Exploiting Sensors on Mobile Phones to Improve Wide-Area Localization

Clemens Arth, Alessandro Mulloni, Dieter Schmalstieg
Graz University of Technology
{*arth,mulloni,schmalstieg*}*@icg.tugraz.at*

Abstract

In this paper, we discuss how the sensors available in modern smartphones can improve 6-degree-of-freedom (6DOF) localization in wide-area environments. In our research, we focus on phones as a platform for large-scale Augmented Reality (AR) applications. Thus, our aim is to estimate the position and orientation of the device accurately *and* fast *– it is unrealistic to assume that users are willing to wait tenths of seconds before they can interact with the application. We propose supplementing vision methods with sensor readings from the compass and accelerometer available in most modern smartphones. We evaluate this approach on a large-scale reconstruction of the city center of Graz, Austria. Our results show that our approach improves both accuracy and localization time, in comparison to an existing localization approach based solely on vision. We finally conclude our paper with a real-world validation of the approach on an iPhone 4S.*

1. Introduction

Highly accurate 6 degree-of-freedom (6DOF) localization is the first and most important part of any Augmented Reality (AR) application. The position and orientation of the user's device in the environment must be estimated, before any augmentation can occur. In mobile AR, we face the challenge of performing this estimate on a phone, typically in wide-area environments. Due to the interactive nature of AR applications, localization time has a direct impact on the user experience of an AR application, because it determines how long the user must wait before interaction with the application can start. We therefore need to localize the phone (a) *accurately* in terms of position (sub-meter accuracy) and orientation ($< 5^\circ$ angular error), and (b) *fast*, such that the initialization phase takes a few seconds at most.

In the Computer Vision (CV) community the localization problem has mainly been solved on a coarse scale using computationally demanding algorithms. Exemplary works include [1, 8, 12, 14, 17, 18]. With the exception of [8], the localization task is solved with accuracies up to several meters. Furthermore *localization* is meant to determine a position only (2DOF or 3DOF), rather than a full 6DOF pose, thus these approaches are not directly suitable for AR.

Due to its special relevance, highly accurate and real-time localization on mobile phones is a topic mainly discussed in the AR community. A system for 2DOF outdoor localization was proposed in [15], while a SLAM-like system called PTAM was proposed in [10]. In [4, 6] systems for landmark recognition are proposed, but in both approaches the authors omit to perform any computational tasks on smartphones explicitly. In recent work, we discussed the use of large-scale point-cloud reconstructions for wide-area localization in indoor and outdoor environments [2, 3]. Our work was the first to show 6DOF localization in wide-area environments with sub-meter accuracy on a mobile device.

Apart from GPS, sensors have been rarely used for localization so far. Compass information for orientation tracking was recently utilized [13], while gyroscopes have been studied in [11] for improved feature recognition. Our work adresses this research gap.

We are the first to investigate the joint usage of vision and multiple sensors (GPS, compass, accelerometer) for highly accurate 6DOF localization in wide-area environments on mobile phones. The main contribution of this work is therefore a novel method for partitioning 3D features, such that they can be efficiently matched using sensor information, and the evaluation of previously proposed *gravity-aware* features [11] in outdoor localization. Evaluating our approach in an established localization framework [3], we achieve results superior to previous reports. The performance is considerably improved both in terms of robustness and speed, by exploiting the sensors built into most current smartphones. We validate our approach on an *iPhone 4S* in a real-world scenario, showing the direct applicability of our contribution to the field of mobile AR.

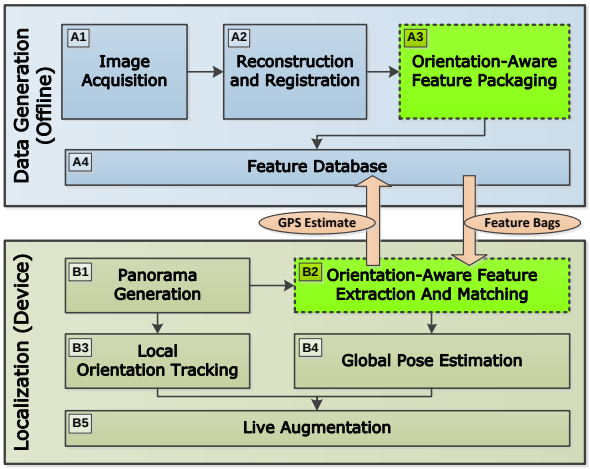


Figure 1: Flowchart of our localization approach.

2. Localization Approach

As shown in Figure 1, our localization approach is divided into an offline data-generation step and an online localization step. In the following, we refer to each blocks of the flowchart by the code in its top-left corner.

