An accurate and efficient camera-based indoor positioning approach for intralogistic environments

M. Jung
Scientific Assistant
Technische Universität München
Lehrstuhl fml

W. A. Günthner
Full Professor
Technische Universität München
Lehrstuhl fml

Indoor positioning of forklift trucks is often employed to track and trace goods in a warehouse. In the industry camera-based systems, which track artificial reference points (markers) on the floor or at the ceiling, have gained increasing attention. These systems use algorithms inspired by augmented reality software to estimate the camera's pose relative to a marker for 6 degrees of freedom. In this paper we show that in a simplified, yet realistic setup, where camera and marker plane are coplanar, 3 degrees of freedom are enough and devise an accurate, robust and highly computationally efficient algorithm to determine the relevant pose information. The advancements include position-based marker-tracking using a simple motion model, marker segmentation in the uncalibrated image and pose estimation using just the pinhole camera model. Conducted experiments show that both the algorithm's performance and accuracy are superior to state-of-the-art pose estimation algorithms for this simplified setup.

Keywords: *Indoor positioning, motion model, 3DoF pose estimation, computational efficiency, position-based marker tracking, forklift trucks, camera calibration.*

1. INTRODUCTION

The known position of conveyors is usually used to increase transparency of the transport process or is, in case of autonomous vehicles, even required for normal operation. On forklift trucks, which this paper focuses on, real-time position data can be used to identify pallets by the forklift's position during pick-up or drop-off and hence detect wrongful or unsatisfying transport actions immediately. Other applications of (real-time) position data are e.g. navigation, scheduling and safety purposes or analysis of warehouse movements [12, 20, 22].

All currently applicable approaches for positioning of forklift trucks in warehouses are marker-based, i.e. need some form of artificial reference points of which the absolute position is known. Radiocommunication-based approaches suffer from various drawbacks, like multi-path propagation and absorption. Hence optical systems have gained increasing attention in the industry. For autonomous vehicle positioning usually laser-scanners and specific reflectors as markers are used due to high accuracy and reliability requirements. For manually operated forklift trucks cheaper and physically more robust camera-based systems have prevailed [27, 29]. These systems at least partially [29] rely on fiducial markers also used in a lot of augmented-reality software and employ algorithms which solve the Perspective-n-Point problem (PnP) [6] to estimate the vehicles 6 degrees of freedom (DoF) pose relative to at least one currently visible marker [1, 5, 19, 23, 28]. Another significant issue with computer vision applications is the ressource requirements – to date either expensive special purpose hardware or powerful industry computers are required to be present on forklift trucks in order to perform the necessary computations. In the subsequent chapter we show why this is the case and later on propose methods to reduce the computational load to a negligeble level - potentially allowing such a positioning application to run in the background on slow standard forklift data terminals.

2. PERFORMANCE ANALYSIS

To get an overview of the bottlenecks in augmented reality software, the ArUco library [7] is chosen for analysis, because its algorithms are typical for marker-based AR-Software, the source code is freely available, the processing steps are well explained [2], and it has successfully been used for forklift localization in our testing area under realistic conditions in a previous research project [10]. The test is setup with a Smartek GC1392C [24] camera featuring a resolution of 1392x1040 pixels and a Tamron 219HB lense [26] both mounted on the roof of a forklift, pointing straight up and facing towards markers attached at the ceiling approx. 11 m above ground (see **Figure *2***).

Since the runtime of a computer vision algorithm can depend on the scene presented to the camera, we drive a short course roughly shaped like an "8" (see **Figure *1***) throughout large parts of the testing area while grabbing a grayscale video containing a total of 599 frames from the camera. A Laptop featuring a Core i7-2620M CPU with 2 Cores (though only 1 is used for the experiments), 4 GB RAM and Windows 7 Enterprise as operating system serves as execution platform. For all experiments the time required for the grabbing of each frame from the camera is neglected, as this step is always necessary. The camera's intrinsic parameters for calibration are determined using the software package from the Robotics Institute of CMU [21].

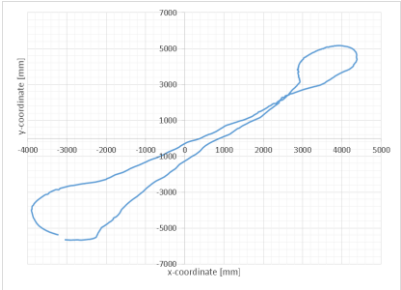


Figure 1. "8-like" shaped test course

Table *1* shows the measured average runtime per frame for each step (according to [2] over all 599 recorded frames.

