ELSEVIER

Journal logo

Robust detection and tracking of annotations for outdoor augmented reality browsing

Elsevier use only: Received date here; revised date here; accepted date here

Abstract

A common goal of outdoor augmented reality (AR) is the presentation of annotations that are registered to anchor points in the real world. We present an enhanced approach for registering and tracking such anchor points, which is suitable for current generation mobile phones and can also successfully deal with the wide variety of viewing conditions encountered in real life outdoor use. The approach is based on on-the-fly generation of panoramic images by sweeping the camera over the scene. The panoramas are then used for stable orientation tracking, while the user is performing only rotational movements. This basic approach is improved by several new techniques for the re-detection and tracking of anchor points. For the re-detection, specifically after temporal variations, we first compute a panoramic image with extended dynamic range, which can better represent varying illumination conditions. The panorama is then searched for known anchor points, while orientation tracking continues uninterrupted. We then use information from an internal orientation sensor to prime an active search scheme for the anchor points, which improves matching results. Finally, global consistency is enhanced by statistical estimation of a global rotation that minimizes the overall position error of anchor points when transforming them from the source panorama in which they were created, to the current view represented by a new panorama. Once the anchor points are redetected, we track the user's movement using a novel 3-degree-of-freedom orientation tracking approach that combines vision tracking with the absolute orientation from inertial and magnetic sensors. We tested our system using an AR campus guide as an example application and provide detailed results for our approach using an off-the-shelf smartphone. Results show that the re-detection rate is improved by a factor of 2 compared to previous work and reaches almost 90% for a wide variety of test cases while still keeping the ability to run at interactive frame rates.

Keywords: Augmented reality; Annotation; Tracking; Mobile phone;
PACS: the PACS codes can be found at the home page of NIMA (left column, under Contents Services):
http://www1.elsevier.com/homepage/sak/pacs/homepacs.htm

1. Introduction

Augmented Reality (AR) browsers are a new class of outdoor AR application intended for smartphones. The core function of an AR browser is simply to display mostly textual annotations that are registered to places or objects in the real world used as anchor points and are given as absolute global coordinates. Current commercial solutions rely on non-visual

sensors of the smartphone, namely GPS, magnetometer and linear accelerometers [22] [13], to determine where annotations should appear in the camera image.

However, performance of these sensors is poor. Magnetometers suffer from noise, jitter and temporal magnetic influences, often leading to deviations of tens of degrees in the orientation measurement. Even if we assume sufficient positional accuracy from GPS, which may often not be the case for consumer-grade devices in densely occluded urban environments, large orientation deviations imply that annotations will simply appear on the wrong location.

A smartphone's built-in camera allows attacking the localization problem by computer vision. However, visual detection and localization in outdoor scenes is still challenging, since it must address temporal variations such as large illumination changes. This problem is exacerbated by the fact that the coverage of the environment with reference views may be very unbalanced, and that the limited computational power of smartphones restricts the techniques that are applicable in practice. AR browsing also requires that annotations stay registered after the initial detection, which requires not only one-time detection but also real-time tracking even under fast motions. The challenge of meeting all these requirements simultaneously has limited the generality of previous outdoor AR tracking solutions on smartphones.

For an improved user experience, we can exploit the characteristics of smartphones and the AR browsers running on them. On the one hand, smartphones allow fusion of camera and non-visual sensors. On the other hand, AR browsers are usually operated while the user is standing still and only performing rotational movements. Previous work exploits this rotational motion to generate panoramas on the fly and then use them for vision-based orientation tracking [25]. Later work extended the panorama creation in a way that allowed users to annotate objects within the panorama. These annotations can be shared with other users visiting the same spot as annotations anchor points were redetected in newly created panoramas by matching small image patches [12]. These tasks – panoramic mapping and matching of annotations anchor points – can be carried out simultaneously in real time, leading to an uninterrupted user experience. This paper presents an enhanced approach, which significantly improves the performance of both re-detection and of tracking over the basic system (summarized in section 3). Panoramas are created with an extended dynamic range representation, which can better represent the wide variety of illumination conditions found outdoors (section 4.1). The internal orientation sensors are used to prime an active search scheme for the anchor points, which improves the matching results by suppressing incorrect assignments (section 4.2). Finally, global consistency is enhanced by statistical estimation of a global transformation that minimizes the overall position error of anchor points when transforming them from the source panorama in which they were created, to the current view represented by a new panorama (section 4.3). This step considers multiple hypotheses for association of anchor points to known candidates, and as a result further suppresses wrong associations. Once the anchor points are redetected, we track the user's movement using a novel 3-degree-of-freedom orientation tracking approach that combines vision tracking with the absolute orientation from inertial and magnetic sensors (section 5). This fusion improves tracking performance even under fast motion and tracking failures and provides important input for initialization of the visual tracking component.

We tested our system using an AR campus guide application as a test case and provide detailed results for our approach using an off-the-shelf smartphone (section 6). Results show that the re-detection rate is improved by a factor of 2 by the enhancements reported in this paper and reaches almost 90% for a wide variety of test cases.

2. Related work

Previous work can be roughly divided into two research directions. Firstly, work on systems allowing the user to annotate the environment using an AR interface. Secondly, work which deals with re-detection and tracking of objects in outdoor environments.

Early work displaying annotations using augmented reality were conducted by Feiner et al. as

in the MARS project [7]. This approach can be seen as the conceptual origin for the recent development of commercial AR-browser applications running on smartphones such as Wikitude [13] or Layar [22]. These commercial systems present annotations from databases that were created offline and positioned using GPS references. In contrast, some recent research work deals with placing annotations online, within the AR application.

A number of approaches exist for this online annotating. For example, Reitmayr et al. [16] used an existing 3D model of the environment to calculate the exact position of the annotation by casting a ray into the scene. Later Reitmayr et al. [19] described a set of techniques to simplify the online authoring of annotations in unknown environments using a simultaneous localization and mapping (SLAM) system.

