Instant Motion Tracking and Its Applications to Augmented Reality

Jianing Wei, Genzhi Ye, Tyler Mullen,
Matthias Grundmann, Adel Ahmadyan, Tingbo Hou
Google Research
{jianingwei, yegenzhi, tmullen, grundman, ahmadyan, tingbo}@google.com

Abstract

Augmented Reality (AR) brings immersive experiences to users. With recent advances in computer vision and mobile computing, AR has scaled across platforms, and has increased adoption in major products. One of the key challenges in enabling AR features is proper anchoring of the virtual content to the real world, a process referred to as tracking. In this paper, we present a system for motion tracking, which is capable of robustly tracking planar targets and performing relative-scale 6DoF tracking without calibration. Our system runs in real-time on mobile phones and has been deployed in multiple major products on hundreds of millions of devices.

1. Introduction

Mobile phones carry an enormous amount of computational power in a small package, making them an excellent platform for real-time computer vision and augmented reality applications. Recent releases of ARCore [1] and ARKit [2] scaled Augmented Reality (AR) to hundreds of millions of mobile devices across major mobile computing platforms. Their success is built on advances in computer vision, *e.g*. SLAM [20, 13, 18] and increases in on-device computational power.

A critical component of AR is the ability to anchor virtual content to the real world by tracking the environment. Tracking provides the 3D transform that enables the accurate placement and rendering of virtual content in the real world. Augmentation using virtual content can be simply overlaying a 2D texture, or rendering complex 3D characters into real scenes.

In this paper, we propose a novel instant motion tracking system, based on robust feature tracking, as well as global and local motion estimation. With a shared motion analysis module, our system is capable of performing both planar target tracking and anchor region-based 6DoF tracking (using a mobile device's orientation sensor). Unlike SLAM, our system does not require calibration or initialization to introduce parallax. It is also amenable to tracking moving regions. By removing the need for calibration, it enables AR applications to be deployed at a large scale. By removing the need for initialization, we can place AR content instantaneously (even on moving surfaces), without requiring users to translate their phones first.

Our main contributions are:

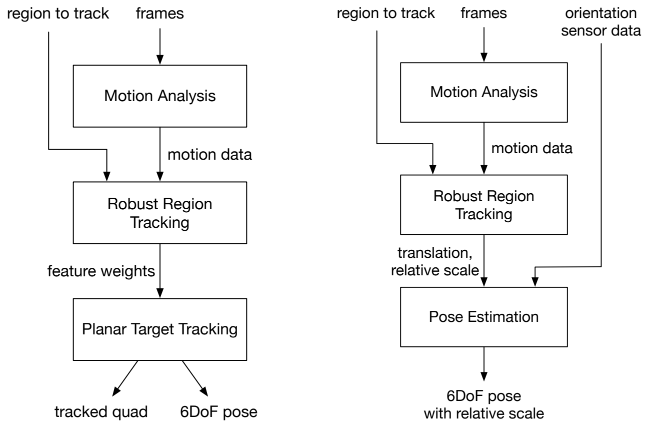
- A system that is robust in the face of degenerate cases like planar scenes, no user motion, and pure rotation.
- Calibration-free placement of AR components without a complex initialization procedure.
- Real-time performance on mobile phones, serving millions of users AR content across a wide range of mobile devices.

2. Related work

Standard SLAM pipelines [20] require users to perform parallax-inducing motions, the so-called *SLAM wiggle* [3], to initialize the world map [14]. These systems can fail in the presence of degenerate cases, such as slow-moving cameras, pure rotations, planar scenes, and tracking distant objects. *Homographies*, on the other hand, can accurately and robustly describe the motion in such cases [16, 10].

Accurate initialization improves the resilience of SLAM algorithms and makes optimization converge faster. Researchers have relied on Structure-from-Motion (SfM) techniques, *e.g*. rotation averaging [3, 7], or closed-form solutions [9] to initialize the camera trajectories and the world map. However, these techniques still require parallax-inducing motion and accurate calibration, rendering them problematic for instant AR placement.