Table 1. Runtime analysis

Step	Purpose	Runtime (ms)
Histogram equalization	Suggested by [10] for increased robustness of marker detection	2
Calibration [11, 15]	Allow accurate measurements by correcting distortions in the raw images	**40**
Marker detection	Find marker candidates	**7**
Marker identification	Determine the id of the candidates (if possible)	<1
Pose estimation default openncv algorithm [16]	Determine pose relative to detected marker	<1

Under the given circumstances it takes the test computer approx. 50 ms to evaluate a single image, limiting the number of processed images per second to be evaluated to <20. The two most prominent steps are obviously calibration and marker detection. According to Amdahl's law [8] a significant speedup can only be achieved by improving those two steps.

3. ADVANCEMENTS

In this chapter we first explain the basic parameters of the investigated indoor positioning system. We then propose ways of addressing the shortcomings of this approach regarding execution speed, robustness and accuracy.

3.1 Setup

We choose the same setup as [9] (see **Figure *2***). The markers are mounted at the ceiling and setup to be coplanar to the camera's image plane with the floor also assumed to be planar. In a real-world environment this kind of setup is often preferable, as markers at the ceiling are subject to less decay over time and the line-of-sight requirement between camera and marker can easy be satisfied.

Since a forklift only features 3 degrees of freedom (x,y,yaw) regarding positioning in a warehouse any measured, yet physically impossible rotation (i.e. roll, pitch angles) can be ignored. Such errors are typically caused by sensor noise and lighting conditions. Previous research shows, that for forklift positioning completely supressing the 2 irrelevant rotational degrees of freedom leaves only 4 degrees of freedom (DoF) to measure and leads to more accurate positioning.[9, 13]

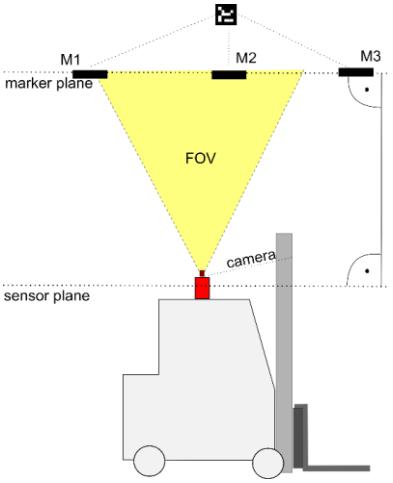


Figure 2. Camera and marker setup

3.2 Speedup

The most time-consuming steps can be significantly improved. The camera calibration (correction of distortions resulting from image sensor mount and lense errors) is necessary to perform accurate measurements with any camera [11], hence the error has to be corrected in every image and cannot be omitted entirely.

However, in order to make this step computationally insignificant, the correction can be decoupled from the finding of the marker in the image:

1. The marker is detected in the distorted image – since no relevant measurement is performed in this step, the distortion does not affect the final pose estimation result - as long as detection of the marker is possible. E.g. markers that consume large portions in the distorted image will be difficult to detect because the distortion will also modify their shape. This is why such an optimization is usually not implemented in augmented reality applications. For positioning applications small, yet robustly detectable markers are preferable.
2. The correction of the distortion is only performed on the 4 corners of the detected marker, which are relevant to the pose estimation algorithm.

This reduces the amount of pixels that need correcting from more than a million to the number of detected markers times 4.

Marker detection is the second most time-consuming step. A promising strategy towards reducing the runtime of a computer vision algorithm is to reduce the amount of pixels the algorithm has to consider by restricting it to a smaller region of interest (ROI) [4]. [18] determines the region of interest by predicting the future physical position of markers using a kalman filter and projecting that position onto the image plane of the camera again. Then the marker is detected within the projected region of the next frame. The same concept can be applied to our special case of indoor positioning, where all markers are placed at fixed positions and only the camera is moved relative to them. Hence all markers can be tracked and not only the currently visible ones

(see **Figure 3**). The following steps have to be performed:

1. Prediction of the camera position for the next frame.
2. Computation of the camera's field of view in that position.
3. Projection of all markers within the field of view onto the image plane. To handle errors in the position measurement and deviation from the derived prediction the regions of interest are increased in size. To that end the areas to be projected onto the image plane are virtually increased in size by an arbitrary ±10cm in every direction, thus creating a bigger distance between the corner points on the image plane as well, effectively creating an additional margin.
4. Determination of the rectangle-shaped region of interest containing the projected corner points for further investigation. Since the id of the marker assumed in the specific region is known, it can later be used to improve the marker identification step.

