Evaluation for Hybrid Location Estimation System of Image Retrieval and SLAM

YAMASAKI Kohei *[a], SHISHIDO Hidehiko[b], KITAHARA Itaru[b], KAMEDA Yoshinari[b],
[a]Graduate School of System and Information Engineering, University of Tsukuba, Ibaraki, Japan
[b]Center for Computational Science, University of Tsukuba, Ibaraki, Japan

ABSTRACT

The availability of pedestrian location estimation is one of the critical issues to realize a reliable navigation system for pedestrians in daily scenes. We propose a new pedestrian location estimation system that utilizes both the image-retrieval approach we have developed and a SLAM (Simultaneous Localization and Mapping) approach. Both approaches need only one single camera unit as a sensor, and the location is estimated by computer vision technology on both approaches. The problem here is that high processing cost is required to operate two approaches simultaneously. It could be impractical to run these two on a single wearable computing unit. We solve the problem by executing the two approaches on two separate computers that are connected with a computer network. We have implemented a preliminary system that unites the two approaches in hybrid fashion over two computers. We measured its performance in typical daily scenes on our campus. The result is promising for further implementation.

Keywords: SLAM, image retrieval, location estimation, pedestrian, visually impaired, CPU load

1. INTRODUCTION

We have been working on a pedestrian navigation system for visually impaired people. An ideal system is that it does not rely on external facilities so that it can be used anywhere. A system that uses only one camera unit as a sensor could be an ideal one. The navigation of visually impaired people needs a highly robust self-localization approach. The ultimate goal is to realize a highly available navigation system using a wearable device that has one camera. We have proposed an image retrieval approach [4] to achieve this goal. Thanks to the recent progress of the technology, SLAM[3] also becomes a right approach for pedestrian location estimation. Since these two approaches have different properties, it should be a good idea to achieve high availability by executing them simultaneously.

We propose a system that uses these two approaches simultaneously. The problem is that high processing costs are required to execute them simultaneously. We solve this problem by using two computing devices. One is called a remote computing device, and the other is called a wearable computing device. A computer network connects the two computing devices. The amount of data transfer and the procedure order between the two devices should be carefully estimated, and the system should be designed to keep the performance close to their isolated executions. By operating the two approaches in a hybrid fashion, a highly available pedestrian position estimation system is realized.

2. RELATED WORKS

To improve the accuracy of position and orientation estimation and map creation, distributing SLAM processing on multiple computer resources have been proposed[2]. However, in any SLAM methods, once failed the position estimation, the system needs to rebuild the map. It needs some time to bootstrap and recovers its functionality. It could be harder for Visual SLAM[1]. On the contrary, if only the system receives a taken frame on the route, the image retrieval approach [4] always works. The problem is that both need high processing costs, and they assume to occupy almost full computation power of a computer.

3. PROPOSED ARCHITECTURE

3.1 Deployed Two Localization Approaches

Figure 1 shows an overview of a system that executes two approaches of image retrieval and SLAM on different computing devices. We perform SLAM on the wearable computing device and image retrieval on the remote computing device.

Image Retrieval

Figure 2 shows location estimation by the image retrieval approach[4]. We use the premise that visually impaired people have a planned walking path that is determined ahead of their actual walking out. We pre-record the route before estimating the location. The camera is headed to capture the frontal view on the pre-walking of the determined path. The captured video is stored in the system, and it will be used as a database. On actual walking out, a query image is captured on the path, and the image retrieval approach answers the location on the way by comparing the similarity of the query image and the video frames in the database. It is done based on the pairing of image feature keys.

SLAM

In this paper, we use RealSense SLAM[6], which can be used with Intel® RealSense ™ Camera ZR300 [5]. ZR300 is equipped with a visible light camera, fisheye camera, depth camera, accelerometer, and gyroscope. RealSense SLAM can estimate the camera unit location precisely in an ordinary environment, yet it may have a chance of losing the location in some cases.

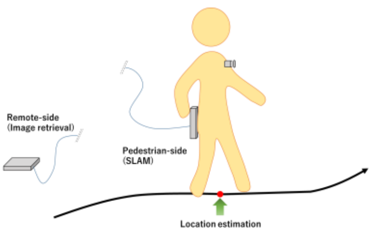
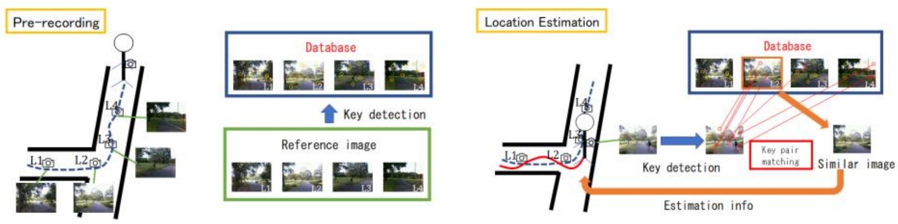