The approach presented by Piekarski and Thomas [15] uses triangulation for placing annotations. Rays are cast from different positions in the environment into the direction of the annotation and then intersected.

Wither et al. [26] used aerial photographs to support the annotation process. After casting a ray into the direction of the object to be annotated in the AR view, a secondary view shows an aerial photograph, allowing the user to move the annotation along the ray. Later Wither replaced this manual placement along a ray with a single-point laser range finder [27].

The work in [12] proposed another method allowing the user to place annotations in a panoramic view of the environment. This technique, which is the foundation for this paper, is further summarized in section 3. The main drawback of this technique is its poor detection performance under strong temporal variations.

Several previous PC-based outdoor AR systems rely on a combination of vision, GPS and inertial measurement unit (IMU) sensors to obtain a global 6DOF registration within the earth reference frame [7][23]. These sensors have recently also become available in smartphones, but the inexpensive, low-power MEMS devices used in smartphones perform poorly compared to dedicated industrial sensors used in previous larger AR setups.

In all these devices GPS provides 3D positional information, while orientation is estimated from linear accelerometers (measuring the local gravity vector) and magnetic compasses (measuring the local magnetic field vector). Typically, electromagnetic fields and conductive materials in both the environment and the hardware setup itself distort a magnetometer's measurement. Azuma et al. [3] provide an insightful description of the performance of such sensors and the resulting significant registration errors, especially if annotated objects are far away [4].

Several approaches exist to overcome the inherent limitations of using sensors alone. Careful calibration of the magnetic sensors' scale, bias and non-orthogonal parameters, as well as influences such as hard- and soft-iron effects in close proximity, can reduce the deviations between measurements and the true magnetic field vector. Calibration can be based on the assumptions of measuring the same vector under different orientations [29], measuring invariants of a setup such as the angle between the north vector and gravity vector [10], or manual calibration using measurements in relation to ground-truth [3]. However, in many cases a one-time calibration is not sufficient, as the errors change with time and location. Therefore online calibration methods [8] are required to adapt to varying distortions. The hybrid orientation tracking presented in section 5 can be seen as a kind of online calibration.

Camera-based tracking methods can provide higher accuracy and update rate than pure non-visual sensor-based systems, but they usually rely on a model of the environment. Here, the device's pose is measured in relation to the model using visual features [24]. Klein and Murray [11] presented a SLAM-based tracker that builds the model of the environment on the fly but only works in small workspaces. Arth et al. [2] presented a method for localizing a mobile user's 6DOF pose in a wide area using a sparse 3D point reconstruction and visibility constraints. It is well known that fusion of vision with non-visual sensor data allows for more robust performance under fast motion and tracking failures [17][20][28] and provides important input for initialization of the visual tracking component [6][18]. However, little research work on tracking

with sensor fusion on smartphones has been done to date, possibly because of poor sensor quality, limited computational power or the relatively recent availability of sensor-equipped smartphones.

In unknown environments, visual tracking cannot provide absolute measurements, but it can provide constraints that allow calibrating sensors online. Azuma et al. [5] used relative rotation measurements obtained through 2D feature tracking to learn the distortions in a magnetic compass. In earlier work [21], we looked at overcoming short-term distortions through tracking the difference between vision-based orientation tracking and a compass. Any significant change to this difference over time was interpreted as a failure of one subsystem, and the system logic consequently switched to the more reliable one. However, this scheme did not allow for compensating an initial distortion in the magnetic sensor. The approach described in this paper estimates the difference over time and can therefore reduce larger distortions in the compass.

3. Panoramic augmented reality annotations

In the following we will briefly introduce our previous work on panoramic mapping and tracking as the system described in this paper is based on a panoramic map of the environment, created in a simultaneous mapping and tracking step and used for continuous real-time orientation tracking. We further give an overview on our previous work of using the panorama as an intermediate representation of the environment on which template-based matching of annotations anchor points is performed as a background activity. Concurrently the orientation tracking allows real-time updates to the AR user interface used for displaying the annotations.

3.1. Panoramic mapping and tracking

The panoramic mapping and tracking is based on the assumption that the user performs only rotational movements with the camera phone at an annotated spot, while translational movements can be neglected. The user's position is determined with GPS. This assumption allows the current camera frame to be mapped incrementally onto a cylinder to create a 2D environment map (see Figure 1).

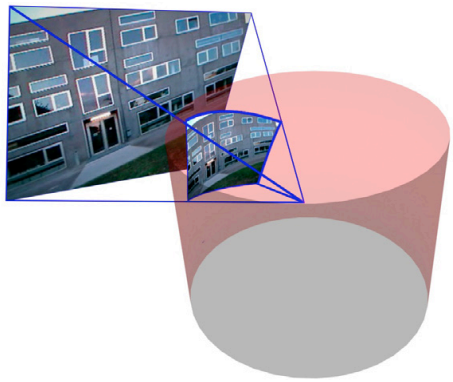

Figure 1. Projection of the camera image onto the cylindrical map.

Identifying and processing only those parts of the current camera frame, which are not yet mapped, helps to increase the speed of this algorithm, as only a few (usually <1000) pixels have to be mapped per frame.

After updating the panoramic map, the algorithm computes the rotational tracking information for each frame. This step employs an active search scheme together with a motion model assuming constant motion. FAST keypoints [14] are extracted at each frame for the current camera frame and compared against the keypoints in the current panoramic map. To compute the FAST keypoints on the unfinished panoramic map, the map is divided into tiles. If all the pixels within a tile are mapped, the tile is considered finished. Finished tiles are searched for FAST keypoints in a background thread. The available keypoints are then used for updating the tracking information. The full algorithm of panoramic mapping and tracking is running in real-time at 30 fps on current generations smartphones such as the HTC HD2 making the panorama generation only dependent on how fast the user captures the environment by rotating the camera. A more detailed overview of the implemented approach and timings are given in [25].

