compared it to the lasermeter as shown in Figure 5(a). We also analyzed the standard deviations of the translation parameters $\mathbf{t}_x$, $\mathbf{t}_y$ and $\mathbf{t}_z$ and the orientation parameters R_α, R_β and R_γ (the Euler angles around the x-, y- and z-axis) across the 500 samples. Their growth in relation to the distance to the camera is shown in Figure 5(b).

The results show that our system is able to measure the position of the marker with an accuracy of about ±1 mm even at a distance of almost 2 meters. Unfortunately the lasermeter we used only can measure with an accuracy of ±1.5 mm. Note that the

[2] *Precaster Enterprises* HANS CA770 laser meter. See: http://www.precaster.com.tw



(a) Comparison between the lasermeter and the optical measurement.

(b) Standard deviations of the measured 6DOF of the marker.

Figure 5: Results of our relative position accuracy measurements from about 400 mm to about 1900 mm distance to the camera/lasermeter. (a) The plot shows the difference from the optical measurements to the lasermeter. The error bars are due to the lasermeter. (b) Standard deviations of marker pose parameters in relation to the distance to the camera.

depth accuracy is strongly dependent on the quality of the intrinsic camera calibration.

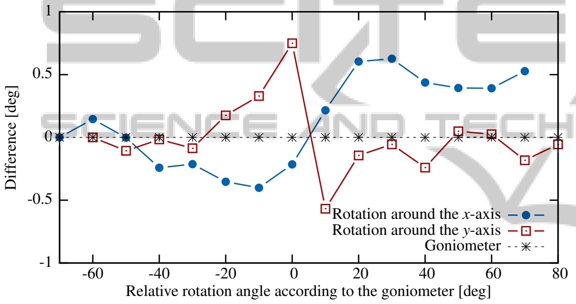


Figure 6: Results of our relative rotation accuracy measurements. The plot shows the difference from the optical measurements to a goniometer.

Next we analyze the rotational accuracy of the pose estimation. For this, the marker was fixed at a distance of 1 meter to the camera and attached to a goniometer. We rotated the marker around its x-axis from $-70°$ up to $70°$ and around the y-axis from $-60°$ up to $80°$ in $10°$ steps. These angle boundaries are due to self occlusions of the LED pattern that occur in our particular experimental setup.

We again recorded 500 measurements at each angle α and compared the relative angle differences. For this, we converted the marker rotation matrix $R_{M,\alpha}$ into the corresponding unit quaternion $\mathbf{q}_{M,\alpha}$ and calculated the average $\bar{\mathbf{q}}_{M,\alpha}$. We then determined the measured relative rotation as the angle of $\bar{\mathbf{q}}_{M,\alpha}\bar{\mathbf{q}}_{M,0}^{-1}$ for every following orientation and compared it to the goniometer as shown in Figure 6. Our results show that the rotational measurement error is below $\pm 1°$.

4.4 Robustness to Pose Ambiguities

In this experiment we demonstrate the robustness of the proposed method to pose ambiguities at large distances to the camera that monocular tracking systems usually suffer from. For this we fixed the camera on a static tripod and attached the marker to a wheeled tripod. Starting at about 50 cm distance to the camera we recorded a sequence of 2290 frames while manually moving the marker away up to 7.5 m. Based on this pre-recorded sequence we monitored the number of pose flips (Figure 7) using our proposed method and compared it to the following three other methods:

Method 1: is the simplest approach estimating the pose by minimizing the reprojection error of only the six coplanar correspondences i.e.

$$P_M = P_1 = \arg\min e_{proj}(P, 5) \tag{9}$$

without considering other solutions.

Method 2: also only considers the six coplanar correspondences but makes use of the improvement of (Schweighofer and Pinz, 2006) by choosing

$$P_M = \underset{P \in \{P_1, P_2\}}{\arg\min} \; e_{proj}(P, 5), \tag{10}$$

based on the cumulated projection error of all six correspondences for both solutions.

Method 3: is similar to method 1 but uses all 7 (non-coplanar) correspondences to determine

$$P_M = P_1 = \arg\min e_{proj}(P, 6), \tag{11}$$

considering the 3D structure of the marker.

All methods are based on P_H, the pose roughly estimated from homography.

