Simultaneous Tracking, Tagging and Mapping for Augmented Reality

Yixiao Kang, Yiyang Xu, Chao Ping Chen*, Gang Li, Ziyao Cheng
Smart Display Lab, Shanghai Jiao Tong University, Shanghai, China
Email: ccp@sjtu.edu.cn

Abstract

We present a method of simultaneous tracking, tagging and mapping (STTM) for the augmented reality (AR) by feeding off the deep-SORT-based object tagging and lightweight unsupervised deep loop closure.

Keywords

augmented reality; simultaneous tracking, tagging and mapping; object detection; deep learning; loop closure.

1. Proposed Method

Pipeline: Fig. 1 outlines the pipeline of the proposed STTM. Our method starts with measurement preprocessing. All necessary values for bootstrapping the subsequent nonlinear optimization-based visual-inertial odometry (VIO) are obtained by initialization [1]. The VIO module closely combines the position and pose data of the inertial measurement unit (IMU) with the re-tracked feature from the closed loop detection to complete the re-location. Finally, the pose graph module performs the global pose graph optimization to eliminate cumulative error and to enable the map reuse.

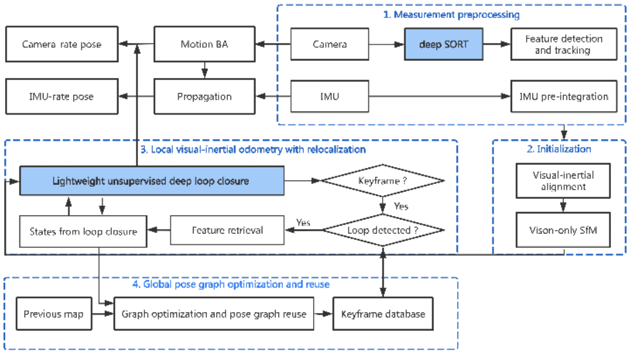

1. Measurement preprocessing
Camera rate pose
Motion BA
Camera
deep SORT
Feature detection and tracking
IMU-rate pose
Propagation
IMU
IMU pre-integration
3. Local visual-inertial odometry with relocalization
2. Initialization
Lightweight unsupervised deep loop closure
Keyframe ?
Yes
Visual-inertial alignment
States from loop closure
Feature retrieval
Yes
Loop detected ?
Vison-only SfM
4. Global pose graph optimization and reuse
Previous map
Graph optimization and pose graph reuse
Keyframe database

Figure 1. Pipeline of the proposed STTM, which features deep-SORT-based object tagging and lightweight unsupervised deep loop closure.

Platform: Intel Core i9-9880H; 16GB 2666MHz DDR4 memory; AMD Radeon Pro 5500M; macOS Big Sur 11.2.1; iPad Pro 12.9-inch (4th generation); ARKit 4; MOT16 benchmark.

2. Results and Discussion

Tracking: In the initialization stage of our experiment, a virtual box generated by the extracted feature point information is inserted into the coordinate, as shown in Fig. 2. Then, an estimated trajectory with a closed loop is recorded for testing the STTM, as shown in Fig. 3, by which the x/y/z coordinates of the camera, total distance of travel, and number of features can be tracked in real time.

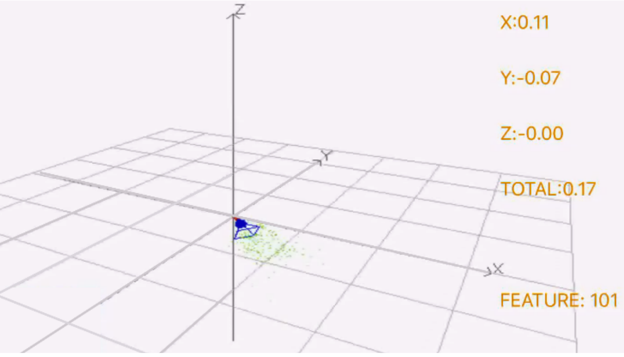


Figure 2. A virtual box is inserted into the coordinate during the initialization stage.