Assuming the prediction is always good enough to find the marker near the predicted position and a marker is always visible in and can be extracted from every field of view, only the predicted regions of interest need to be evaluated in order to extract the position information from the second frame onwards. Since the region of interest is much smaller than the original image for small-appearing markers in the image, the computation time of the positioning algorithm is dramatically reduced as a result - even in the worst case scenario only a region of interest has to be evaluated. The full frame only needs to be evaluated once during setup while all following measurements can rely on the previous position.

In order to satisfy the two conditions above, both the determined pose and the derived predicted pose need to be accurate in order to target the proper region of interest. Also the segmentation algorithm which detects the marker candidates in the image and the algorithm which identifies the marker's id both need to be robust.

The markers need to be strategically placed below the ceiling in such a way, that every possible field of view from a valid camera pose contains at least one marker which is not (partially) occluded. To address the first condition we employ a simple vector-based motion model, which assumes the motion in all degrees of freedom continues as before according to the following formula (for fixed time intervals):

$$\vec{x}_{t+1} = \vec{x}_t + (\vec{x}_t - \vec{x}_{t-1}) \quad (1)$$

If the changes in motion are small, the deviation from previous course and speed are also small and and hence the prediction is reasonably close to the actual value. For forklifts common in indoor environments possible deviations in speed are due to acceleration (below +1 m/s^2 [25],[17]) and deceleration (-1.3 m/s^2 or more required by [14]). The deviation due to deceleration between two frames can be determined using formula (2).

$$\Delta s = -\frac{1}{2} a t^2 \quad (2)$$

If we assume a deceleration of -1.5m/s^2 and a camera running at 25 Hz, then $t = \frac{1}{25} s$ and $|\Delta s| < 1.2$ mm, which is way below the accuracy of the positioning system. As long as the forklift is driving in a straight line, no significant deviation is to be expected.

Deviation from the predicted course can also happen due to an angle change caused by steering of the driver. Unfortunately forklift data sheets usually do not contain information regarding angular acceleration, hence the position deviation caused by this effect is difficult to estimate. Worst case scenario of this deviation is a rapid in-place turn of the forklift, as higher speeds do not allow for high angular acceleration without causing the forklift to flip. Initial experiments indicate the rotation prediction error to be lower than 1° for reasonable driving. Another important influence on the quality of the prediction is the accuracy of the previous positionings. Since the acuracy of the system is unknown, its effect cannot be quantified a priori.

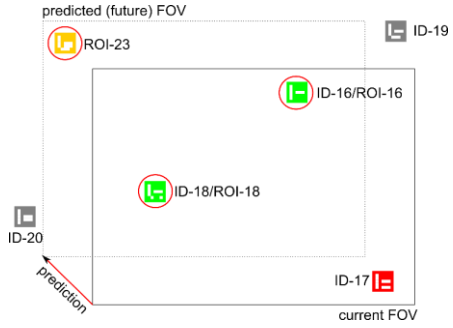


Figure 3. Current and predicted field of view (FOV) as seen from the marker plane towards the camera plane. Color denotes the state of the marker in the predicted FOV stays visible→green, no longer visible→red, becomes visible→ orange, not visible→ gray. Circles denote the region of interest.

3.3 Robustness

Since the restriction to a specific region of interest reduces the number of pixels to be considered significantly, additional, yet on the full image time-consuming algorithms, like sharpening the image can be used to increase the robustness of marker detection. Since the region of interest is small, the performance impact is low.

Also decoding the marker's id in a region of interest is no longer necessary because the region of interest is already associated with that information. Hence only the marker corners need to be detected. The identification step is reduced to mapping the 4 predicted marker corners to the most likely quadrilateral to be a marker:

1. The most likely quadrilateral features the minimal sum of squared differences between the 4 projected (target) lengths of the marker and the detected quadrilateral's (actual) line lengths.
2. The corner points of the quadrilateral then can be mapped to the marker's corners to get the proper orientation of the marker (assignment problem) by minimizing the sum of squared differences between the respective corner point coordinates:

$$\min \sum_{i=1}^{4} (\vec{p}_{mi} - \vec{p}_{ci})^2 \quad (3)$$

Since there are only 4! = 24 combinations, they can simply all be computed and compared to obtain the best mapping. If evaluation of all regions of interest fails, the entire frame still can be searched for markers at the cost of performance.