The results show that method 1 performs worst with the most pose flip occurrences in total, starting at about 2 meters distance to the camera. Method 2 demonstrates how the strategy of Schweighofer and Prinz reduces the number of pose flips when using a purely coplanar marker. Results show that in this case the flips first occur at a distance of about 3 meters to the camera and that their frequency rises with increasing distance. Although method 3 performs better than method 2 regarding the total number of flips,

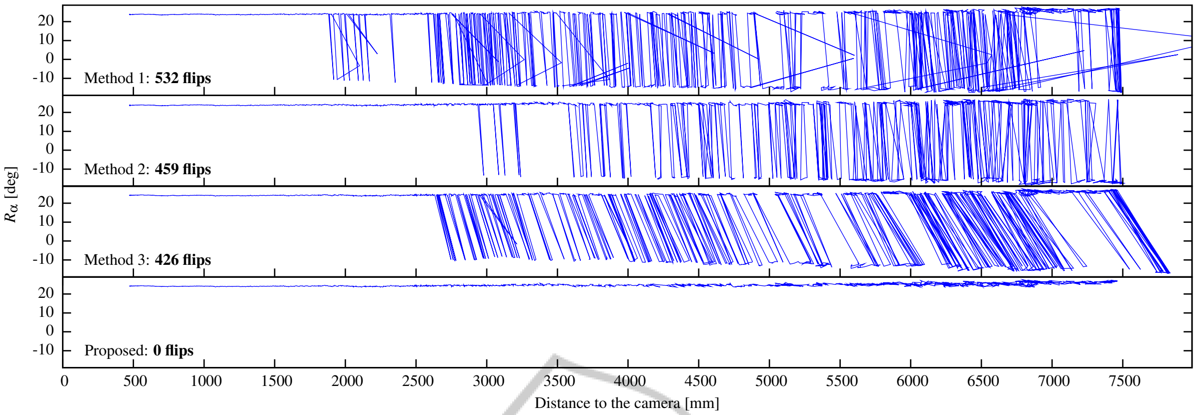
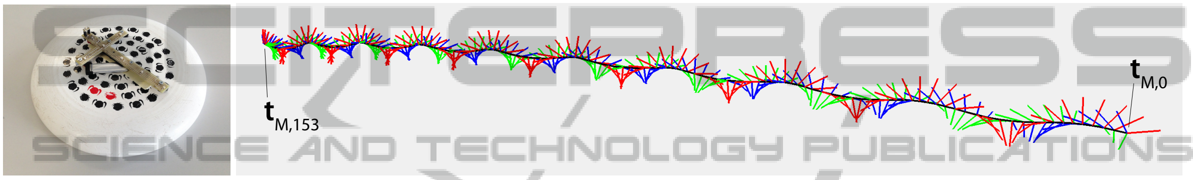


Figure 7: Results of our pose flip comparison experiment. We plot the R_α in relation to $\mathbf{t}_z$ because it was influenced the most of all six pose parameters by the flips in our setup. The values are plotted in chronological order from frame 0 to frame 2289. Negative values of R_α indicate the occurrence of a pose flip.



(a) Frisbee with the marker.

(b) A 3D visualization of the tracked trajectory rendered from the cameras perspective.

Figure 8: Results of our rapid motion tracking experiment. (a) The frisbee that was used with the attached prototype marker. (b) A visualization of the recorded trajectory where we draw the coordinate axes of the marker at each tracked location. The x-, y- and z-axis of the marker are visualized red, green and blue and their length is equal to the radius of the frisbee (140 mm).

the first flips already occur about 0.5 meters closer to the camera. It can also be seen that the marker pose is estimated further away from the camera due to the elevation of $\mathbf{M}_6$ when the minimization ends up in the false minimum.

Our proposed method did not flip once in the whole sequence, which demonstrates the effectiveness of the pointer LED even at large distances despite its relatively small elevation. In all our experiments we did not manage to cause pose flips in the range where the LEDs were sufficiently visible to the camera to determine a pose.

4.5 Robustness to Rapid Motion

In this last experiment we attached the prototype marker to a frisbee (see Figure 8(a)) in order to demonstrate the capabilities of our system to robustly capture rapid motion. The exposure time of the camera was set to 2 ms for this experiment. Although this hardware setup would not be feasible for a realistic sports analysis of frisbee throws (at least not in this form), it is still a challenging example of combined rapid translational and rotational movement of a rigid object. We captured and tracked a 0.955 seconds long disc throw filmed from diagonal above the scene. The sequence consists of 154 frames that were all successfully tracked (see Figure 8(b)). The captured trajectory[3] started at $\mathbf{t}_{M,0} = (2577.4, 60.3, 4564.8)^\top$ and ended at $\mathbf{t}_{M,153} = (-2896.8, -660.8, 7093.4)^\top$ including nine full turns. Accordingly the frisbee traveled a total distance of 6.073 meters at an average speed of approximately 23 km/h. Note that also in this experiment no pose flips occured.