Planar trackers are widely used in SfM applications and panoramic image registration [21]. [16] studied planar tracking for augmented reality applications. Direct region tracking algorithms typically use a homography to warp an image patch from the template to the source and minimize the difference [5]. [6] proposed a region-based planar tracker using a second-order optimization method for minimizing SSD errors. [17] is another region tracker using second-order optimization to minimize the sum of conditional variances. Lucas-Kanade and compositional trackers [5] require re-evaluating the Hessian of the loss function



(a) Planar target tracking

(b) 6DoF tracking

Figure 1: A diagram of our instant motion tracking system.

at every iteration. Inverse compositional trackers [5] speed up tracking by avoiding re-evaluation of the Hessian matrix.

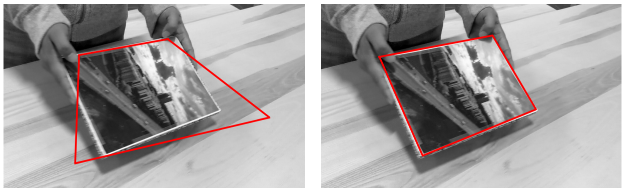
In [15], the authors propose a homography-based planar detection and tracking algorithm to estimate 6DoF camera poses. Recently, [4] used detected surfaces from an image retrieval pipeline to initialize depth from the surface map. [8] adopted gradient orientation for direct surface tracking. Correlation filters are also utilized to estimate rotation as well as scale for 4DoF tracking [11]. [12] built a planar object tracking dataset and surveyed some of the related work in planar tracking. [10] proposed a model selection algorithm to detect which model, homography, or essential matrix describes the motion better.

3. Instant motion tracking

Our instant motion tracking system consists of a motion analysis module, a region tracking module, and either a planar target tracking module for planar surfaces as shown in fig. 1a, or a pose estimation module for calibration-free 6DoF tracking as shown in fig. 1b. In this section, we will briefly describe each of these modules.

3.1. Motion analysis

The motion analysis module extracts and tracks features over time, classifies them into foreground and background, and estimates a temporally coherent camera motion model. We create temporally consistent tracks by assigning each feature path a unique ID. We describe the camera motion by the highest degree of freedom model that can be robustly computed, depending on number and distribution of features, ranging from a 2DoF translation model, to a 4DoF similarity model, and finally to an 8DoF homography model. The feature extraction and tracking can be done using standard methods, including finding good features to track [19]. The feature locations with their IDs and motion vectors (with camera motion subtracted) for an entire frame, along with the full-frame camera motion model, are packed into the *motion data* and sent to the region tracking module.



(a)

(b)

Figure 2: A comparison between (a) homography tracking and (b) perspective tracking of 500 consecutive frames.

3.2. Region tracking

Using solely the motion data produced by the motion analysis module, the region tracking algorithm tracks individual objects or regions while discriminating them from others. To track an input region, we first crop the motion data to a corresponding dilated sub-region. Then, using iteratively reweighted least squares (IRLS) we fit a parametric model to the region's weighted motion vectors to determine the region's movement across consecutive frames.

Our importance weights w_i for each vector v_i are of the form $w_i = \frac{p_i}{e_i}$ with p_i being the prior of a vector v_i's importance and e_i the iteratively refined fitting error. Each region has a tracking state that defines the prior p_i, and includes the mean velocity, the set of inlier and outlier feature IDs, and the region centroid. Note that by relying on feature IDs we implicitly capture the region's appearance since each feature's patch intensity stays roughly constant over time. Additionally, by decomposing a region's motion into that of the camera motion and the individual object motion, we can even track featureless regions.

An advantage of our architecture is that the motion analysis yields a compact motion metadata over the full image, enabling great flexibility and constant computation independent of the number of regions tracked. For example, we can easily track multiple regions simultaneously, and cache metadata across a batch of frames to quickly track regions both backwards and forwards in time; or even sync directly to a specified timestamp for *random access* tracking.