3.2. Template based annotation matching

In [12] we present an approach, which uses the panoramic representation of the environment to augment the live view with annotations. The system determines the position of an annotation by using an image patch stored on a remote server. As soon as a new user approaches an annotated spot, the application downloads all image patches of annotated panoramas in the close proximity and matches them against new panorama while this is produced using the algorithm described in 3.1. The matching itself relies on normalized cross correlation (NCC). To avoid excessive matching against the full panorama at each frame, the matching is scheduled to only test finished tiles, which were also used for creating the keypoints as described in 3.1. Consequently, each panorama tile is only tested once against the list of image templates.

Another speed up is achieved by using a hierarchy of tests. A Walsh transform is computed as a pre-check before applying the more expensive template matching using NCC. This reduces the numbers of NCC operations, as only the cases that pass a threshold when matching Walsh transforms are tested with NCC.

Matching the annotation templates against the map rather than the camera image allows us to schedule the matching to guarantee a desired frame rate: Each finished tile is not checked immediately, but put into a queue instead. During each frame, the system schedules only as much work from the queue as allowed by the given time budget. Since the operations are simple and their timings are predictable, we can easily limit the workload so that the time budget is not exceeded.

Our system can therefore run at constant speed on any phone that is able to perform real-time panoramic mapping and tracking. On fast phones, annotations are detected quickly, whereas on slower phones it takes longer. Matching one cell against 12 annotations takes ~28ms on an HTC HD2. Targeting a frame rate of 20Hz (50ms per frame) allows scheduling ~10ms for detection of every frame. Figure 2 shows the workflow of the system presented in [12]: Peter starts by creating a panoramic map and labels objects of interest. The annotations, Peter's GPS location and a description of the visual

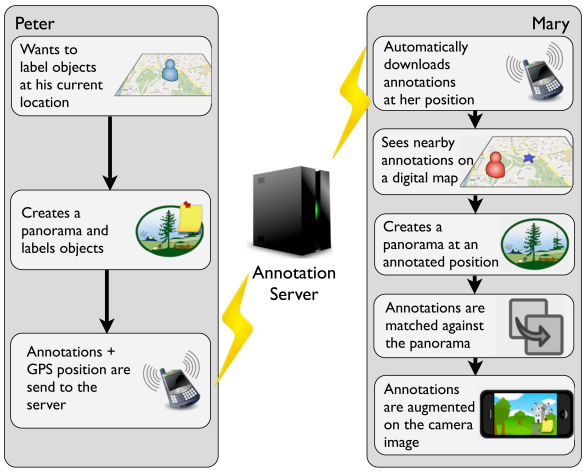

Figure 2. The workflow of the panoramic AR annotation system involves two users – Peter creates annotations and at a later time Mary browses through these annotations.

appearance of the annotated area are transmitted to a server. Later, Mary wants to retrieve annotations authored by Peter. Her phone notifies her when she is close to locations annotated by Peter, using GPS information. A map view allows her to reach a spot close to where Peter was when he created the annotations. After pointing up, the phone uses a newly created panorama for efficiently matching Peter's annotations to the environment. Mary's phone displays the corresponding annotation, as soon as the supporting area of a particular annotation is re-detected. Mary is now able to create additional annotations herself.

The main drawback of this approach is that it relies entirely on the vision-based matching, and is therefore susceptible to temporal variations such as shadows or vegetation changes. Furthermore, the template matching is carried out on the whole panorama without using prior knowledge to optimize the search area. All annotations are treated independently, which means that the position resulting from an earlier matching process does not assist later matches.

All this resulted in matching scores, which are very good (about 90%) for searching annotations under the same environmental conditions (position of the sun, weather conditions). However, if environment conditions are different, the matching

scores in tests dropped by almost a factor of 2 (about 56%). This is clearly not sufficient for using this approach in social computing application, where annotations should be reliably reproduced for extended periods of time.

4. Enhanced re-detection of annotations in panorama maps

The focus of this work is the improvement of the low re-detection rate of annotations in the case of different environment conditions by keeping the general workflow as presented in 3.2. Thus the users are guided to a spot containing previously created annotations using GPS. While the users points his phone to the environment the system creates a panorama, which we use simultaneously to detect the annotations. The differences against previous work are three improvements for the stability of re-detection.

Firstly, the basic quality of the panoramic map must be enhanced, so that later matching can tolerate stronger deviations in appearance. This requires capturing more information in the original panorama, which is achieved by deploying an extended dynamic range representation of the map.

Secondly, commonly available smartphone hardware is exploited more consequentially. Non-visual sensor measurements are used to narrow down the search area of the vision-based re-detection. Moreover, the sensor-based tracking is used as a backup in case the vision-based system fails.

Thirdly, we estimate a global transformation T, which aligns the source panorama and the target panorama, using reliable statistical techniques. By applying T, we can map all annotations stored on the server corresponding to a source panorama to a newly created target panorama. In the following, a detailed description of these steps is given.

4.1. Extended dynamic range panoramic maps

The basic template matching of image patches describing annotations and the panoramic map is strongly dependent on the image quality of the panoramic map.

A main problem in this process is the automatic adjustment of exposure and white balance of built-in cameras in current generation smartphones. The camera chip performs arbitrary processing to deliver a "nice" image, without letting the application programmer control or even understand the process. While this automatic image processing seems to have no strong effect towards the tracking and therefore does not adversely affect the stitching success, it results in visible boundaries in the panoramic map (see Figure 3), where contributions from multiple frames are stitched together. These patches show discontinuities in brightness caused by variations in the exposure settings. Later in the matching, the discontinuities introduce artificial gradients, which heavily affect the template-based matching of the anchor points. The situation is made worse by the fact that discontinuities can appear both, in the image patches describing the annotations, which are extracted from the panoramic map, and in the newly created panoramic map used for re-detecting the annotations.

The best solution to suppress such discontinuities caused by exposure changes would be to use a camera that allows the programmer to fix the exposure rate. Such a programmable camera would even provide the possibility to create true high dynamic range images, if the response function could be determined for the integrated camera. However, to the best of our knowledge, the only mobile device capable of controlling camera parameters is the Nokia N900 with Frankencam API [1]. It seems unlikely that fully programmable cameras will become widespread in the foreseeable future.