3.4 Accuracy

For the given setup, Hohenstein proposes an algorithm (PPP) to directly compute the pose through use of the equations from the pinhole camera model. To obtain the position of the camera relative to a marker, the centre of the marker in the image plane is determined and together with the camera intrinsic parameters (f and c) from the calibration the position in the x- and y-plane can be computed analytically using equation (4). [3, 13]

$$x = \frac{z \cdot (x_{screen} - c_x)}{f_x},\ y = \frac{z \cdot (y_{screen} - c_y)}{f_y} \quad (4)$$

Hohenstein uses the rate between the markers image perimeter and the corresponding physical perimeter to estimate the z-coordinate and then computes x- and y-coordinates. The orientation (yaw angle) can be efficiently computed as the mean value of the angles between the lines passing through the marker's edges and the corresponding horizontal/vertical lines. [13]

Since the distance between marker and floor is constant over time, it only needs to be determined once with a high degree of accuracy, e.g. using a laser rangefinder. The determined values (minus the height of the camera above ground) can then be used to eliminate another potential error source in the pose estimation, turning the approach into a true 3 degrees of freedom pose estimation. From hereon it is denoted as PPP+.

If more than one marker can be detected, the weighted average, with the distances of the markers to the image border serving as weights, of all individual poses is computed and taken as final camera pose. This method was previously also used by Günthner et al. to reduce the impact of measurement errors due to noise and camera calibration issues, later of which more often occur in the outer parts of the camera field of view. [9]

4 EXPERIMENTS

To evaluate our approach regarding speed, robustness and accuracy a series of experiments are conducted.

4.1 Speedup

Regarding execution time a speedup of almost 26 has been achieved (see **Table *2***), i.e. the program now runs 26 times as fast as before. The required effort for the entire processing chain was reduced to less than 2ms per frame on average. Evaluation of the region of interest (image enhancement, marker detection/identification) are the only significant steps, all other program parts are still negligible.

Table 2. Improved runtime per processing step

Step	Runtime before (ms)	Runtime improved (ms)	Note
Histogram equalization	2	0	Integrated into detection
Calibration	40	0,008*	
Marker detection	7	2	
Marker identification	<1	0	obsolete
Pose estimation	<1	0,004*	Improved: for PPP+
* values within measurement accuracy of the clock			

4.2 Accuracy and Robustness

The following experiments aim at comparing the pose estimation algorithms regarding accuracy.

At first a synthetic test is performed, towards identifying the impact of noise on the pose estimation for the three algorithms. Only one marker is used in this setup, the forklift is moved to a position, such that the marker is visible at 9 different positions in the image (see **Figure *4***).

1	2	3
4	5	6
7	8	9

Figure 4. The 9 marker positions in the image

At each of the 9 positions 100 images are recorded. Since the position does not change for 100 images, any change in the estimation can be attributed to noise. For each algorithm the mean value is computed over all 100 samples and the Euclidian distance of those samples to this mean as well as the difference between the maximum value and the minimum value are determined. The standard deviation of the distances shows the fluctuations in the measurements, while the maximum difference shows the worst case. To combine all 9 positions into one, the average of the standard deviations is taken for comparing the algorithms against each other by one single number. The maximum deviation over all 9 positions is also computed. This experiment (**Table *3***) shows, that of all 3 algorithms, PPP+ is least influenced by noise followed by RPP and Opencv's default algorithm, which is also the least precise according to the maximum difference values.

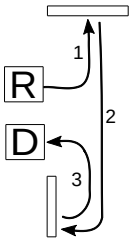
Table 3. deviation per algorithm in static setup

	Translation		Rotation	
Pose estimation	Mean Stdev (mm)	Max(Max-Min) (mm)	Mean Stdev (°)	Max(Max-Min) (°)
Opencv default	4,96	94,8	0,128	2,28
RPP	3,09	21,3	0,0515	0,538
PPP+	1,76	18,5	0,0382	0,414

Correspondence to: Matthias Jung, scientific assistant
Technische Universität München,
Boltzmannstr. 15, 85748 Garching, Germany
E-mail: jung@fml.mw.tum.de

To stress the algorithms under realistic conditions a forklift is tracked with all three algorithms while driving a double-cycle course:

1. A pallet is picked up from goods receipt and stored in a high-rack storage area.
2. Directly afterwards another pallet is picked up from a different high-rack storage area and
3. brought to dispatch.