3.3. Planar target tracking

Many object-centric AR use cases require tracking of planar targets to augment them with virtual objects. Examples of planar targets include QR codes, AR markers, and image/texture targets. Unlike region tracking (discussed in section 3.2), planar target tracking is capable of providing absolute orientation and absolute position–with known physical size. In this section, we propose a variant of the region tracking algorithm tailored to tracking any planar target with known shape (commonly, a rectangle) with high accuracy and resilience. For the sake of brevity, we limit

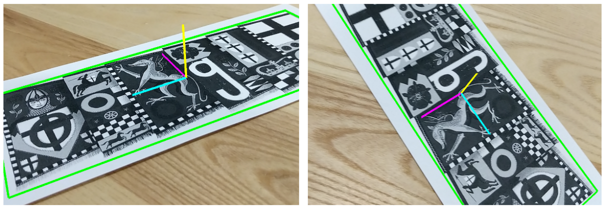


Figure 3: Planar target tracking results from two different perspectives, with 3D local coordinates overlaid.

our discussion to a rectangular-shaped planar target.

The goal of planar target tracking is to estimate the image coordinates of the four corners of the target (a quadrilateral) across frames. In this scenario, a homography transformation is commonly used to describe the inter-frame movement of the quadrilateral [16]. Specifically, a homography matrix is first estimated from feature correspondences between frames, and applied to update the position of the quadrilateral. While a homography has 8 degrees of freedom, in reality, the rigid body transformation that the target undergoes in 3D space is limited to 6 degrees of freedom. Consequently, the under-constrained nature of the homography transform produces quadrilateral shapes which are not physically possible. These estimation errors, accumulated over time, cause skew artifacts (even disregarding camera lens distortion) as shown in fig. 2a.

Instead, we advocate using a perspective transform to estimate the updated corner locations of the quadrilateral. Given the 3D coordinates of features in the previous frame and the corresponding 2D coordinates in the current frame, we solve for the rigid body transformation (3D rotation and translation) from the target local coordinates to the camera coordinates using Levenberg-Marquardt optimization. In fig. 2b, we demonstrate that our approach reduces skew artifacts and maintains the tracking quality and accuracy for as long as possible.

3.4. Pose estimation: calibration-free 6DoF

The method for planar target tracking described above is restricted to scenarios with known object geometry. However, our goal is to enable 6DoF pose estimation for *all* kinds of targets, to enable users to place 3D virtual objects in the viewfinder, making them appear to be part of the real-world scene. Our key insight to enable this is to decouple the camera's translation and rotation estimation, treating them instead as independent optimization problems.

We employ our image-based region tracker to estimate translation and relative scale differences. The result models the 3D translation of a tracked region w.r.t. the camera (using a simple pinhole camera model).

Separately, the device's 3D rotation (roll, pitch, and yaw) is retrieved from the built-in gyroscope. The local orientation is calculated relative to a canonical global orientation, which we compute on initialization. Using the fused gravity vector from the accelerometer sensor, the "up" direction is observable, which yields a sensible initial orientation if we further assume initial object placement on a relatively horizontal surface. This final assumption is purely based on how we imagine users will place virtual objects, and works well in practice.

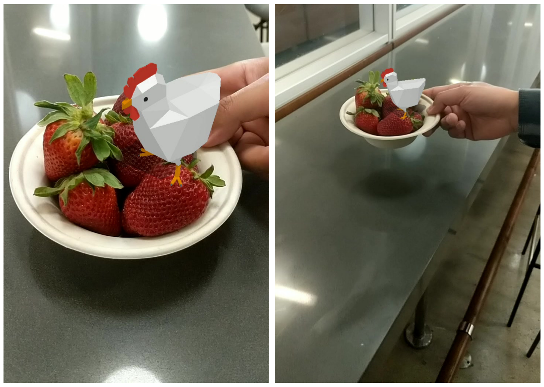


Figure 4: AR sticker effect: tracking a moving, non-planar region across a change in camera perspective.

To make the effect more robust, we also allow for limited region tracking outside the camera's field of view–enabling a virtual object to reappear in approximately the same spot when panning away and back again. Combining visual information for 3D translation and IMU data for 3D rotation lets us track and render virtual content correctly in the viewfinder with 6DoF, with the initial object scale being set by the user. With this parallelization, the system is fast and efficient, can track moving or static regions, and furthermore requires no calibration–it works on any device with a gyroscope.