Thus we created a different approach that allows the creation of extended dynamic range (EDR) images on phones without any access to the exposure settings. While this approach must necessarily rely on simple estimation, it can compensate for the most severe artefacts introduced by auto-exposure. For this purpose, we map the first camera frame into the panoramic map and use the pixel intensities as a baseline for all further mappings. All subsequent frames are heuristically adjusted to match the intensities found in the first frame, by estimating the overall change of the exposure setting between the first and the current frame.

Figure 3. (Top) A panorama image containing visual artefacts, which are caused by the automatic and continuous exposure adjustment of current mobile phone cameras. (Bottom) A panorama image that was created by extending the dynamic range during the mapping into the panorama and applying a tone mapping afterwards.

We achieved this by using the FAST keypoints, which are computed in the current camera frame and the panoramic map. As these keypoints are already generated for tracking purposes (see 3.1), this step does not generate an additional overhead. We compute the difference of intensities for all pairs of matching keypoints found in the camera frame and in the panoramic map. The average difference of these point pairs is used to correct the current camera frame by adding the difference to each pixel before mapping it into the panorama. This simple correction significantly reduces the discontinuities of intensities. The panoramic map is built using 16 bits per color channel, which was empirically found to be sufficient to avoid any clipping errors when adjusting pixel values in the described way, without consuming too much memory bandwidth for a smartphone. The display of the panoramic map with extended dynamic range is done with a simple linear tone-mapping operator. A resulting panorama image is showed in Figure 3. As it can be seen, discontinuity artefacts are noticeably reduced, which is confirmed by our experimental results.

4.2. Sensor fusion for improved re-detection

Current generation smartphones regularly include GPS, compass, accelerometer and recently even miniature gyroscopes. The accuracy of these sensors is usually inferior to a well-tuned visual tracking technique, but non-visual sensors are complementary because of their robust operation. We therefore integrated the compass and the accelerometers to create a better re-detection of annotations.

The improved re-detection is achieved by narrowing down the search area for the vision-based template matching using the information obtained from the internal sensors. The region in the panorama where the annotation is likely to be located-based is determined based on a direction estimate from the internal sensors.

The panoramic map is created at a resolution of 2048x512 pixels from 320x240 pixel sized camera

Figure 4. (Bottom) A source panorama that was used to create the annotations. (Top) A newly created panorama with the best candidates for placing the annotation resulting from the template-based matching. For every annotation anchor point we store a maximum of three best matches. The green dots in the upper image have the best matching scores and are therefore used for label placement. The red ones are the second and third best matches of an annotation, which makes them a candidate for a possible correct match.

images. A typical camera has a field of view of ~60°, so the camera resolution is close to the map resolution: 320 pixels / 60° · 360° = 1920 pixels. The theoretical angular resolution of the map is therefore 360° / 2048 pixels = 0.176 degrees per pixel. Assuming a maximum error of the compass of ±10° we can expect to find the annotation in a window of ±57 pixels around the estimated position. We consider an area 3 times larger that this window, but weight the NCC score with a function that penalizes by distance from the active search window. Thus we only consider matches outside the primary search area if they have a very good matching score.

4.3. Matching annotations using a global transformation

In the previous approaches, the annotations were considered independent of each other during the re-detection. Thus, the detected position of an annotation was not used to optimize the re-detection of other annotations. Moreover, empirical analysis revealed that the main reason for wrong results from the NCC template matching came from more than one good match for one annotation (see Figure 4). This led to the problem that single annotations could not be detected reliably or were detected at the wrong location, whereas other annotations were robustly detected at the correct spot. This situation calls for additional geometric verification.

We approach the problem by considering the annotations in the source panorama (the panorama which was used to create the annotations) as a set for which a consistent geometric estimate must be achieved. Therefore, the detection is extended by the requirement to find a global transformation T, which maps the set of annotations from the source panorama into the target panorama (representing the current environment) with a minimized average error. As we assume the panoramas to be made at the same position, the transformation is a pure rotation aligning source and target panorama with three degrees of freedom.

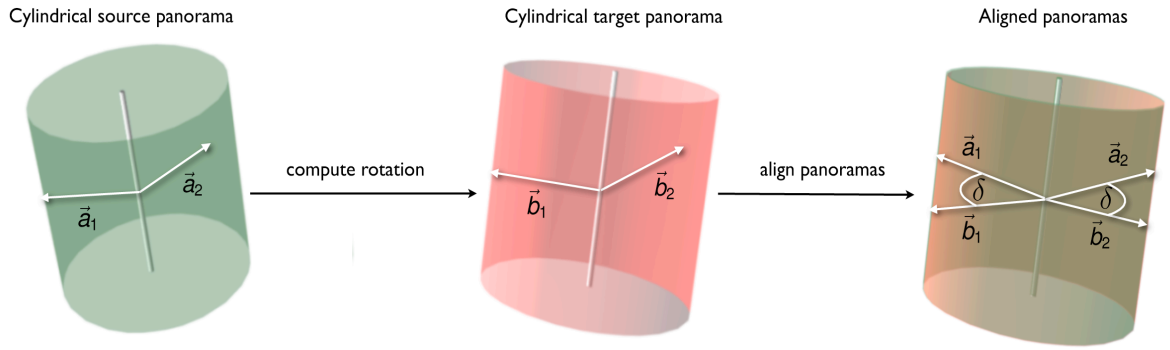

Figure 5. Illustration describing the alignment of two cylindrical mapped panoramas based on the position of the annotations anchor points. The two vectors $\vec{a}_1$ and $\vec{a}_2$ are pointing to two annotation positions in the cylindrical source panorama. The middle cylinder describes a panorama, which is created on the fly on the smartphone. The vectors $\vec{b}_1$ and $\vec{b}_2$ are pointing to two possible annotation positions in this new panorama. Rotating one cylinder into the other in order to align both vectors of each cylinder using absolute orientation with an error δ, results in a rotation, which can be used in a RANSAC calculation to determine a model with a sufficient small error.