Data Generation

First, we capture a set of photographs of the area to reconstruct (**A1**). We then use the photographs to generate a sparse point-cloud reconstruction, using structure-from-motion (SfM) and a modified version of SURF [5]. The reconstruction is registered to global coordinates (**A2**), which is currently done manually. However, alternatively an automatic approach as presented in [9] can be used.

We take into account geographic direction and gravity when extracting the features (Figure 2(a)). Each individual 3D point is not only characterized by a descriptor but also by a *normal vector*. We calculate this normal vector as the mean of the vectors connecting the 3D point and all cameras observing it. Since the reconstruction is upright-oriented, we use the gravity vector as the common feature orientation (similar to [11]), instead of using dominant gradients.

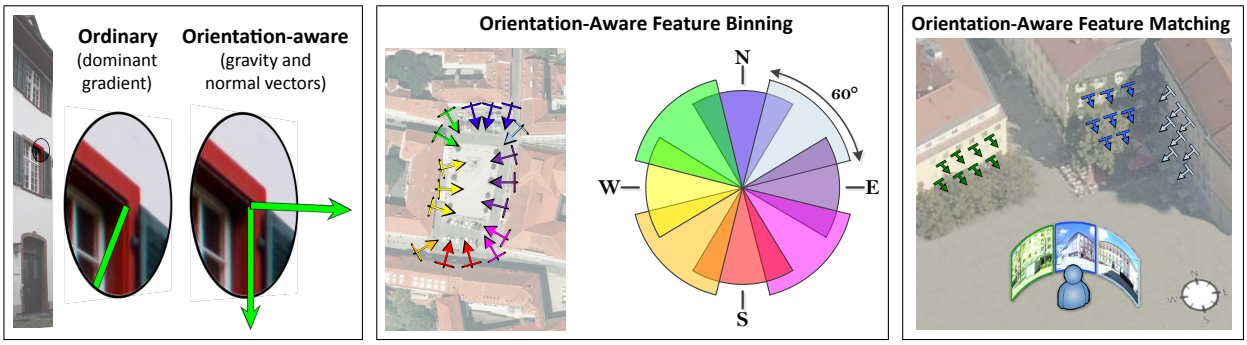
We partition the reconstruction into several overlapping blocks, using a rectangular grid where each block covers 50×50 meters. We store all features of each block as a separate feature bag. The features of a bag are binned based on their orientation with respect to the real geographic direction (**A3**). Since SURF features can be reasonably redetected under $\pm 45°$ of viewpoint change, we use bins covering an angle of about $60°$, such that they slightly overlap (Figure 2(b)). To allow for fast matching, we finally create a tree-like search structure for each bin.

For all blocks of a reconstruction, the corresponding feature bags are stored in a common feature database such that they can be retrieved on demand (**A4**).

Localization

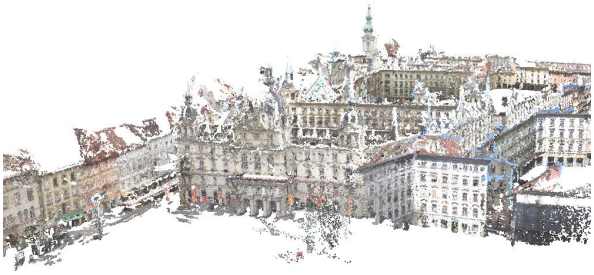
The narrow field of view (FOV) of ordinary mobile phone cameras is a considerable issue for accurate self-localization [2], since a wide baseline for triangulation is missing. We overcome this problem by using an algorithm that uses frame-by-frame tracking to map the live video images onto a 2048×512 pixel panoramic image (**B1**) [16]. As the panorama grows it gives us an increasing FOV on the environment, which guarantees a wider baseline for localization.

While the panorama is created, image features are incrementally extracted (**B2**). We use the accelerometer and magnetometer built into the device to assign each feature a gravity and a normal vector (Figure 2(a)). We expect a GPS estimate to be sufficiently accurate to determine the right 50×50 meter block where the user



(a) (b) (c)

Figure 2: Orientation-aware features. (a) Gradient-based vs. orientation-aware feature extraction. (b) Orientation-aware feature binning using geographic orientation. (c) Orientation-aware feature matching using compass orientation.



(a) Sparse point-cloud reconstruction.

(b) A sample panoramic image of 2048×512 pixel size.

Figure 3: The sparse point-cloud reconstruction and a sample panorama used for the evaluation.

is currently located. We retrieve the corresponding feature bag from the database for matching it against the features from the panorama. Instead of matching an individual feature against all features from the bag, we match it against the bin (Figure 2(c)) corresponding to the feature's normal vector.