4. Results and applications to AR

In fig. 3 we demonstrate planar target tracking results for a real-world image from two different perspectives. Note that the tracking is capable of producing an accurate 2D quadrilateral as well as the 3D local coordinate frame of the image. When the physical size of the image is provided, the rotation and translation will be accurate to real-world scale. In fig. 4 we show an AR sticker effect powered by our calibration-free 6DoF tracking. This technology is driving major AR self-expression applications on mobile phones, achieving on average 27.5 FPS across 415+ distinct devices.

5. Conclusion

In this paper, we present a system of instant motion tracking, enabling planar target tracking and relative-scale 6DoF tracking. Our system is calibration-free and robust to degenerate cases. It has been deployed on hundreds of millions of mobile devices, driving major AR applications.

References

[1] ARCore. https://developers.google.com/ar/. [Online; accessed 22-April-2019]. 1

[2] ARKit. https://developer.apple.com/arkit/. [Online; accessed 22-April-2019]. 1

[3] Alvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, and Ian Reid. Visual SLAM: Why Bundle Adjust? *arxiv*, pages 1–7, Feb. 2019. 1

[4] Clemens Arth, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit. Instant outdoor localization and SLAM initialization from 2.5 D maps. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1309–1318, Nov. 2015. 2

[5] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. In *International Journal of Computer Vision*, volume 56, page 221255, 2004. 1, 2

[6] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 943–948 vol.1, Sep. 2004. 1

[7] L Carlone, R Tron, K Daniilidis, and F Dellaert. Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization. ICRA. 1

[8] Lin Chen, Haibin Ling, Yu Shen, Fan Zhou, Ping Wang, Xiang Tian, and Yaowu Chen. Robust visual tracking for planar objects using gradient orientation pyramid. *J. of Electronic Imaging*, 28(1), Jan. 2019. 2

[9] Javier Domínguez-Conti, Jianfeng Yin, Yacine Alami, and Javier Civera. Visual-Inertial SLAM Initialization: A General Linear Formulation and a Gravity-Observing Non-Linear Optimization. *IEEE Computer Graphics and Applications*, 2018. 1

[10] S Gauglitz, C Sweeney, and J Ventura. Live tracking and mapping from both general and rotation-only camera motion. *ISMAR*, pages 13–22, 2012. 1, 2

[11] Yang Li, Jianke Zhu, Steven C.H. Hoi, Wenjie Song, Zhefeng Wang, and Hantang Liu. Robust estimation of similarity transformation for visual object tracking. In *The Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, January 2019. 2

[12] Pengpeng Liang, Yifan Wu, Hu Lu, Liming Wang, Chunyuan Liao, and Haibin Ling. Planar object tracking in the wild: A benchmark. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 651–658, 2018. 2

[13] Simon Lynen, Torsten Sattler, Michael Bosse, Joel Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, 2015. 1

[14] Mahesh Ramachandran Alessandro Mulloni. User Friendly SLAM Initialization. *International Symposium on Mixed and Augmented Reality*, pages 1–10, Oct. 2013. 1

[15] Christian Pirchheim and Gerhard Reitmayr. Homography-based planar mapping and tracking for mobile phones. *IEEE International Symposium on Mixed and Augmented Reality*, pages 27–36, 2011. 2

[16] Simon J.D. Prince, Ke Xu, and Adrian David Cheok. Augmented reality camera tracking with homographies. *IEEE Computer Graphics and Applications*, 22(6):39–45, Nov. 2002. 1, 3

[17] R. Richa, R. Sznitman, R. Taylor, and G. Hager. Visual tracking using the sum of conditional variance. In *International Conference on Intelligent Robots and Systems*, pages 2953–2958, Sep. 2011. 1

[18] Thomas Schneider, Marcin Dymczyk, Marius Fehr, Kevin Egger, Simon Lynen, Igor Gilitschenski, and Roland Siegwart. maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, pages 1418–1425, 2018. 1

[19] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 1994. 2

[20] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, UK, 2000. Springer-Verlag. 1

[21] Daniel Wagner, Alessandro Mulloni, Tobias Langlotz, and Dieter Schmalstieg. Real-time panoramic mapping and tracking on mobile phones. *2010 IEEE Virtual Reality Conference (VR)*, pages 211–218, Mar. 2010. 1