Camera- and IMU-based Pose Tracking for Augmented Reality

Florian Faion, Antonio Zea, Benjamin Noack, Jannik Steinbring, and Uwe D. Hanebeck

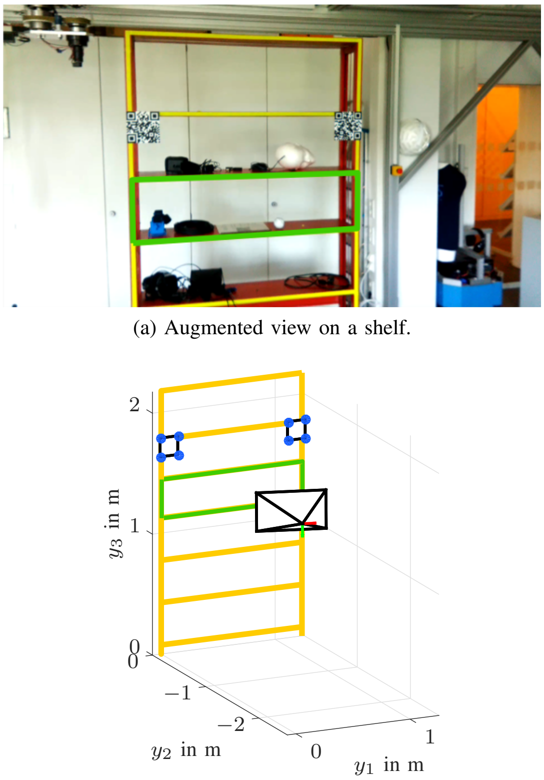
***Abstract*— In this paper, we propose an algorithm for tracking mobile devices (such as smartphones, tablets, or smartglasses) in a known environment for augmented reality applications. For this purpose, we interpret the environment as an extended object with a known shape, and design likelihoods for different types of image features, using association models from extended object tracking. Based on these likelihoods, and together with sensor information of the inertial measurement unit of the mobile device, we design a recursive Bayesian tracking algorithm. We present results of our first prototype and discuss the lessons we learned from its implementation. In particular, we set up a "pick-by-vision" scenario, where the location of objects in a shelf is to be highlighted in a camera image. Our experiments confirm that the proposed tracking approach achieves accurate and robust tracking results even in scenarios with fast motion.**

I. INTRODUCTION

Augmented reality is one of the core technologies [1] in Industry 4.0. A formal definition can be found in [2], where it is introduced as *"expansion of physical reality by adding layers of computer-generated information to the real environment"*. The specific AR-application which motivated this work is optimizing industrial production flow by assisting workers in tasks related to picking up objects. This technology is known as *pick-by-vision* [3], and extends traditional approaches, such as *pick-by-paper*, *pick-by-voice*, and *pick-by-light*. Its working principle is shown in Fig. 1a, where a single storage bay in a shelf is highlighted in the camera image. For this augmentation, essentially two types of information are required: first, the location of the target object, which usually is provided by a database, and second, the 6DOF pose of the camera (or smartglasses display) with respect to the storage environment. In this paper, we are concerned with developing a tracking system for the second requirement. In doing so, we only want to use sensors which are embedded in the mobile device, i.e., camera and inertial measurement unit (IMU).

A. Related Work

Recently, the first commercial pick-by-vision systems [1] entered the market by Samsung, SAP, DHL, and Picavi, among others. However, these systems currently do not come with 6DOF pose tracking of the smartglasses yet. Instead, they rely on combinations of various types of barcodes and quick response (QR) codes, which are to be scanned by the worker in order to confirm performed tasks. Detecting these markers, however, requires a high image quality which, in turn, requires a sufficiently careful movement of the camera. More advanced prototypes with pose tracking, such as the ones proposed by KNAPP AG [4] or SAP, look promising, but are still in the development stage. In the following, we highlight some of the underlying technologies which are related to the proposed tracking approach.



(a) Augmented view on a shelf.

(b) Model for tracking and augmentation.

Fig. 1: Working principle of the proposed algorithm. Based on a geometric model of the environment (b), the camera pose is estimated and the augmented view on the scene is calculated (a).

Visual odometry [5], as well as visual simultaneous localization and mapping [6], allow for tracking the 6DOF pose of a camera. However, these approaches originally assume an a priori unknown environment. Dead reckoning based on an IMU [7] uses sensor information about an object's acceleration and rotational speed in order to infer the traveled trajectory. In doing so, it can be seen as a complementary technology to the visual approach. Thus, there has been effort to combine both approaches [8] in order to take advantage of their specific properties.