To compute rotation *T*, we describe the position of an anchor point in the source panorama by representing anchor coordinates as a 3D vector from the camera position to a point on the cylindrical panorama (see Figure 5). We extended the workflow as presented in section 3.2 to also store this 3D vector together with the image patch for each annotation. This dataset describing the annotation is uploaded to a remote server and tagged with the GPS address of the current position as depicted in Figure 2. We do not upload any panoramic image, as only this dataset is required to redetect the annotations. As the size of the dataset is in the range a few kilobytes (~2 kilobytes for the image patch + text information) it can be easily handled via a 3G connection.

Once a user approaches a place where annotations were created, the mobile phone accesses the closest datasets based on the GPS position. We take into account that GPS can be inaccurate and therefore we download all datasets that were created within proximity of 50m. After downloading the datasets the anchor points are redetected using the template-based matching and annotations are initially placed using the best match. But instead of using only the best match, we also keep the best three candidate matches based on NCC score for later use. For all found candidate matches, we compute the vector-based position in the target panorama as we did for the original annotations in the source panorama.

While online tracking and mapping continues, a RANSAC based approach running in a background thread determines and updates a global rotation *T*. This rotation aims to optimally map the set of all annotations from the source panorama to the target panorama by aligning the panoramas.

We randomly select two annotations and one of their three best candidate positions in the target panorama as input for finding the best rotation using RANSAC. To find the best match, the rotation *T* between the two coordinate systems is calculated so that two vector pairs $\vec{a}_1$, $\vec{a}_2$ and $\vec{b}_1$, $\vec{b}_2$ can be aligned to each other while minimizing an L^2 norm of remaining angular differences. We use the absolute orientation between two sets of vectors [9] to compute this rotation. The resulting rotation is the hypothesis for the RANSAC algorithm. All annotations are mapped to the target panorama using the current estimate for *T*, and the difference of the resulting 2D position in target map space to the annotation position found through template matching is determined. If the distance is below a threshold, the annotation is counted as inlier and its error is also counted as inlier. Its error is then added to an error score.

For a hypothesis with more than 50% inliers, a normalized error score is determined by dividing the raw error score by the number of inliers. The normalized score determines if the new *T* replaces the previous best hypothesis. This process is repeated until a *T* with an error score below a certain threshold is found. Such a *T* is then used to transform all annotations from the source to the target panorama. Annotations for which no successful match could be found can now also be displayed at an appropriate position, although with less accuracy because their placement is only determined indirectly.

Obviously, the source and target panorama are never taken from the exact same position, and the resulting systematic error can affect the performance of the robust estimation. We empirically determined that a 50% threshold for inliers and a 10 pixel threshold for the normalized error score in 2D map coordinates yields a good compromise between minimizing overall error and reliable performance of the RANSAC approach.

Finding the best rotation to align the two panoramas requires about ~30ms for 8 annotations but the panoramas are not aligned each frame, as it is only necessary to update the model once new candidates for annotations anchor points are detected based on the vision-based template matching.

5. Hybrid orientation tracking

Once we have redetected the anchor points of the textual annotations, we need to track orientation changes to guarantee a continuous precise augmentation of the annotations in the users current view. The re-dection using the absolute orientation as described in section 4.2, requires measurements from the magnetic compass and linear accelerometers to estimate the absolute orientation of the device, because the vision-based tracking only estimates orientation with respect to an arbitrary initial reference frame. Moreover, the vision-based orientation tracking has difficulties in dealing with fast motion, image blur, occlusion and other visual anomalies. On the other hand, the vision-based tracking is more accurate than the sensor-based orientation estimate. Therefore, we fuse the two orientation measurements to obtain a robust and accurate orientation.

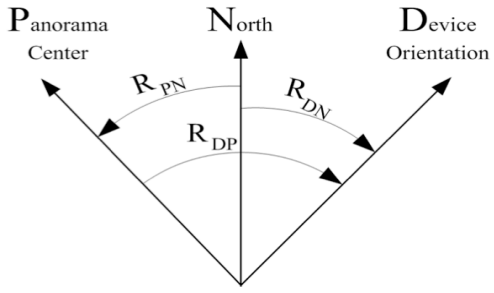

Figure 6. Overview of the rotations between world reference system *N*, device reference system *D* and panorama reference system *P*.

In principle, the vision-based tracking would be sufficient for accurate orientation estimation, but it only provides relative measurements. Therefore, we use the sensor-based orientation to estimate the global pose of the initial reference frame of the vision-based tracker and then apply the incremental measurements to this initial and global pose. A first estimate can be obtained through simply reading the sensor-based orientation at the same time the vision-based tracker is initialized.

However a single measurement of the sensor-based orientation will be inaccurate. Therefore, we continuously refine an online estimation of the relative orientation between the initial vision-based tracking frame and the world reference frame.

We assume a north-oriented world reference frame *N* given locally by the direction to magnetic north and the gravity vector. The inertial and magnetic sensors measure the gravity and magnetic field vectors relative to a device reference frame *D*. The output of the sensors is a rotation R_{DN}[1] that maps the gravity vector and the direction of north from the world reference frame into the device reference frame (see Figure 6).

The visual orientation tracker provides a rotation of the device R_{DP} from the reference frame *P* of the

[1] The subscripts in R_{BA} are read from right to left to signify a transformation from reference frame A to reference frame B.

panorama into the device reference frame *D*. In principle, the device reference frame differs for camera and other sensors. For simplicity, we assume a calibrated device in the following, for which the two reference frames can be considered identical.