The shape of the course covers both straight forward parts and turns in both directions, which the prediction algorithm has to cover. We repeated the course 5 times to get a sufficient statistical basis for evaluation (9286 images). All improvements were activated at once, only the pose estimation algorithm is compared against both the default opencv method and the robust planar pose algorithm [23]. Both candidates were chosen because they performed well in previous experiments [10]. Another goal is to gain more detailed insight into the (correct) prediction of the region of interest and the relationship between the margin size and the prediction rate.

A prediction is classified as a success if at least one marker was extracted in one of the regions of interest and as a failure otherwise. The success rate is hence the number of successes divided by the total number of frames minus the number of experiments (since prediction in the first frame is impossible). In case of a failure, the entire image has to be searched for markers (at the cost of performance). To study the influence of the margin added around the predicted area, it is varied in sizes (results see **Table *4***).

Due to lack of a reference system the accuracy of the algorithms cannot be directly compared while the forklift is moving. Hence two indicators are employed that at least allow for indirect comparison of the pose estimation algorithms. The first indicator is the success rate. Since the image sequences are the same, any difference in the success rate between algorithms can at least partially be attributed to the accuracy of the pose estimation. The informative value of this indicator is limited as the cause for a prediction failure could have additional other reasons. The second indicator therefore takes the distance between predicted and the actual (afterwards computed) position into account (**Table *5***). This way a more detailed picture should emerge in case there is a significant difference between the algorithms performance.

Even though the use of PPP+ delivered the best result, the effectiveness of tracking is very high for all pose estimation algorithms. Also an improved robustness can be noted, as the number of frames, where no marker was found without tracking was reduced from 72 to 0. The redundantized marker identification step has the most significant impact on the increase in robustness.

When the PPP+ algorithm is employed, only 3 predictions failed for two reasons:

1. For few frames there is a significant discrepancy between a frame and its successor (severe speed gradients between two frames). The most likely reason is that the frames were not precisely recorded with the assumed sampling rate. To avoid such situations in the future the timestamps to all camera frames should be recorded if possible for later analysis of this problem.
2. The markers absolute position in the world coordinate system may need to be measured more accurately. Otherwise a correct prediction can be identified as a failure, just because the marker is not where it is configured to be.

Table 4. Success rate for different margin sizes

	Success rate		
Margin (cm)	10	5	0
Opencv default	0.9945	0.9851	0,9601
RPP	0.9950	0.9858	0,9609
PPP+	0.9997	0.9926	0,9767

Table 5. Deviaton from predicted course during double cycle experiment

	Translation (mm)		
	Mean	Median	Stdev
Opencv default	52,6	24,4	113
RPP	57,5	22,7	134
PPP+	21,7	9,71	33,5
	Rotation (°)		
	Mean	Median	Stdev
Opencv default	0,89	0,46	1,51
RPP	0,97	0,38	1,97
PPP+	0,32	0,15	0,53

The success rate compared to the window size shows that PPP+ is superior and can cope better with a reduced margin. The other two algorithms still present a good performance, there is no significant difference between the default algorithm from opencv and RPP. The same behaviour can be observed from the deviation between the predicted values and the actual position. The mean and median values of PPP+ are more than 50% lower (>60% for rotations) than for the other algorithms, which present a similar performance if compared to one another. The high standard deviations of the first two algorithms indicate a high fluctuation. The raw data also shows big differences between predicted and measured location for some measurements. Over all PPP+ significantly outperforms the other two algorithms regarding prediction accuracy and hence also measurement accuracy.

5 CONCLUSION

In this paper we showed a couple of techniques that can be successfully applied to marker-based optical positioning to make execution speed negligible on modern hardware. Specifically the combination of tracking the markers based on a motion model significantly improves both performance and robustness at the same time. Combined with the simple pose estimation algorithm also presented in this paper, the prediction quality can be further improved and lead to almost 100% prediction success, both in synthetic experiments and realistic logistic scenarios.

The good results from the experiments warrant further investigation. The achieved speedup makes execution on legacy hardware, even old forklift data terminals potentially possible. Also low-cost embedded devices might now be able to handle the required computational load. Use of such systems would lead to much cheaper localization for forklift trucks. The accuracy of the system should also be futher investigated. Through comparison to a reference system the accuracy could be determined in detail and further applications might be possible once the basic performance indicators of the approach are known.