Image feature descriptors, such as SIFT [9] or SURF [10] allow for the use of natural (more or less) unique markers.

All authors are with Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany. `florian.faion@kit.edu`

Other technologies related to the proposed approach are odometry based on depth cameras [11], and using the object's visual hull [12] to estimate its pose.

B. Contribution

In this paper, we develop an algorithm for tracking the 6DOF pose of a mobile device, which is robust enough for the use in a pick-by-vision system. As the main contribution, we show how to apply techniques from extended object tracking to address this problem. For this purpose, we model the geometry of the known environment and interpret it as the shape of an extended object, which is measured in the form of pixels in the camera image. Defining a probabilistic model for these measurements lets us apply association models from extended object tracking. Note that, in contrast to many vision based approaches, we do not actively use intensity of a pixel but rather its position in the image.

C. Outline

In Sec. II, we give an overview of the proposed tracking algorithm and its components. Subsequently, in Sec. III, Sec. IV, and Sec. V, we show how to incorporate different types of measurements into the algorithm. Finally, in Sec. VI and Sec. VII, we present a tracking experiment, and discuss its results.

II. Tracking Algorithm

This section gives an overview of the proposed tracking algorithm. We start with introducing the desired state parameters, followed by a discussion of the considered measurement modalities. Finally, we show how to design a recursive Bayesian tracking algorithm for this estimation problem and discuss its relationship to extended object tracking.

A. System State

We are interested in estimating the 6DOF pose (position and orientation) and dynamics of a mobile device. The desired information can be stacked into a nine-dimensional state vector

$$\underline{x} = \begin{bmatrix} \underline{t} \\ \dot{\underline{t}} \\ \underline{r} \end{bmatrix} , \tag{1}$$

which encodes the camera position and velocity in the form of two three-dimensional vectors $\underline{t}$ and $\dot{\underline{t}}$, and its orientation as a three-dimensional rotation vector $\underline{r}$ in axis-angle representation [13]. In this representation, the unit vector $\underline{r}/\|\underline{r}\|$ encodes the rotation axis and $\|\underline{r}\|$ describes the angle of rotation around this axis. Given the rotation vector $\underline{r}$, a 3×3 rotation matrix $\mathbf{R} = \Phi(\underline{r})$ can be derived using the Rodrigues formula (vice versa $\underline{r} = \phi(\mathbf{R})$). We decided to exclude other motion related parameters (such as rotational velocity or acceleration) from the state, and instead incorporate them as noisy input parameters provided by the IMU.

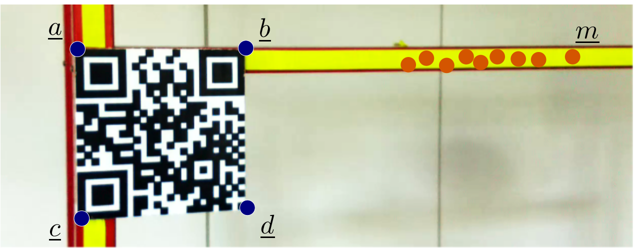


Fig. 2: Unique ($\underline{a}$, $\underline{b}$, $\underline{c}$, $\underline{d}$) and ambiguous ($\underline{m}$) markers, which are used for tracking.

B. Measurements

The system state $\underline{x}$ should be estimated based on measurements from the sensors embedded in the mobile device. Typically, these are an IMU and an RGB-camera. We assume that the IMU measures a linear acceleration $\ddot{\underline{t}}$ (where the gravity has been internally removed), and a rotational velocity $\dot{\underline{r}}$ in axis-angle representation. For the camera, we assume that it can measure two different types of markers: unique ones and ambiguous ones (see Fig. 2).

Unique markers yield measured pixel positions whose origin in 3D space is exactly known. As an intuitive example, consider a QR code, which is placed at a known location in the tracking environment. When we observe the QR code in the camera image, we exactly know a one-to-one correspondence of the four corner pixels

$$\underline{q} = \begin{bmatrix} \underline{a} \\ \underline{b} \\ \underline{c} \\ \underline{d} \end{bmatrix} \tag{2}$$

to their originating points $\underline{A}$, $\underline{B}$, $\underline{C}$, $\underline{D}$ in 3D space. Measurements from more than one QR code are aggregated into a set $Q = \{\underline{q}_1, \ldots, \underline{q}_{N_Q}\}$.