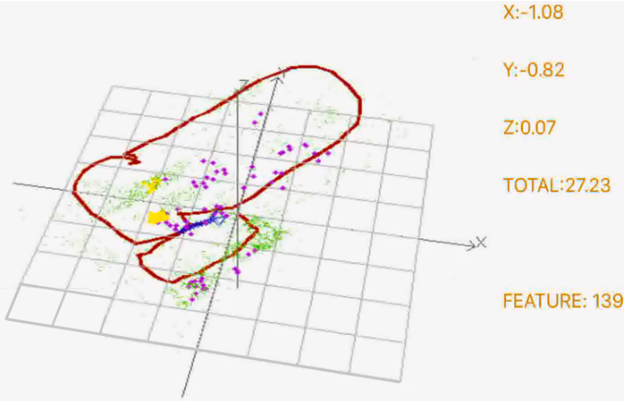
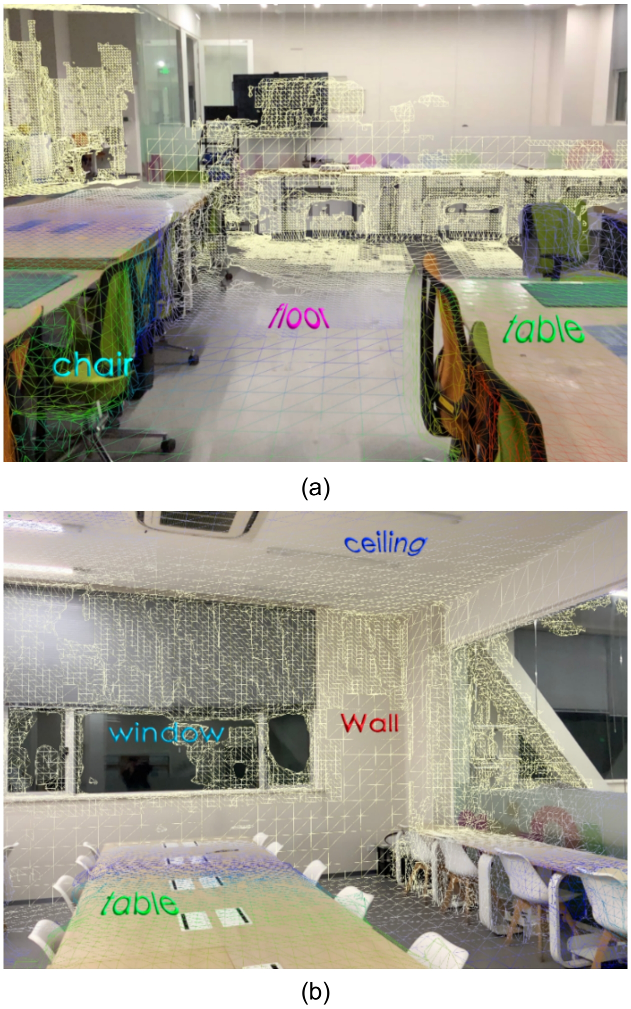


Figure 3. Indoor experimental result for proposed STTM. Total trajectory length is 27.23 m. A closed loop is recorded for testing the STTM.

Tagging: Deep SORT [2], which is based on the Kalman filtering and frame-by-frame data association with Hungarian method, performs smoothly at high frame rates. Meanwhile, with a convolutional neural network (CNN), it can improve the robustness against misses and occlusions and it is also able to recognize the pre-trained objects, including the table, chair, window, wall, ceiling, floor *etc.*, as shown in Fig. 4. Compared with other object tracking algorithms, it exhibits superior performance in terms of the accuracy and reliability. Specifically, it has relatively low identity switches (781) and mostly lost rate (8.2%) on the MOT16 benchmark.

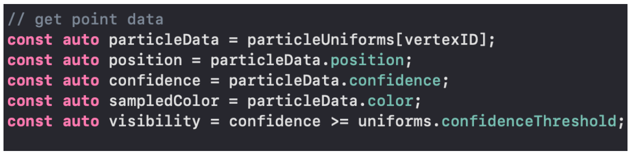


(a)

(b)

Figure 4. Object tagging by the deep SORT.

Point Cloud: To generate the 3D point cloud with the depth information, LiDAR scanner, which is based on the time-of-flight (ToF) technique, is employed to measure the distance between the objects and camera. Those points are organized using a certain data structure. The codes, as shown in Fig. 5, are written in Swift programming language, with which the vertexID, spatial information *etc.* can be directly obtained using the class in ARKit. Plus, the confidence and threshold can be tweaked to reduce the noise and to remove the incomplete data.



```
// get point data
const auto particleData = particleUniforms[vertexID];
const auto position = particleData.position;
const auto confidence = particleData.confidence;
const auto sampledColor = particleData.color;
const auto visibility = confidence >= uniforms.confidenceThreshold;
```

Figure 5. Codes to acquire the point data on ARKit.

Mapping: With the point cloud, a polygonal algorithm is adopted to generate the meshes. We choose Delaunay triangulation algorithm for its simplicity and stability. Delaunay triangulation has the following properties: (1) it runs in *O (N log N)* time; (2) it maxes out the minimum angle, which makes the triangulation unique for most cases [3]. Fig. 6 plots the result of Delaunay triangulation when applied to 20 random points.