REFERENCES

[1] Abidi, M. A. and Chandra, T. 1995. A new efficient and direct solution for pose estimation using quadrangular targets: algorithm and evaluation. *IEEE Trans. Pattern Anal. Machine Intell.* 17, 5, 534–538.

[2] Aplicaciones de la Visión Artificial. 2014. *ArUco. a minimal library for Augmented Reality applications based on OpenCv*. http://www.uco.es/investiga/grupos/ava/node/26. Accessed 15 June 2015.

[3] Bradski, G. and Kaehler, A. 2012. *Learning OpenCV. Computer Vision in C++ with the OpenCV Library*. O'Reilly & Associates, Sebastopol, CA.

[4] Brinkmann, R. 2008. *The art and science of digital compositing. Techniques for visual effects, animation and motion graphics*. Morgan Kaufmann Publishers/Elsevier, Amsterdam, Boston.

[5] Dementhon, D. F. and Davis, L. S. 1995. Model-based object pose in 25 lines of code. *Int J Comput Vision* 15, 1-2, 123–141.

[6] Fischler, M. A. and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6, 381–395.

[7] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6, 2280–2292.

[8] Gene Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. In *AFIPS spring joint computer conference, 1967*, 483–485.

[9] Günthner, Hohenstein, Jung. 2014. *Das Staplerauge*. Lehrstuhl für Fördertechnik Materialfluss Logistik, München.

[10] Günthner, Hohenstein, Jung. 2014. *Das Staplerauge*. Forschungsbericht, TU München.

[11] Heikkila, J. and Silven, O. A four-step camera calibration procedure with implicit image correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

[12] Heißmeyer S, Overmeyer L, Müller A. Optical Indoor Positioning of Vehicles. In *Logistics Journal, Vol. 2013, Iss.*

[13] Hohenstein. 2014. *Systementwurf und Umsetzung einer funktionsintegrierenden Gabelstaplerlokalisierung für eine wandlungsfähige und effiziente Transportausführung*. Dissertation. Technische Universität München, München.

[14] ISO. 2008. *Powered industrial trucks and tractors -- Brake performance and component strength*, 6292.

[15] Itseez. 2014. *OpenCV. Geometric Image Transformations*. http://tinyurl.com/otxnd32. Accessed 20 November 2014.

[16] Itseez. 2014. *OpenCV. Camera Calibration and 3D Reconstruction*. http://tinyurl.com/adtegnz. Accessed 20 November 2014.

[17] Jungheinrich AG. 2014. *EFG 110/110k/113/115. Electric three-wheel truck (1,000/1,250/1,500 kg)*. http://tinyurl.com/qd8zfht. Accessed 17 June 2015.

[18] Klaus Dorfmüller-Ulhaas. 2007. *Robust Optical User Motion Tracking Using a Kalman Filter*. Bericht, Universität Augsburg.

[19] Lepetit, V., Moreno-Noguer, F., and Fua, P. 2009. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int J Comput Vision* 81, 2, 155–166.

[20] Liu, H., Darabi, H., Banerjee, P., and Liu, J. 2007. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Trans. Syst., Man, Cybern. C* 37, 6, 1067–1080.

[21] Mirai Higuchi, Ankur Datta, Takeo Kanade. 2012. *Software Package for Precise Camera Calibration*. http://tinyurl.com/q3xqdga. Accessed 17 June 2015.

[22] Rizos, C., Roberts, G., Barnes, J., and Gambale, N. Experimental results of Locata: A high accuracy indoor positioning system. In *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1–7.

[23] Schweighofer, G. and Pinz, A. 2006. Robust pose estimation from a planar target. *IEEE Trans. Pattern Anal. Machine Intell.* 28, 12, 2024–2030.

[24] smartek vision. 2015. *Datasheet Giganetix Camera. GC1392M GC1392C*. http://tinyurl.com/o2arl64. Accessed 15 June 2015.

[25] Still GmbH. 2015. *RX 50 Technical Data. Electric Forklift Truck*. http://tinyurl.com/ndef2fc. Accessed 17 June 2015.

[26] TAMRON. 2008. *219HB*. http://tinyurl.com/pg2oz9x. Accessed 17 June 2015.

[27] TotalTrax, I. 2015. *The Sky-Trax System™*. http://tinyurl.com/ocw9vsm. Accessed 18 June 2015.

[28] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. 2003. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Machine Intell.* 25, 8, 930–943.

[29] ZENO TRACK GmbH. 2014. *Fahrzeugortung*. http://www.zenotrack.com/index.php?id=48. Accessed 22 September 2014.