Ambiguous markers do not have a known one-to-one correspondence and, hence, contain less information than unique ones, but are generally much easier to extract. As an example, consider the colored line segments in Fig. 2, which are mounted in the tracking environment. As indicated in the figure, measurements yield a set $M = \{\underline{m}_1, \ldots, \underline{m}_{N_M}\}$ of two-dimensional pixel coordinates $\underline{m}$. The ambiguity originates from the fact that each pixel is known to originate from a point $\underline{M}$ in the 3D space where yellow tape is mounted, but not which one exactly. In consequence, we have to deal with a one-to-many association problem.

We decided for these measurement modalities due to the following reasons. First, the IMU provides information in situations where no marker is visible or detected. Second, unique markers are required for (re)-initialization periods, as they allow for accurate localization. Finally, as image quality degrades in periods with fast motion and IMU-based dead reckoning quickly diverges, we also require ambiguous markers. Note that, despite the fact that we use artificial markers in this paper, we could have also used natural landmarks.

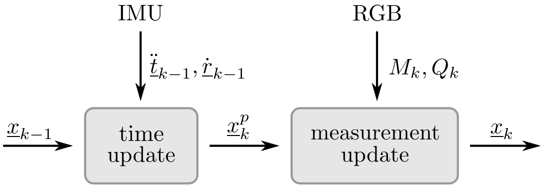


Fig. 3: Processing pipeline of the proposed tracking algorithm.

C. Recursive Bayesian Estimation

The task now is to design a tracking algorithm that continuously estimates the state parameters $\underline{x}_k$ at each time k from the various measurement modalities. For this purpose, we model the state as a random variable $\underline{x}_k$ with probability distribution $p(\underline{x}_k)$ and use a recursive Bayesian estimator [14] to maintain it according to the processing pipeline in Fig. 3.

The *time update step* determines how the prior distribution $p(\underline{x}_{k-1})$ at the time $k-1$ will evolve to the time k according to

$$p(\underline{x}_k^p) := \int_{\mathbb{R}^9} \underbrace{p(\underline{x}_k^p|\underline{x}_{k-1})}_{\text{IMU}} \cdot p(\underline{x}_{k-1}) \; \mathrm{d}\underline{x}_{k-1} \; , \tag{3}$$

where $p(\underline{x}_k^p)$ denotes the distribution of the predicted state, and the transition probability $p(\underline{x}_k^p|\underline{x}_{k-1})$ is specified by a dynamic model [15], which follows from the IMU data.

The *measurement update step* lets us incorporate marker measurements M_k, Q_k into a given distribution $p(\underline{x}_k^p)$ according to Bayes' rule

$$p(\underline{x}_k^p|M_k, Q_k) \propto p(M_k, Q_k|\underline{x}_k^p) \cdot p(\underline{x}_k^p) \; , \tag{4}$$

where the likelihood $p(M_k, Q_k|\underline{x}_k^p)$ rates how well the state parameters $\underline{x}_k^p$ fit to the measurements with respect to the sensor-specific uncertainty. In order to close the recursion, the updated distribution is considered to be the new prior $p(\underline{x}_k) := p(\underline{x}_k^p|M_k, Q_k)$. For the case that no marker has been detected, the predicted distribution is set to the new prior.

Assuming the sensor noise terms to be independent between the unique and ambiguous markers, the likelihood can be factorized as

$$p(M_k, Q_k|\underline{x}_k^p) = \underbrace{p(M_k|\underline{x}_k^p)}_{\text{ambiguous markers}} \cdot \underbrace{p(Q_k|\underline{x}_k^p)}_{\text{unique markers}} \; , \tag{5}$$

which allows us to consider both types of measurements individually. In the following sections, we derive the components $p(\underline{x}_k^p|\underline{x}_{k-1})$, $p(Q_k|\underline{x}_k^p)$, and $p(M_k|\underline{x}_k^p)$. The relationship to extended object tracking will become apparent in the likelihood for the ambiguous markers M_k, where we have to deal with an instance of the association problem, as state parameters and measurements are only connected through their unknown measurement sources.