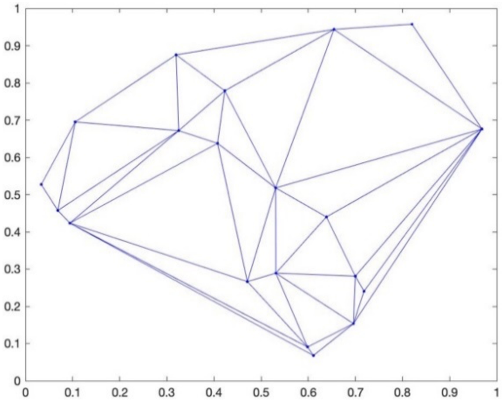
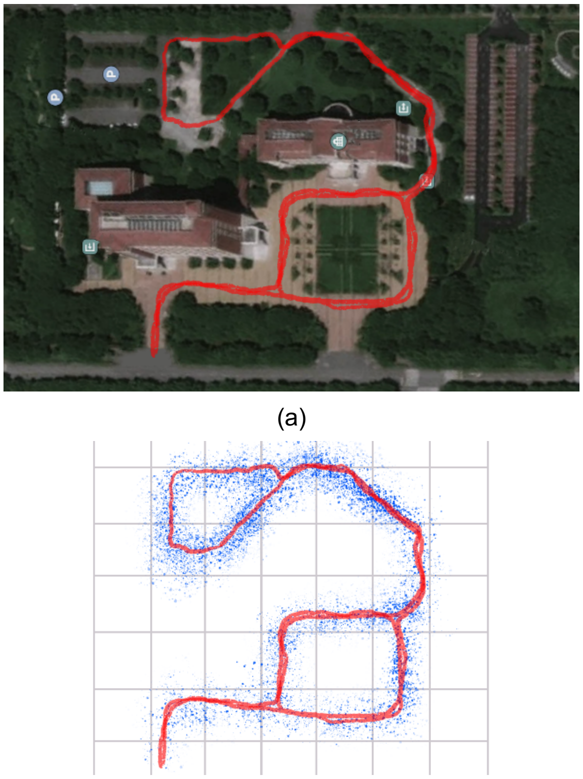


Figure 6. Delaunay triangulation when applied to 20 random points.

Loop Closure: As for the loop closure, instead of resorting to the conventional bag-of-words model, lightweight unsupervised deep loop closure [4] is employed to compensate the camera pose variations by translating high-dimensional raw images into low-dimensional descriptor spaces. We choose an unsupervised, convolutional autoencoder network, which is designed for the loop closure. It is capable of the robust and efficient location identification. Our model could extract features directly from the original image in an unsupervised manner, without the need of requiring the training data with tags or being trained in a specific environment, which makes it more lightweight and versatile.



(a)

(b)

Figure 7. Outdoor experimental result of the proposed STTM. Total trajectory length is 1.2 km. (a) The overlapped trajectory is composed of two round-trip paths, which are superimposed onto a satellite image for the comparison. (b) Trajectory (red) and feature points (blue).

3. Conclusions

As compared to the existing methods of simultaneous localization and mapping (SLAM) [5–7], our STTM offers two major advantages. First, object tagging is added to the framework of SLAM. Second, lightweight CNN-based loop closure is more robust and suitable for wearable AR devices [8].

4. Acknowledgements

This work is sponsored by National Natural Science Foundation of China (61901264, 61831015) and Science and Technology Commission of Shanghai Municipality (19ZR1427200).

5. References

[1] T. Qin, P. Li, and S. Shen, “VINS-mono: a robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics* **34**(4), 1004–1020 (2018).

[2] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *24th IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649 (2017).

[3] M. H. Gross, O. G. Staadt, and R. Gatti, “Efficient triangular surface approximations using wavelets and quadtree data structures,” *IEEE Transaction on Visualization and Computer Graphics* **2**(2), 130–143 (1996).

[4] A. R. Memon, H. Wang, and A. Hussain, “Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems,” *Robotics and Autonomous Systems* **126**, 103470 (2020).

[5] B. Yu, Y. Li, C. P. Chen, N. Maitlo, J. Chen, W. Zhang, and L. Mi, “Semantic simultaneous localization and mapping for augmented reality,” in *SID Display Week*, pp. 391–394 (2018).

[6] Y. Li, B. Yu, C. P. Chen, N. Maitlo, W. Zhang, and L. Mi, “Monocular SLAM using probabilistic combination of point and line features,” in *OSA Imaging and Applied Optics*, p. 3W2G.7 (2018).

[7] Y. Li, C. P. Chen, N. Maitlo, L. Mi, W. Zhang, and J. Chen, “Deep neural network-based loop detection for visual simultaneous localization and mapping featuring both points and lines,” *Wiley Advanced Intelligent Systems* **2**(1), 1900107 (2020).

[8] C. P. Chen, L. Mi, W. Zhang, J. Ye, and G. Li, “Waveguide-based near-eye display with dual-channel exit pupil expander,” *Displays* **67**, 101998 (2021).