Established correspondences are passed to a robust 3-Point-Pose (3PP) algorithm which finally determines a full 6DOF global pose (**B4**) [7]. In doing so we register our panorama to a world reference frame. The panorama tracker also gives us a local orientation estimate (**B3**), which can be combined with the global pose for final usage in our AR applications (**B5**).

3. Experimental Results

We focus our experiments on validating that our new system considerably improves both the robustness and the speed of our previous approach [3]. We take the *time to localize* (T2L) as a measure for speed: this is the time between application start-up and a localization estimate – practically, the time a user must spend capturing a panorama, before localization succeeds. T2L is also proportional to the FOV of the panorama that must be captured for a successful localization.

We reconstructed an area of $\sim 400 \times 100$ meters in Graz, Austria. An exemplary snapshot of the 3D point cloud is shown in Figure 3(a). Using this reconstruction, we conducted both a quantitative test on a PC and a qualitative validation on a mobile phone.

Quantitative Test

We captured 204 panoramic images (Figure 3(b)) using a *Point Grey Ladybug 3* spherical camera. Each image was aligned to gravity by estimating a correction factor from vanishing points and applying a warping operation. Thereafter, the north direction and the ground-truth position was determined manually. We used the panoramas to compare localization performance between our sensor-aided system and the previous method. We consider localization as successful if the translational distance from the ground truth position is below a specified threshold: we only use translational distance because having a correct position estimate and a wrong orientation estimate is highly unlikely.

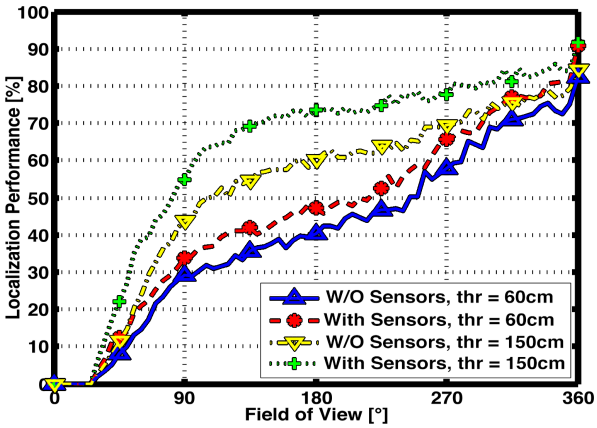


Figure 4: Localization performance without and with sensors, for two different distance thresholds.

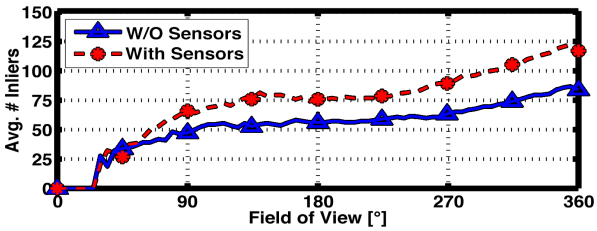


Figure 5: Number of inliers without and with sensors.

We simulate panoramas with varying FOVs by cropping the panoramas from 30° to 360° in steps of 5°, initially pointing towards a building façade. In Figure 4, we show a comparison between our sensor-aided system and the previous method, for two different distance thresholds. Since a small FOV violates the wide-baseline requirement of the 3PP algorithm, a bigger improvement is gained for a looser distance threshold. Although localization performance is already high for our previous method, we can see that sensors manage to further improve it by up to 15%. It is important to stress the proportionality between FOV and T2L: pushing the performance curve towards the upper left corner means that also T2L is decreased significantly.

	Panorama generation	Feature extraction	Feature matching	Pose estim.
W/o sensors [3]	**9.1**	4.9	36.8	0.7
With sensors		**4.9**	**8.2**	**0.2**

Table 1: Average execution time of our method, with and without sensors, on an *iPhone 4S* (all timings in milliseconds).

	Avg. T2L [s]	Min./Max. T2L [s]	Avg./Max. Speedup
W/o sensors [3]	14.18	3.17/22.30	**1.71/3.88**
With sensors	**8.30**	**2.80/14.41**	

Table 2: T2L with and without sensors for 21 video sequences on an *iPhone 4S*.

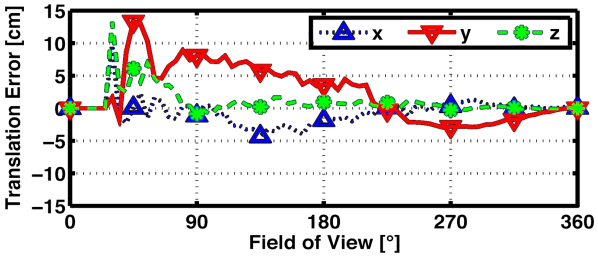


Figure 6: Mean translation error for successful localization estimates using a distance threshold of $30cm$.