III. Inertial Measurements

Let us first look at the time update step, where we have to specify the transition probability $p(\underline{x}_k^p|\underline{x}_{k-1})$ with respect to the IMU data. As introduced in Sec. II-B, the IMU consists of an accelerometer and a gyroscope, which respectively provide linear acceleration $\ddot{\underline{t}}_{k-1}$ and rotational velocity $\dot{\underline{r}}_{k-1}$ at a given time step $k-1$. Let us assume these values to be additively distorted by six-dimensional, zero-mean, white Gaussian noise $\underline{v} \sim \mathcal{N}(\underline{0}, \mathbf{C}_v)$ where $\underline{v} = [\underline{v}_t^\mathrm{T}, \underline{v}_r^\mathrm{T}]^\mathrm{T}$. Note that this noise sometimes is modeled in more sophisticated ways, e.g., time-variant, or non-zero-mean. An equivalent, but more intuitive representation of the transition probability is in the form of its dynamic model $\underline{x}_k^p = a(\underline{x}_{k-1}, \underline{v})$ that specifies how the system state evolves over time. Incorporating the typical relationships between path, velocity and acceleration and the IMU data, this dynamic model yields

$$\begin{aligned} a(\underline{x}_{k-1}, \underline{v}) &= \begin{bmatrix} \underline{t}_k \\ \dot{\underline{t}}_k \\ \underline{r}_k \end{bmatrix} \\ &= \begin{bmatrix} \underline{t}_{k-1} + \dot{\underline{t}}_{k-1} \cdot \tau_k + \frac{1}{2} \cdot \mathbf{R}_{k-1}(\ddot{\underline{t}}_{k-1} + \underline{v}_t) \cdot \tau_k^2 \\ \dot{\underline{t}}_{k-1} + \mathbf{R}_{k-1}(\ddot{\underline{t}}_{k-1} + \underline{v}_t) \cdot \tau_k \\ \phi\left(\mathbf{R}_{k-1}\Phi(\dot{\underline{r}}_{k-1} + \underline{v}_r) \cdot \tau_k\right) \end{bmatrix} \; , \end{aligned} \tag{6}$$

where τ_k denotes the time difference between $k-1$ and k. In addition, applying the rotation matrix $\mathbf{R}_{k-1} = \Phi(\underline{r}_{k-1})$ ensures that the inertial measurements, which are given in the local coordinate system of the IMU, are correctly transformed to the global coordinate system.

IV. Unique Markers (QR Codes)

In this section, we specify the likelihood $p(Q_k|\underline{x}_k^p)$ for the unique marker measurements. For the following derivations, we consider a single time step k, as well as a single state instance, which allows us to drop their indices. As discussed in Sec. II-B, a unique marker refers to a one-to-one correspondence between measured pixel $\underline{a} = [a_1, a_2]^\mathrm{T}$, and its source $\underline{A} = [A_1, A_2, A_3]^\mathrm{T}$ in 3D space. For the likelihood, we have to establish a relationship between state parameters $\underline{x}$ and measurement $\underline{a}$. The essential idea is to project the known origin $\underline{A}$ of the measurement onto a hypothetical camera image, which is specified by the camera pose $\underline{x}$. Then, for the true state parameters, the predicted pixel position should coincide with the measured position. To implement this idea, we require two components. First, all unique sources $\underline{A}$ have to be specified in 3D, as illustrated by the blue points in Fig. 1b. Second, we need a camera model $\underline{a} = \mathrm{proj}(\underline{A}, \underline{x})$, which lets us project 3D points $\underline{A}$ onto their corresponding pixels $\underline{a}$ for a given camera pose $\underline{x}$. Assuming a pinhole model, this projection is given by

$$\gamma \cdot \begin{bmatrix} a_1 \\ a_2 \\ 1 \end{bmatrix} = \mathbf{K} \left(\mathbf{R} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} + \underline{t} \right) \; , \tag{7}$$

where $\mathbf{K}$ is the 3×3 intrinsic camera matrix, the scalar γ is dropped after homogenization, and $\mathbf{R}$, $\underline{t}$ are the pose parameters of the camera, which are encoded in the state $\underline{x}$.

