

Instant 3D Object Tracking with Applications in Augmented Reality

Adel Ahmadyan * Tingbo Hou * Jianing Wei * Liangkai Zhang Artsiom Ablavatski
 Matthias Grundmann,
 Google Research
 1600 Amphitheatre Pkwy, Mountain View, CA 94043
 {ahmadyan, tingbo, jianingwei, liangkai, artsiom, grundman}@google.com

Abstract

Tracking object poses in 3D is a crucial building block for Augmented Reality applications. We propose an instant motion tracking system that tracks an object’s pose in space (represented by its 3D bounding box) in real-time on mobile devices. Our system does not require any prior sensory calibration or initialization to function. We employ a deep neural network to detect objects and estimate their initial 3D pose. Then the estimated pose is tracked using a robust planar tracker. Our tracker is capable of performing relative-scale 9-DoF tracking in real-time on mobile devices. By combining use of CPU and GPU efficiently, we achieve 26-FPS+ performance on mobile devices.

1. Introduction

Tracking in monocular videos is a challenging and well studied problem in computer vision. While 2D tracking is mature with robust solutions [16, 13, 3, 4, 12, 10], 3D tracking from monocular RGB images remains an open problem. Current approaches to 3D tracking [7, 17, 18] require complex initialization procedures to estimate depth, and are not very robust and have high computational cost.

The objective of 3D tracking is to track the 3D bounding box of a rigid object throughout frames when both camera and object motions are present. The object pose in 3D is uniquely determined by its 3D bounding box and has 9 DoF including orientation, translation, and physical size.

In this paper, we propose a system to detect and track an object’s 3D pose in real-time. Initially, we detect the object and estimate its pose using a deep neural network [6]. The detection network does not require prior knowledge of the object’s shape, size or CAD model to be known and can detect category-level unseen objects. This model can run in real-time on mobile devices and can be used in a *tracking-by-detection* paradigm. When the model is applied to every frame, the detection output may suffer from jitter due

to the prediction noise from the model. This jitter is undesirable for AR applications. We adopt a detection-plus-tracking framework to mitigate this issue. This framework mitigates the need to run the network on every frame, allowing the use of heavier and therefore more accurate models, while keeping the pipeline real-time on mobile devices. It also retains object identity across frames and ensures that the prediction is temporally consistent, effectively reducing the jitter.

Our detection-plus-tracking system works as follows. We first locate the object’s pose using the detection network, estimating its 3D bounding box. We then project the bounding box’s 3D vertices to the image plane and track the 2D points, which rest on a plane, using a planar tracker [16]. Finally, we lift the tracked 2D points to 3D using the EPnP [8] algorithm to estimate the 3D bounding box in subsequent frames. This work extends our previous works on 3D object detection [6], instant motion tracking [16], and 3D object tracking [2].

Our tracker has three properties: it is robust, instant, and real-time. The tracker is very efficient in utilizing both CPU and GPU on-device for tracking and detection, respectively, thus achieving real-time (26-FPS+) performance on mobile devices. The detection is performed on a single RGB image and the planar tracker is instant. Thus our whole pipeline does not require any parallax-inducing motion to initialize and locate the object and its overall latency is low. Finally, with the assumption that 3D bounding box sits on a planar ground, the planar tracker is very robust given typical AR applications.

Our main contributions are:

- We propose an end-to-end system to track the object’s 3D pose (orientation, translation, and size up to a scale). This system uses a CNN to initialize the pose, then track it using a planar tracker.
- The proposed system is calibration-free and does not require any complex initialization sequence or any hardware beyond camera or IMU sensors. It does not require prior knowledge of object’s shape or model and can detect and track unseen objects.

*Equal contribution

- The end-to-end system (including detection) runs in real-time on mobile devices.

To encourage researchers and developers to experiment and prototype based on our pipeline, we open-sourced our on-device ML pipeline in [1], including an end-to-end demo mobile application and our trained detector for shoes and chairs. We hope that sharing our solution with the wide research and development community will stimulate new use cases, applications, and research efforts.

2. Related work

Efficient and robust tracking is an essential component for any AR application. Object tracking has been studied and practiced extensively in computer vision. Planar and region-based trackers [13, 3, 4, 12, 10] rely on 3D geometry to estimate the camera motion and track objects. Recently, neural nets have been utilized to learn and estimate motion. Generally tracking system consists of two components:
a) detector: which detects the objects in each frame and estimates their 2D or 3D bounding boxes, and
b) a matching algorithm, which tracks object correspondences between frames.