4.6 Failure Modes

Each marker can only be tracked if all seven LEDs are visible and separable in the respective frame. Hence self-occlusions caused by $\mathbf{M}_6$ lead to tracking failures. Filtering multiple false positive LED detections fails, if e.g. they appear at different sides around a marker and therefore do not end up in a single separate cluster. When tracking multiple markers, tracking will fail if their projected LED patterns are too close or even overlap due to the nature of the employed k-means algorithm.

[3] All measurements are in mm.

5 CONCLUSION AND FUTURE WORK

In this paper, we presented HSRM-Tracking, a method for robustly estimating and tracking the pose of multiple infrared markers with a single monocular camera. Thereby, individual markers can be correctly recognized in each single camera frame and distinguished based on the cross ratio of four collinear LEDs. Our evaluation results show that *HSRM*-Tracking is able to precisely capture fine and rapid movement up to 1000 Hz in a large area neglecting bandwidth limitations of current cameras.

The proposed method could easily be adapted for use in a multi-camera system where each camera runs in parallel in a separate tracking thread. Thus, cameras with different frame rates could be combined to track the markers asynchronously and contribute to a synchronized result whenever a new measurement is available, making camera synchronization unnecessary. Being able to estimate the marker pose from a single camera would also vastly increase the tracking volume of a multi camera setup and could be used in conjunction with stereo methods, whenever the marker is visible in more than one camera. Such a setup would also benefit from the LED identification scheme, since the markers could be used in order to dynamically calibrate the multi-camera system without having to solve stereo correspondence problems.

REFERENCES

Faessler, M., Mueggler, E., Schwabe, K., and Scaramuzza, D. (2014). A monocular pose estimation system based on infrared LEDs. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Fitzgerald, D., Foody, J., Kelly, D., Ward, T., Markham, C., McDonald, J., and Caulfield, B. (2007). Development of a wearable motion capture suit and virtual reality biofeedback system for the instruction and analysis of sports rehabilitation exercises. In *EMBS 2007*, pages 4870–4874.

Herout, A., Szentandrasi, I., Zacharia, M., Dubska, M., and Kajan, R. (2013). Five shades of grey for fast and reliable camera pose estimation. In *IEEE Conference on Computer Vision and Pat. Rec.*, pages 1384–1390.

Lakemond, R., Fookes, C., and Sridharan, S. (2013). Resection-intersection bundle adjustment revisited. *ISRN Machine Vision*, 2013:8.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168.

Lourakis, M. A. and Argyros, A. (2009). SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30.

MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441.

Naimark, L. and Foxlin, E. (2005). Encoded led system for optical trackers. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 150–153.

Nister, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770.

Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE.

Ong, S. K. and Nee, A. (2004). *Virtual Reality and Augmented Reality Applications in Manufacturing*. Springer Verlag.

Ozuysal, M., Calonder, M., Lepetit, V., and Fua, P. (2010). Fast keypoint recognition using random ferns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):448–461.

Prisacariu, V. and Reid, I. (2012). Pwp3d: Real-time segmentation and tracking of 3d objects. *International Journal of Computer Vision*, 98(3):335–354.

Schmaltz, C., Rosenhahn, B., Brox, T., and Weickert, J. (2012). Region-based pose tracking with occlusions using 3d models. *Machine Vision and Applications*, 23(3):557–577.

Schweighofer, G. and Pinz, A. (2006). Robust pose estimation from a planar target. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2024–2030.

Vito, L., Postolache, O., and Rapuano, S. (2014). Measurements and sensors for motion tracking in motor rehabilitation. *Instrumentation Measurement Magazine, IEEE*, 17(3):30–38.

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2008). Pose tracking from natural features on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 125–134.

Welch, G. and Foxlin, E. (2002). Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Comput. Graph. Appl.*, 22(6):24–38.

Xu, C., Kuipers, B., and Murarka, A. (2009). 3d pose estimation for planes. In *ICCV Workshop on 3D Representation for Recognition (3dRR-09)*.

Yang, H., Wang, F., Xin, J., Zhang, X., and Nishio, Y. (2012). A robust pose estimation method for nearly coplanar points. In *Proceedings NCSP '12.*, pages 345–348.

Zetu, D., Banerjee, P., and Thompson, D. (2000). Extended-range hybrid tracker and applications to motion and camera tracking in manufacturing systems. *IEEE Transactions on Robotics and Aut.*, 16(3):281–293.