Under the assumption of isotropic Gaussian noise for the measured pixel location, the likelihood can be defined as illustrated in Fig. 4 (left). In this figure, the predicted QR code corners $\mathrm{proj}(\underline{A}, \underline{x})$ and their uncertainties are drawn as

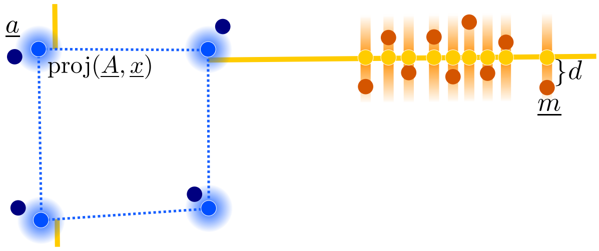


Fig. 4: Schematic of the likelihood components for unique and ambiguous markers for the scene in Fig. 2. For a given instance of state parameters $\underline{x}$, the yellow and blue lines are projections of the unique and ambiguous markers.

shaded, blue circles. Evaluating the likelihood then simply requires evaluating the Gaussians in the measurements $\underline{a}$.

Now let us assume that we have measured a total of N_Q QR codes, i.e., $Q = \{\underline{q}_1, \dots, \underline{q}_{N_Q}\}$, and let us assume independence between the measurement noise terms. Then, we end up with the following expression

$$\begin{aligned} p(Q|\underline{x}) &= \prod_{i=1}^{N_Q} p(\underline{q}_i|\underline{x}) \\ &= \prod_{i=1}^{N_Q} \mathcal{N}\left(\begin{bmatrix} \underline{a} \\ \underline{b} \\ \underline{c} \\ \underline{d} \end{bmatrix}_i ; \begin{bmatrix} \mathrm{proj}(\underline{A}, \underline{x}) \\ \mathrm{proj}(\underline{B}, \underline{x}) \\ \mathrm{proj}(\underline{C}, \underline{x}) \\ \mathrm{proj}(\underline{D}, \underline{x}) \end{bmatrix}_i , \sigma_q^2 \cdot \mathbf{I} \right) , \end{aligned} \quad (8)$$

with σ_q^2 being the variance of the Gaussian.

Note that, given the corners of a QR code and their measurement sources in 3D space, calculating the camera pose is an instance of the classical perspective-n-points (PNP) problem in computer vision, with solutions, e.g., in [16], which can be used for calculating an initial pose estimate.

V. Ambiguous Markers (Line Segments)

Similar to the unique markers, we can specify a likelihood $p(M|\underline{x})$ for the measured ambiguous markers $M = \{\underline{m}_1, \dots, \underline{m}_{N_M}\}$. As introduced in Sec. II-B, each measurement $\underline{m}$ is known to originate from a 3D source $\underline{M}$, which lies on a line segment. Unfortunately, we know neither from which line segment a measurement occurred, nor from which source on a given line segment it occurred.

This one-to-many association is a common problem in extended object tracking, and in previous research on line segments [17] we found two major concepts for designing the likelihood. As the first option, a spatial distribution model would explicitly incorporate the probability for each point on each line segment of being the true measurement source. Specifying this probability is difficult, as it requires taking into account occlusions and other unpredictable factors. An easier option is the greedy association model, which imposes the “most likely” source over all line segments to have produced the measurement $\underline{m}$. A visual explanation of this model can be found in Fig. 4 (right), where for each point $\underline{m}$, the closest source is used as an approximation for the true measurement source.

Then, for implementing a likelihood based on this model, we again require several components. First, all 3D line segments $L_1, \dots, L_{N_L}$ which potentially could produce measurements must be specified (yellow lines in Fig. 1b). This can be done by manually measuring and storing the 3D start- and end-points for each segment. Second, the projection in (7) must be extended to allow for projecting 3D line segments L onto their 2D equivalents $\underline{l}$ in the hypothetical camera image. For this purpose, start- and end-point of each segment can be projected individually using (7). Finally, in order to find the closest point on a 2D line $\underline{l}$ for a given pixel $\underline{m}$, we need a distance function $\mathrm{dist}(\underline{m}, \underline{l})$. This distance, in turn, can be calculated in closed-form using standard algebraic techniques.

Based on these prerequisites, we can find the closest distance over all line segments according to

$$d_i = \min_{j=1 \dots N_L} \mathrm{dist}\left(\underline{m}_i, \mathrm{proj}(L_j, \underline{x})\right) . \quad (9)$$

Then, this scalar distance value can be used to design one-dimensional likelihoods for each measured pixel in the form of

$$\begin{aligned} p(M|\underline{x}) &= \prod_{i=1}^{N_M} p(\underline{m}_i|\underline{x}) \\ &= \prod_{i=1}^{N_M} \mathcal{N}(d_i; 0, \sigma_d^2) , \end{aligned} \quad (10)$$

where we again assume isotropic Gaussian measurement noise with variance σ_d^2 and independence between all measured pixels. A visual interpretation of this likelihood is given in Fig. 4 where each shaded orange line represents one of the Gaussians in (10). Note that this model ignores the width of the line segments which can be compensated by increasing the noise variance [18].