Our aim is to estimate the invariant rotation R_{PN} from the world reference frame *N* to the panorama reference frame *P* (see Figure 6). Composing the rotations from world to panorama to device reference frame, we obtain

$$R_{DP} \cdot R_{PN} = R_{DN} \quad (1)$$
$$R_{PN} = R_{DP}^{-1} \cdot R_{DN} . \quad (2)$$

Using equation (2) we can estimate the relative rotation R_{PN} from simultaneous measurements from the vision-based and the sensor-based tracking. At every timestamp *t*, we record measurements g_t for the gravity vector *g* and m_t for the magnetic field vector *m,* both *g* and *m* defined in the world reference frame. A rotation $R_{DN} = [r_x\ r_y\ r_z]$ is calculated such that

$$g_t = R_{DN} \cdot g \text{ , and} \quad (3)$$
$$m_t \cdot r_z = 0 . \quad (4)$$

The resulting rotation accurately represents the pitch and roll measured through the linear accelerometers, while the magnetic field vector may vary within the plane of up and north direction (X-Y plane). This reflects our observation that the magnetic field vector is noisier and introduces errors into the roll and pitch of the device. For the video frame available at timestamp *t*, our vision tracker provides a measurement of the rotation R_{DP}. Given the two measurements R_{DN} and R_{DP}, we can compute R_{PN} through equation (2). To filter repeated measurements of R_{PN}, we use an extended Kalman filter (EKF) operating on the rotation R_{PN}.

To represent the filter state, we model rotations with 3 parameters using the exponential map of the Lie group SO(3) of rigid body rotations. The filter state at time *t* is an element of the associated Lie algebra *so*(3), represented as a 3-vector μ,. This element describes the error in the estimation of the rotation R_{PN}. μ is normal distributed with $\mu \sim N(0,P_t)$ with a fixed covariance *P*. It relates the current estimate $\hat{R}_t$ to the real R_{PN} through the following relation

$$R_{PN} = \exp(\mu) \cdot \hat{R}_t , \quad (5)$$

where exp(.) maps from an element in the Lie algebra *so*(3) to a rotation *R*. Conversely, log(*R*) maps a rotation in SO(3) into the Lie algebra. As we are estimating a constant, we assume a constant position motion model, where μ does not change and the covariance grows through noise represented by a fixed noise covariance matrix.

The measurement equation for the filter state μ states that the expected measurement equals the current rotation $\hat{R}$ and the difference is the identity rotation:

$$\log(R_{DN} \cdot \hat{R}_t^{-1}) = \mu . \quad (6)$$

The measurement Jacobian of (6) is now simply the identity matrix. This Jacobian is used in the extended Kalman filter framework to update the state μ. Finally, we correct for the new error estimate and update the current rotation $\hat{R}_t$ by left multiplying $\exp(\mu)$ to it. After this we reset the error μ again to 0.

The global orientation of the device within the world reference frame is computed through concatenation of the estimated panorama reference frame orientation R_{PN} and the measured orientation from the visual tracker R_{DP} as described in equation (1). Thus we combine the accurate, but relative orientation from visual tracking with a filtered estimate of the reference frame orientation. The implementation as a recursive filter is efficient and fast, requiring only little memory and processing power.

6. Experiments and results

We implemented and evaluated our approach on a common smartphone (HTC HD2) as part of a campus information system. During the evaluation, we focused on two main criteria: Firstly, the re-detection rate used for detecting the annotation anchor points, and secondly, the accuracy of the hybrid tracker used for tracking the orientation.

6.1. Re-detection performance

To test the re-detection performance, we created 12 panoramas at different positions on our campus,

aiming at obtaining a diverse set of images and environmental conditions. The average distance between these panoramas was ~50m. For each panorama, we created 4-6 annotations, leading to 58 annotations in total. For better comparison, we created panorama images both using the extended dynamic range approach presented in section 4.1 and using standard 8-bit dynamic range. We then proceeded to attempt matching the collected annotations against newly created panoramas resulting from the recorded video streams.

To test the matching performance under different lighting settings (see bottom Figure 7), we created the panoramas and the annotations on a sunny day one hour before sunset and tried to mach them to material from a different day taken about noon. This led to situations in which certain building parts had noticeable shadows but the annotation templates however did not show these shadows. We also collected material with lighting artifacts including lens flares and bright spots at the position of the sun, which where mapped directly into the panorama image making it very difficult to match annotations in such areas.

As our approach requires the user to be at the same position from where the annotations and the source panorama were created we evaluated the matching performance within a 2m radius to the original position. As GPS was sometimes inaccurate we had the case that at one position two annotated spots were assumed to be within 50m resulting in the fact that the application downloaded the datasets of both annotated spots and choose the one achieving the highest scores in the NCC-based template matching for further processing.

The evaluation procedure was set up so that all combinations of re-detection enhancements were systematically tested. The baseline system without any enhancements resulted in a re-detection rate of about 40%, which is less than reported in [12] because of the more difficult environmental conditions used to create the data sets. The results are summarized in Figure 7. The sensor fusion improves re-detection by about +15%, to a point where the RANSAC approach for determining the global transformation finds enough inliers, so that the combined sensor fusion and global transformation technique delivers 86% re-detection rate. The EDR

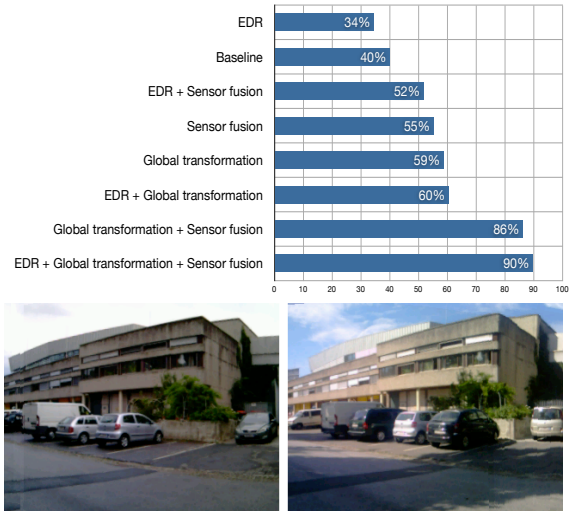

Figure 7. Re-detection evaluation. (Top) Overview of the re-detection results. (Bottom) Fragments of two panorama images showing the different environment conditions during the evaluation.

representation seems only effective in improving already very good results a bit further, while EDR applied alone on difficult situations can even slightly reduce matching performance. However, the combination of all three enhancements leads to an overall re-detection of 90%, which is more than twice the original performance and probably satisfactory for everyday operation.