[7] detects an object’s 3D bounding box and then estimates the 3D motion between frames to track the box. Employing a Kalman filter after 3D detection was investigated in [17] and shown to achieve good performance. In [9] and [11], tracking the 3D bounding box with stereo cameras for autonomy applications were studied. [15] used 3D cues for tracking vehicles’ 2D bounding boxes. Recently, in [18] the authors propose to use a deep neural network for both 3D detection and tracking.

3. Instant 3D tracking

Figure 1 shows an overview of our 3D tracking system. Initially, the frames are passed through a single-stage CNN, as shown in Figure 2, to predict the object’s pose and physical size from a single RGB image.

The model backbone has an encoder-decoder architecture, built upon MobileNetV2[14]. We employ a multi-task learning approach, jointly predicting an object’s shape with detection and regression. The shape task predicts the object’s shape signals depending on what ground truth annotation is available, *e.g.* segmentation. This is optional if there is no shape annotation present in the training data. For the detection task, we use the annotated bounding boxes and fit a Gaussian to the box, with the center at the box’s centroid and standard deviations proportional to the box size. The goal for detection is then to predict this distribution with its peak representing the objects center location [5]. The regression task estimates the 2D projections of the eight bounding box vertices. To obtain the final 3D coordinates for the bounding box, we leverage a well established pose estimation algorithm (EPnP) [8]. It can recover the 3D

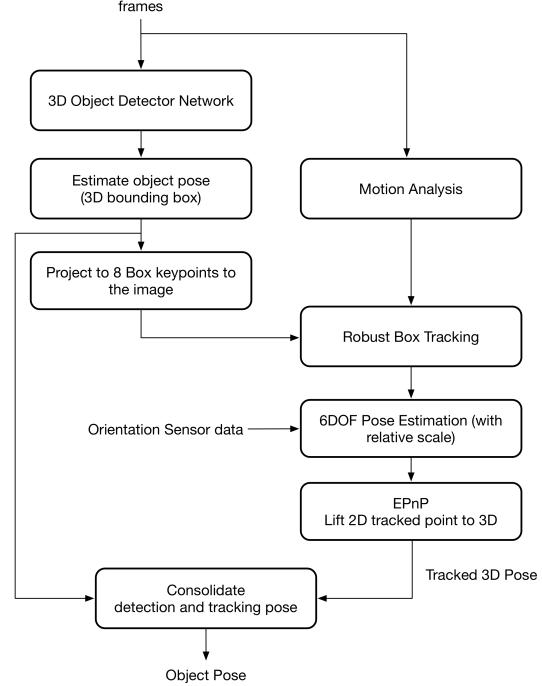


Figure 1: Overview of our 3D tracking system.

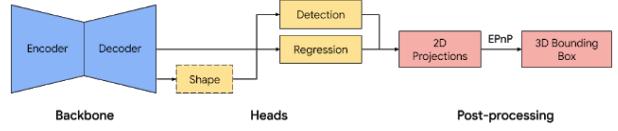


Figure 2: Our network detects the object and estimates the 3D bounding box.

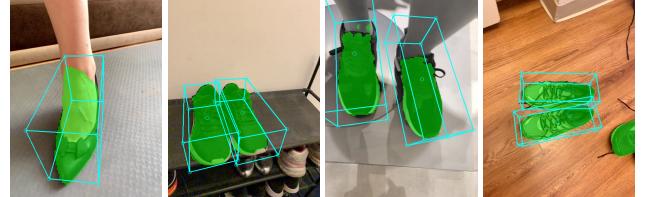


Figure 3: Estimated 3D bounding box and the segmentation mask produced by our object detector network.

bounding box of an object, without a priori knowledge of the object dimensions. Given the 3D bounding box, we can easily compute pose and size of the object. Figure 2 shows our network architecture and post-processing. The model is light enough to run real-time on mobile devices (at 26 FPS on an Adreno 650 mobile GPU). The details of our model is described in [6].

After initializing the object’s pose with the object detec-



Figure 4: Tracking a 3D bounding box with both camera motion and object movement present.

tor network, we compute the nine key-points of the bounding box and project them to the image. The nine key-points consists of the bounding box’s eight vertices plus its center. We track these points using a planar tracker [16]. Our tracking system consists of a motion analysis module, a robust box tracker, and a pose estimation module for calibration-free 6DoF tracking [16]. Tracking the nine 2D points using a 6-DoF relative scale tracker [16] is sufficient for tracking the object’s 9DoF pose. At every frame, we lift the 2D points back to 3D using the EPnP algorithm [8] to estimate the 3D bounding box (with 9-DoF) up-to scale. Consequently, our model inference only needs to run every few frames, resulting in high efficiency in our mobile pipeline. When a new prediction is made, we consolidate the detection result with the tracking result based on the area of overlap.