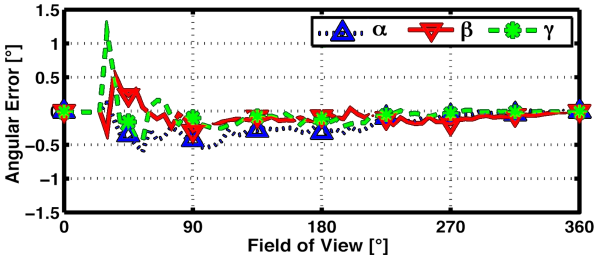


Figure 7: Mean rotation error for successful localization estimates using a distance threshold of $30cm$.

Figure 5 shows a comparison of the two methods in terms of inliers. Due to the sensor-aided feature management, the number of inliers can be increased by up to 50%. This in turn increases robustness considerably, since the average percentage of inliers is only between $5-10\%$ of the total number of feature correspondences.

As shown in Figure 6 for a distance threshold of $30cm$, if localization succeeds the error in translation is below $15cm$ for all three dimensions, and decreases with increasing FOV. Similarly, the angular error is below $1.5°$ and decreases, as depicted in Figure 7.

Qualitative Validation

We implemented our approach on an *iPhone 4S* and evaluated the performance on the device. We recorded 21 different videos (and sensor measurements) in the area of one block with the phone, starting with random view directions and keeping the velocity of rotation around the vertical axis constant. We then processed all videos on the device, both with and without using sensor information.

Table 1 shows the average execution time of our method on the device. Sensors mainly have an impact on matching time, because with sensors the features are matched only against the feature bin corresponding to their normal vector, and not against the whole feature bag. Pose estimation is also sped up by sensors due to the higher percentage of inliers.

The T2L results are shown in Table 2. The sensor-aided method has an average T2L speedup of 1.71 over the previous method, and is at times almost four times faster. In average, users can expect that AR applications using our sensor-aided method will initialize in half of the time compared to applications based on the previous method. The perceivable performance improvement of the sensor-based method is also demonstrated in the accompanying video[1]. Some sample snapshots are depicted in Figure 8.

4. Concluding Remarks

In this paper we presented our work on wide-area localization for smartphones using multiple sensors. To the best of our knowledge, our work is the only one so far discussing the use of sensors on mobile phones in wide-area localization. We show that by the use of sensor data, the robustness and speed of current localization methods can be improved considerably. This result directly impacts the usability of AR applications, because it allows for a much faster startup time.

The performance of vision-based localization systems using sparse SfM reconstructions is dependent on multiple factors. First, the repeatability of feature detectors, especially under extreme lighting conditions, is an influential factor. Similarly, a high discriminability of descriptors is crucial, while a certain level of tolerance to lighting changes is required. We expected gravity-aligned features to expose a level of performance considerably superior to non-aligned features, especially in tackling the problem of repetitive structures prevalent in large-scale urban scenarios. However, contrary to considerations mentioned in [11], gravity-aligned features turned out to suffer from the same issues as non-aligned features, on a higher semantic level however (*e.g.* con-