6.2. Hybrid tracking accuracy

Using the sensors, we can directly calculate the 3x3 rotation matrix representing the phone's orientation. We tested the absolute accuracy of our hybrid orientation tracker using a set of reference points in the environment, which were surveyed using a professional tachyometer at centimeter level accuracy. The distance from the camera to the reference points varied from 26 to 92 meters.

For better reproducibility, the mobile device was mounted on a tripod positioned above the reference point. We measured the accuracy of the tracker by aiming the device's camera at one of the reference points and subsequently turning the device towards all other reference points without resetting the tracker. The device was kept still for about 30

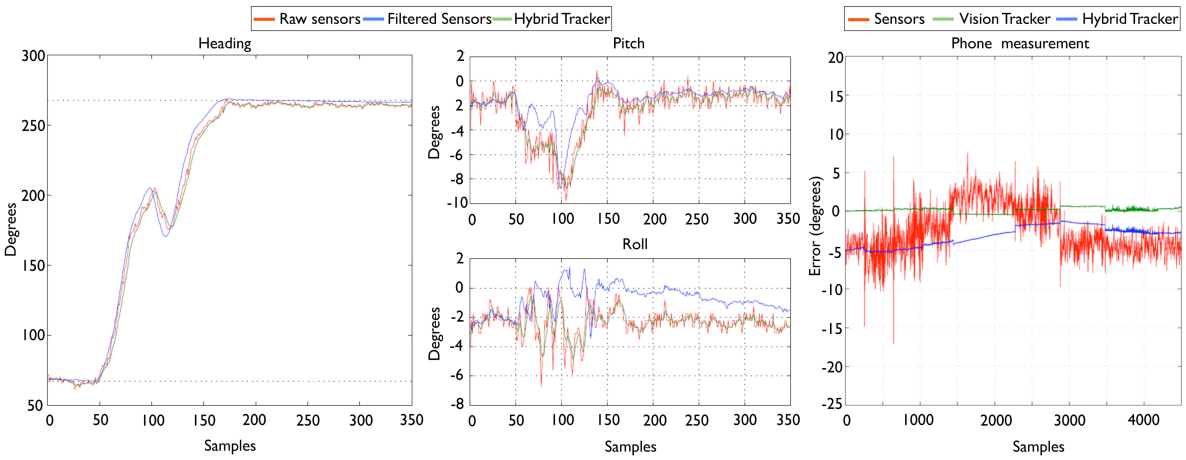

Figure 8. (Left and Middle) Plot of heading, pitch and roll for a free-hand movement of the mobile phone between two reference points. We plot orientation for the raw sensor values, a filtered estimate and the hybrid tracker. (Right) Test sequence showing the errors in degrees to the north for the sequences recorded on the phone. Visual Tracking is only shown as reference as true absolute orientation is usually not known.

seconds at each reference point, while the orientations reported by the sensors, the vision tracker and the hybrid tracker were logged.

The measurement noise used in the evaluation was derived from static observations of the sensors. As the measurement function (2) combines the two inputs, the measurement noises of sensors and visual tracker need to be combined. In practice, the visual tracker has much lower noise and is subsumed in the sensors' noise. The process noise was tuned and set to 10^4, yielding the lowest root mean square error for recorded sequences. Figure 8 (left and middle) depicts a plot of a measurement session, while the phone is turned in clockwise direction from one reference point to the next. Figure 8 (right) shows the error to the closest reference point, effectively showing the error to the ground truth heading.

The results demonstrate two improvements over pure sensor-based orientation tracking. Firstly, high frequency noise is reduced with a very small lag relative to fast motions (see Figure 8 left and middle). The visual tracking is dominating the motion estimation and provides low jitter rotation estimates. Secondly, over time, the error of the filtered rotation is smaller than the sensor-only rotation, because deviations in the compass measurements are averaged over different orientations. Overall, we obtain a responsive, less jittery estimate that is on average more accurate than the orientation derived from the sensors alone and more robust in case of fast motions.

7. Conclusions and Future Work

We presented an approach for the detection and tracking of annotations in mobile AR applications. The used approach allows users visiting the same spot to share annotations augmented in the live camera view. The annotations created by the first user are detected in the view of the second user by matching image patches against a newly created panorama of the environment. To improve the detection we narrow down the search area and apply geometric constraints. Once the annotations are detected we track the user's orientation using a reliable hybrid tracking approach allowing us to correctly augment the annotations in the live camera view. We show that the presented approach outperforms previous approaches in terms of robustness and accuracy. Combining all approaches described in this paper for improving re-detection significantly increased the re-detection rate for the

matching of annotations by a factor of 2 compared to previous work, yielding a 90% re-detection rate under strong temporal variations in the environment. Once detected, the presented sensor fusion approach is used for tracking the users orientation and significantly improves the orientation estimation quality. The approaches presented here are generally applicable to outdoor AR, but specifically improve smartphones, which have rather low quality sensors and limited computation power for computer vision.

Future work should address the problem of a more efficient representation of the anchor points. Storing patches is simple and flexible, but an encoding of the neighborhood relying on feature descriptors suitable for real-time matching may be more efficient. Unfortunately, reliable feature matching under strong temporal variations and with limited input image quality remains an open research topic. Further investigations can be done to improve the selection of the correct annotation dataset by not only using the GPS information but also using the current camera image for vision-based localization. Moreover, further investigations are needed to better understand the relationship of extended dynamic range image capturing on the re-detection results.

Other future work targets the tracker. As the visual tracker itself adds some bias as the relative orientation, estimation can overestimate or underestimate the true angle of rotation, if the focal length of the camera is not accurately known. By adding a correction factor to the filter estimate, it would be possible to estimate this bias and correct it in the final rotation output.

Finally, a purely temporal filtering of errors is not the ideal solution. The filter depends on receiving measurements under different orientations to reduce errors through averaging. Measuring errors for a longer time in a certain orientation will pull the estimate towards that orientation and away from the true average. A more accurate model should consider distribution of the orientation measurements while also weighting old measurements to account for changes over time. Together, both could form a truly dynamic online calibration method.