4. Results and applications to AR

In Figure 4, we demonstrate tracking multiple 3D bounding boxes results from a video. The complete system is implemented in the Mediapipe framework. The model and the code is available at [1]. Our detection network predicts the 3D bounding box with average precision of 0.59 at 0.5 3D IoU. The model weights only 5.54MB. The model’s output also includes shape information such as segmentation mask. The object detector runs at 26.5fps on the mobile



Figure 5: Fitting and rendering a mesh model to the detected objects for AR Applications.

GPU while the 3D tracking runs at 30fps+ on a mobile CPU (Samsung S20 device with Qualcomm’s Snapdragon 865 SoC).

Figure 5 shows an example of how to use the tracked object’s pose for AR applications such as virtual shoe try-on. In each frame, we render a CAD model at the object pose using OpenGL rendering pipeline. The same polygon mesh model is also rendered in Figure 4 inside the bounding box as an occluder to give a 3D effect to the visualization.

We make two key assumptions for the 3D tracking system to work properly: first, there is no gauge ambiguity in the 3D bounding box and the detected eight keypoints are unique. For symmetric objects, e.g. volleyball, this assumption does not hold. As a result, the detection network would predict different orientations each time and that would cause failure when consolidating the detection and tracking results together. The second assumption by the tracking module is that the object’s plane does not significantly change while we are tracking it. This assumption is true for most objects without the roll, however, if the tracked object rolls and changes its plane during tracking, e.g. volleyball, we may lose their track.

5. Conclusion

In conclusion, we present a system for 3D object tracking that enables real-time instant 3D bounding box tracking on mobile devices. Our proposed system uses a neural network to initialize the 3D pose then utilizes a planar surface tracker to track the object’s pose in the video frame. The end-to-end system runs in real-time on mobile devices.

References

- [1] Mediapipe objectron. https://github.com/google/mediapipe/blob/master/mediapipe/docs/objectron_mobile_gpu.md. Accessed: 2020-04-01. [1](#), [2](#), [3](#)
- [2] Adel Ahmadyan and Tingbo Hou. Real-Time 3D Object Detection on Mobile Devices with MediaPipe. ai.googleblog.com/2020/03/real-time-3d-object-detection-on-mobile.html, 2020. [1](#)
- [3] 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](#)
- [4] 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](#), [2](#)
- [5] Li Ding and Lex Fridman. Object as distribution, 2019. [2](#)
- [6] Tingbo Hou, Adel Ahmadyan, Liankai Zhang, Jianing Wei, and Matthias Grundmann. MobilePose: Real-Time Pose Estimation for Unseen Objects with Weak Shape Supervision. *arXiv preprint arXiv:2003.03522*, 2020. [1](#), [2](#)
- [7] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krhenbhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking, 2018. [1](#), [2](#)
- [8] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(N) solution to the PnP problem. *International Journal of Computer Vision (IJCV)*, 81(2):155–166, 2009. [1](#), [2](#), [3](#)
- [9] Peiliang Li, Tong Qin, and Shaojie Shen. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving, 2018. [2](#)
- [10] 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. [1](#), [2](#)
- [11] Aljosa Osep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. Combined image- and world-space tracking in traffic scenes, 2018. [2](#)
- [12] 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. [1](#), [2](#)
- [13] 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](#), [2](#)
- [14] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2018. [2](#)
- [15] Sarthak Sharma, Junaid Ahmed Ansari, J. Krishna Murthy, and K. Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking, 2018. [2](#)
- [16] Jianing Wei, Genzhi Ye, Tyler Mullen, Matthias Grundmann, Adel Ahmadyan, and Tingbo Hou. Instant Motion Tracking and Its Applications to Augmented Reality. In *CVPR Workshop on Computer Vision for Augmented and Virtual Reality 2019*, Long Beach, CA, 2019. [1](#), [3](#)
- [17] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking, 2019. [1](#), [2](#)
- [18] Xingyi Zhou, Vladlen Koltun, and Philipp Krhenbhl. Tracking objects as points, 2020. [1](#), [2](#)