VI. Tracking Example

In this section, we discuss implementation details for the proposed tracking algorithm and present the results of our first experiments. We consider the scenario shown in Fig. 1, where a tablet (HTC Nexus 9) is to be tracked in front of a shelf. The shelf is equipped with two unique QR codes ($15.5 \times 15.5\,\mathrm{cm}^2$) and nine line segments of yellow tape.

A. Implementation

For estimation, we implemented an extended Kalman filter (EKF) using the nonlinear estimation toolbox [19] and the respective generative models of the marker likelihoods. The required uncertainties were set to roughly approximated values. For the IMU, we collected a sequence of sensor data for 10 s, while the tablet was lying still, and then calculated the sampling variance. From this procedure, we obtained variances in the magnitude of $\sigma_t \approx 10^{-2}\,\mathrm{m}^2/\mathrm{s}^4$, and $\sigma_r \approx 10^{-4}\,\mathrm{rad}^2/\mathrm{s}^2$ for accelerometer and gyroscope, respectively. In order to compensate for unmodeled behavior, we increased these values by one order of magnitude. The pixel variances of the markers were set to $\sigma_q^2 = 10^{-1}\,\mathrm{px}^2$ and $\sigma_d^2 = 1\,\mathrm{px}^2$. Initialization of the pose was implemented based on the first QR code measurement using the *solvePNP*-function from OpenCV [20]. The velocity was set to $\underline{0}$. The initial state covariance matrix was set to $10^{-5} \cdot \mathbf{I}$.

The 3D geometry of the features was carefully measured manually, yielding the 3D model shown in Fig. 1b. Intrinsic calibration of the Nexus 9 back camera was performed using the OpenCV [20] calibration routine. The ZBar library [21] was used for extracting the QR code ids and corner pixels in the images. The yellow tape was extracted from the images using a simple threshold segmentation in HSV color space. In addition, we removed outliers according to their distance to the predicted state mean.

B. *Experiment*

For evaluating the algorithm, we recorded a 35 s test sequence of synchronized RGB-video-stream with $1920{\times}1080@30\,\mathrm{Hz}$ and $\sim$100 Hz IMU data, while moving the tablet infront of the shelf. Besides several periods with fast motion, we included three challenging time intervals, where the shelf was out of the camera's field of view at 12 s, 15.5 s, and 19 s. In order to simulate different instances of the experiment from this dataset, we did not use all ambiguous marker measurements, but rather a randomly sampled subset of 200 pixels per time step. The following results are obtained from 100 runs.

C. *Results*

For a selected time interval, the average values of the estimated parameters are visualized in Fig. 5 using three horizontally aligned axes. The first and second axes include curves for the rotation and position, together with hulls which indicate their 3σ-boundary over all runs. Time steps with available marker measurements are indicated in the third axes. Fig. 6 shows the augmented view on the shelf for selected time steps, where the estimates of all runs are drawn together in each image.

From these figures, we can conclude: first, the visual quality of the estimates demonstrates the suitability of the proposed tracking algorithm for AR-applications. Even in periods with fast motion, e.g., in Fig. 6 (c,h,i), the variation of the estimated parameters is only marginal. Second, the variance only increases after periods, where no markers were visible, which can be visually verified in Fig. 5 and Fig. 6 (d-f). Nevertheless, it is surprising how fast dead reckoning lets the quality degrade when no image measurements are available. In consequence, the density of markers should be high enough to minimize these periods. Third, as can be seen, there are almost no QR code measurements available in the depicted time interval. This can be explained by looking at Fig. 6, where many images are subject to heavy motion blur. Unique markers require a high degree of user cooperation, as their detection heavily depends on the image quality. In consequence, their most common application is for (re)-initialization.

In sum, the proposed approach shows a very robust behavior and a visual assessment confirms its suitability for AR-applications.