Acknowledgments

This work was funded by the Christian Doppler Laboratory for Handheld Augmented Reality through the Austrian Research Promotion Agency (FFG) under the contract no. FIT-IT 820922 (Smart Vidente), and through the Austrian Science Fund FWF (W1209). We thank the Stadtvermessungsamt Graz for providing the real-world test dataset.

References

[1] Adams, A., Horowitz, M., Park, S.H., Gelfand, N., Baek, J., Matusik, W., Levoy, M., Jacobs, D.E., Dolson, J., Tico, M., Pulli, K., Talvala, E.-V., Ajdin, B., Vaquero, D. and Lensch, H.P.A.The Frankencamera. ACM Transactions on Graphics, vol. 29 (2010), 1-12.

[2] Arth, C., Wagner, D., Klopschitz, M., Irschara, A. and Schmalstieg, D., Wide area localization on mobile phones, In Proc. International Symposium on Mixed and Augmented Reality (ISMAR), IEEE press (2009), 73-82.

[3] Azuma, R., Hoff, B., Neely, H. and Sarfaty, R., A motion-stabilized outdoor augmented reality system. In Proc. IEEE VR (1999), 252–259.

[4] Azuma, R.,The challenge of making augmented reality work outdoors. Mixed reality: Merging real and virtual worlds (1999), 379–390.

[5] Azuma, R., Lee, J. W., Jiang, B.,Park, J., You, S. and Neumann, U., Tracking in unprepared environments for augmented reality systems. Computer & Graphics (1999), 787–793.

[6] Coors, V., Huch, T. and Kretschmer, U., Matching buildings: Pose estimation in an urban environment. In Proc. ISAR 2000, pages 89– 92, Munich, Germany, October 5–6 2000.

[7] Feiner, S., MacIntyre, B., Höllerer, T. and Webster, A., A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In Proc. ISWC'97, (1997), 74–81.

[8] Hoff B., and Azuma, R., Autocalibration of an electronic compass in an outdoor augmented reality system. In Proc. ISAR 2000, (2000), 159– 164.

[9] Horn, B. K., Closed form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America, (1987), 629–642.

[10] Hu, X., Liu, Y., Wang, Y., Hu Y. and Yan, D., Autocalibration of an electronic compass for augmented reality. In Proc. ISMAR 2005, (2005), 182–183.

[11] Klein, G. and Murray, D., Parallel Tracking and Mapping on a camera phone. In Proc. 8th International Symposium on Mixed and Augmented Reality (ISMAR), IEEE press (2009), 83-8.

[12] Langlotz, T., Wagner, D., Mulloni, A. and Schmalstieg, D., Online creation of panoramic augmented reality annotations

on mobile phones. Accepted for IEEE Pervasive Computing, IEEE press (2010).
[13] Mobilizy. Wikitude. http://www.wikitude.org/.
[14] Ozuysal, M., Fua, P., Lepetit, V., Fast keypoint recognition in ten lines of code. In of Proc. CVPR 2007, IEEE press (2007), 1-8.
[15] Piekarski, W., Thomas, B., Interactive augmented reality techniques for construction at a distance of 3D geometry, In Proc. Workshop on virtual environments, (2003), 19–28.
[16] Reitmayr, G., and Schmalstieg, D., Collaborative Augmented Reality for Outdoor Navigation and Information Browsing. Location Based Services and TeleCartography, (2004), 31–41.
[17] Reitmayr, G. and Drummond, T. W., Going out: Robust tracking for outdoor augmented reality. In Proc. ISMAR 2006, IEEE press (2006), 109–118, Santa Barbara, CA, USA, October 22–25 2006.
[18] Reitmayr, G. and Drummond, T. W., Initialisation for visual tracking in urban environments. In Proc. ISMAR 2007, IEEE press (2007), 161–160.
[19] Reitmayr, G. and Eade, E. and Drummond, T.W., Semi-automatic annotations in unknown environments, In Proc. ISMAR 2007, IEEE press (2007), 67-70.
[20] Ribo, M., Lang, P., Ganster, H., Brandner, M., Stock, C. and Pinz, A., Hybrid tracking for outdoor augmented reality applications. IEEE Comp. Graph. Appl., IEEE press (2002), 54–63.
[21] Schall, G., Wagner, D., Reitmayr, G., Taichmann, E., Wieser, M., Schmalstieg, D. and Hofmann-Wellenhof, B., Global pose estimation using multi-sensor fusion for outdoor augmented reality. In Proc. ISMAR 2009, IEEE press (2009), 153–162.
[22] sprxmobile. Layar reality browser. http://www.layar.com/.
[23] Thomas, B. H., Demczuk, V., Piekarski, W., Hepworth, D. and Gunther, B., A wearable computer system with augmented reality to support terrestrial navigation. In Proc. ISWC'98, (1998), 168–171.
[24] Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D., Pose tracking from natural features on mobile phones. In Proc. 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR), IEEE press (2008), 125–134.
[25] Wagner, D., Mulloni, A., Langlotz T. and Schmalstieg, D., Real-time panoramic mapping and tracking on mobile phones. In Proc. IEEE Virtual Reality 2010, IEEE press (2010).
[26] Wither, J., DiVerd, S. and Hollerer, T., Using aerial photographs for improved mobile AR annotation, In Proc. IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE press (2006), 159-162.
[27] Wither, J., Coffin, C., Ventura, J. and Hollerer, T.,Fast annotation and modeling with a single-point laser range finder, In Proc. 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE press (2008), 65-68.
[28] You, S., Neumann, U. and Azuma, R. Hybrid inertial and vision tracking for augmented reality registration. In Proc. VR 1999, (1999), 260–267.
[29] Zhang, X., and Gao, L., A novel auto-calibration method of the vector magnetometer. In Proc. Electronic Measurement Instruments, ICEMI '09, volume 1, (2009), 145–150.