[1]Complementary video: `http://tinyurl.com/8ykk29u`

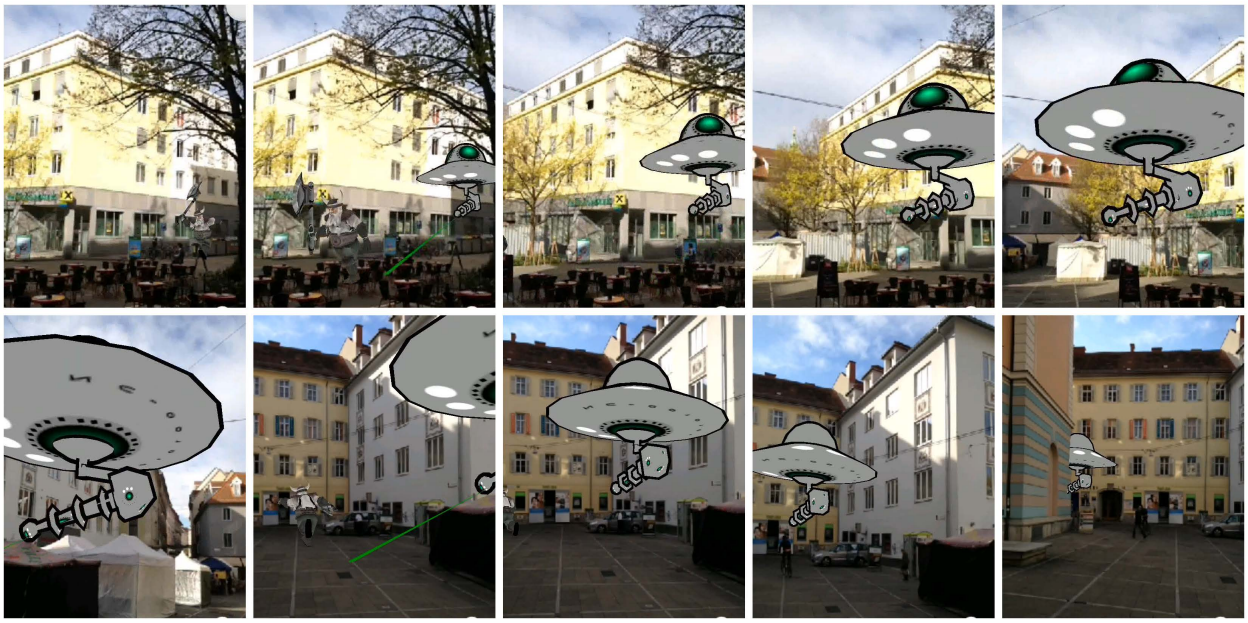


Figure 8: Sample frames from the augmented live video stream, recorded directly on the *iPhone 4S*.

fusing multiple similar windows rather than confusing the corners of a single window). A second issue concerns SfM reconstructions becoming outdated sooner or later through changes in the real environment. Procedures to maintain reconstructions up to date are still missing. In this respect, we argue that more evolved image features and matching procedures, as well as improved SfM reconstruction and maintenance techniques are required to further robustify and improve localization approaches in the future.

Acknowledgements. This work was funded by the Christian Doppler Laboratory for Handheld AR. We thank Albert Walzer for creating the video.

References

[1] M. Agrawal and K. Konolige. Real-Time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS. In *ICPR*, volume 3, pp. 1063–1068, 2006.

[2] C. Arth and *et al*. Wide Area Localization on Mobile Phones. In *ISMAR*, pages 73–82, 2009.

[3] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. Real-Time Self-Localization from Panoramic Images on Mobile Devices. In *ISMAR*, pages 37–46, 2011.

[4] G. Baatz and *et al*. Leveraging 3D City Models for Rotation Invariant Place-of-Interest Recognition. *IJCV, Special Issue on Mobile Vision*, 2011.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *CVIU*, 110(3):346–359, June 2008.

[6] D. Chen and *et al*. City-scale Landmark Identification on Mobile Devices. In *CVPR*, 2011.

[7] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and Analysis of Solutions of the 3 Point Perspective Pose Est. Problem. *IJCV*, 13:331–356, 1994.

[8] A. Irschara, C. Zach, J. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, pages 2599–2606, 2009.

[9] R. Kaminsky, N. Snavely, S. Seitz, and R. Szeliski. Alignment of 3D Point Clouds to Overhead Images. In *IEEE Workshop on Internet Vision (held at CVPR)*, pages 63 –70, 2009.

[10] G. Klein and D. Murray. Parallel Tracking and Mapping on a Camera Phone. In *ISMAR*, pages 83–86, 2009.

[11] D. Kurz and S. BenHimane. Inertial Sensor-Aligned Visual Feature Descriptors. In *CVPR*, pp. 161-166, 2011.

[12] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and Recogn. of Landmark Image Coll. Using Iconic Scene Graphs. In *ECCV*, pp. 427–440, 2008.

[13] G. Schall, A. Mulloni, and G. Reitmayr. North-centred Orientation Tracking on Mobile Phones. In *ISMAR*, pages 267–268, 2010.

[14] G. Schindler, M. Brown, and R. Szeliski. City-Scale Location Recognition. In *CVPR*, 2007.

[15] G. Takacs and *et al*. Outdoors Augmented Reality on Mobile Phone using Loxel-based Visual Feature Organization. In *MIR*, pages 427–434, 2008.

[16] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg. Real-time Panoramic Mapping and Tracking on Mobile Phones. In *VR*, pages 211 –218, march 2010.

[17] A. R. Zamir and M. Shah. Accurate Image Localization Based on Google Maps Street View. In *ECCV*, pages 255–268, 2010.

[18] W. Zhang and J. Kosecka. Image Based Localization in Urban Environments. In *3DPVT*, pages 33–40, 2006.