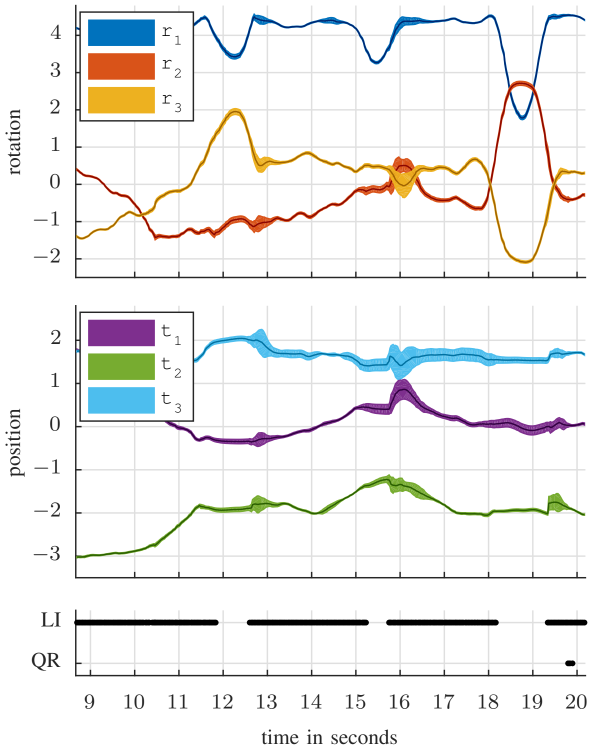


Fig. 5: Average values and 3σ-bounds of the estimated rotation/position parameters (over 100 runs) for a selected time interval. LI and QR denote time steps, where measurement updates based on line segments and qr codes are performed, respectively.

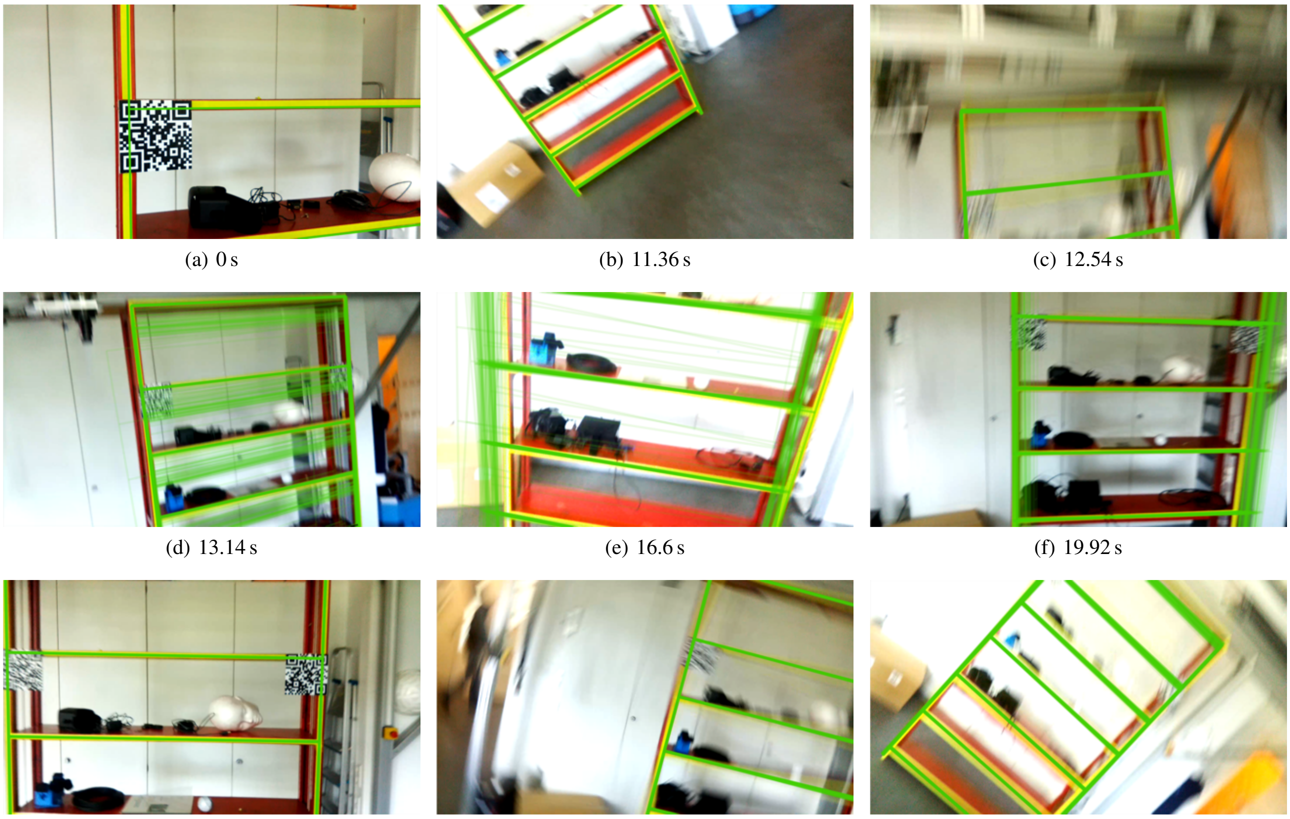
VII. CONCLUSIONS

In this paper, we proposed an approach to track the pose of a mobile device in a known environment, which is equipped with QR codes and (artificial) line segments. For the tracking algorithm, we interpreted the scene geometry as extended objects with a known shape and applied modeling techniques from the field of extended object tracking. The unique QR codes correspond to objects with known one-to-one association of measured pixels to their 3D sources, while the ambiguous line segments correspond to objects with unknown one-to-many association.

In experiments, we showed that our tracking approach is suitable for the use in AR-applications and is robust enough to deal with fast motion and poor image quality. Subsequent development will include improvements such as a better propagation of the sensor uncertainty, and optimization of the marker configuration. In addition, we aim at switching to an infrared camera together with retroreflective markers, in order to increase robustness of segmentation.

ACKNOWLEDGMENT

This work was supported by the German Federal Ministry for Economic Affairs and Energy (BMWi). Finally, we want to thank Achim Langendörfer, Oliver Acatay and Andreas Specker for supporting the experiments.



(a) 0 s

(b) 11.36 s

(c) 12.54 s

(d) 13.14 s

(e) 16.6 s

(f) 19.92 s

(g) 23.87 s

(h) 34.73 s

(i) 35.43 s

Fig. 6: Frames of test sequence, (a) initialization, (b,c) track kept with low variance under fast motion, (d-f) higher variance shortly after turn-away periods, (g-i) track kept with low variance under fast motion.

References

[1] H. Glockner, K. Jannek, J. Mahn, and B. Theis, "Augmented reality," *DHL Customer Solutions & Innovation*, 2014.

[2] B. Furht, *Handbook of Augmented Reality*, 2011, vol. 53.

[3] B. Schwerdtfeger, "Pick-by-Vision : Bringing HMD-based Augmented Reality into the Warehouse," phd thesis, TU Munich, 2009.

[4] P. Stelzer, D. J, and M. Grablechner, "Method and apparatus for visual support of commission acts," 2014.

[5] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, sep 2015, pp. 298–304.

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[7] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *IEEE International Conference on Rehabilitation Robotics*, Zurich, Switzerland, 2011.

[8] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based Localization: Observability Analysis and Consistency Improvement," *Ijrr*, vol. 33, no. 1, pp. 182–201, 2013.

[9] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," *15th international conference on Multimedia*, no. c, pp. 357–360, 2007.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, jun 2008.

[11] F. Steinbruecker, J. Sturm, and D. Cremers, "Real-Time Visual Odometry from Dense RGB-D Images," in *Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV)*, Barcelona, Spain, 2011.

[12] A. Laurentini, "Visual hull concept for silhouette-based image understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 150–162, 1994.

[13] S. Stančin and S. Tomažič, "Angle estimation of simultaneous orthogonal rotations from 3D gyroscope measurements," *Sensors*, vol. 11, no. 9, pp. 8536–8549, 2011.

[14] Z. Chen, "Bayesian Filtering : From Kalman Filters to Particle Filters, and Beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[15] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part I: Dynamic Models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, oct 2003.

[16] X. S. Gao, X. R. Hou, J. Tang, and H. F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[17] F. Faion, A. Zea, M. Baum, and U. D. Hanebeck, "Bayesian Estimation of Line Segments," in *8th IEEE Sensor Array and Multichannel Signal Processing Workshop*, Bonn, Germany, 2014.

[18] A. Zea, F. Faion, M. Baum, and U. D. Hanebeck, "Tracking Simplified Shapes Using a Stochastic Boundary," in *Proceedings of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2014)*, A Coruna, Spain, 2014, pp. 221–224.

[19] J. Steinbring, "Nonlinear Estimation Toolbox." [Online]. Available: https://bitbucket.org/nonlinearestimation/toolbox

[20] G. Bradski, "The OpenCV Library," *Dr Dobbs Journal of Software Tools*, vol. 25, pp. 120–125, 2000. [Online]. Available: http://opencv.willowgarage.com

[21] J. Brown, "ZBar bar code reader," 2012.