Figure 1. System overview



(a) Pre-recording (b) Location Estimation

Figure 2. Image-Retrieval based approach

3.2 Load Distribution System

The image retrieval and the SLAM are executed on different computing devices to distribute the computation load. Figure 3 shows the harmonic architecture of the data process and flow over the two computing devices. The image acquired by the visible light camera is transmitted from the wearable computing device to the remote computing device as a query image. The remote computing device receives the query image from the wearable computing device and makes a queue of the query images. Typically, the image retrieval process takes the first query in the queue and send back the estimated location back to the control process at the wearable computing device. The image retrieval may take some time to estimate the location. As a result, the location is considered to be old to adopt it as a final answer to the user. The control process works to avoid this situation. The control process checks the timestamp of the most recently captured video frame. If the result from the image retrieval process is older than the time-out threshold, it discards the result and sends a signal to clear the queue to the remote computing device.

4. EXPERIMENT

As for the first experiment, we evaluated the performance of executing the two approaches on one computing device. In the experiment, the CPU usage ratio was shown to measure the load.

Then, as for the second and the third experiment, we ran our prototype system on an actual path and investigated the performance.

To confirm the availability of the system on both wired and wireless networks, we applied the system on a long route in a wired network and a short route in a wireless network.

4.1 Environment

In the wired network experiment, we used two notebook PCs as computing devices. We connected these computing devices with a 100BASE-TX cable. We use Surface Pro (CPU: Corei7-6550U, RAM: 16GB, OS: Windows 10 64bit) as a remote computing device. We used iiyama STYLE (CPU: Corei7-8550U, RAM: 16GB, OS: Ubuntu 16.04) as a wearable computing device. We experimented at the campus of the University of Tsukuba. The path is about 500 [m], SLAM frame rate is 30 [fps], and image size is 640x480 [pixel]. The time-out threshold was set to 500[ms] by taking into account the walking speed in the experiment.

In the wireless network experiment, we used NEC Aterm WG2600HP3 for a wireless device. We installed three wireless devices along the route. One was the base unit, and the other two were extensions. Figure 4 shows an overview of the equipment layout. The remote computer device used was iiyama W650SR (CPU: Corei7-4700MQ, RAM: 16GB, OS: Ubuntu 16.04). The wearable computer device used was DAIV-NG4500E2-SH2 (CPU: Corei7-7000HQ, RAM: 16GB, OS: Ubuntu 16.04). We performed the experiment in the hallway of the University of Tsukuba campus. The route is about 40 meters.

4.2 Results

As for the first experiment, Figure 5 shows the CPU usage of each process when the image retrieval and the SLAM are executed on the wearable computing device. The SLAM process uses most of the CPU resources from the start to the end of execution. The image retrieval process cannot use the CPU at all. This means the system cannot obtain location estimation result from the image retrieval. It results in losing availability when a walking person come to a situation where SLAM does not work well.

As for the second experiment in the wired network, we performed an experiment on the proposed system and investigated the result at each video frame. Figure 6 shows some video frames of the pre-recorded video of the experiment path. The total number of captured video frames is 14,744 frames. Figure 6 shows the graph from the 4,360th frame to the 4,430th frame. In Figure 7. (c), the circled area is the frame when the SLAM could not output the location, yet the image retrieval could. As shown in Figure 6(c), the area around the 4400th frame had fewer image features at a short distance, and they were not spread three-dimensionally. It was not easy for the SLAM.

The third experiment is for the wireless network. In the same way as the second experiment, we performed an experiment with the proposed system and investigated the location estimation success frames. Besides, we investigated the amount of data received by remote computing devices and the amount of data transmitted by wearable computing devices. Figure 8 shows a graph of the amount received by the remote computing device and a graph of the amount transmitted by the wearable computing device. The capacity of one sent / received image data is 320 x 240 x 1 [bytes] = 76800 [bytes] = 75 [kB]. Since this is transmitted and received 30 times per second, 75 x 30 [kB/s] = 2250 [kB/s] ≒ 2.2 [MiB/s]. This is almost identical to the result in Figure 8. We suppose that the subtle error is the capacity other than image data such as header data. Figure 9 shows some video frames of the pre-recorded video of the experiment path. Figure 10 shows the graph from the 1,150th frame to the 1,180th frame. It can be seen that there is a frame for which the image retrieval succeeds while the SLAM fails.

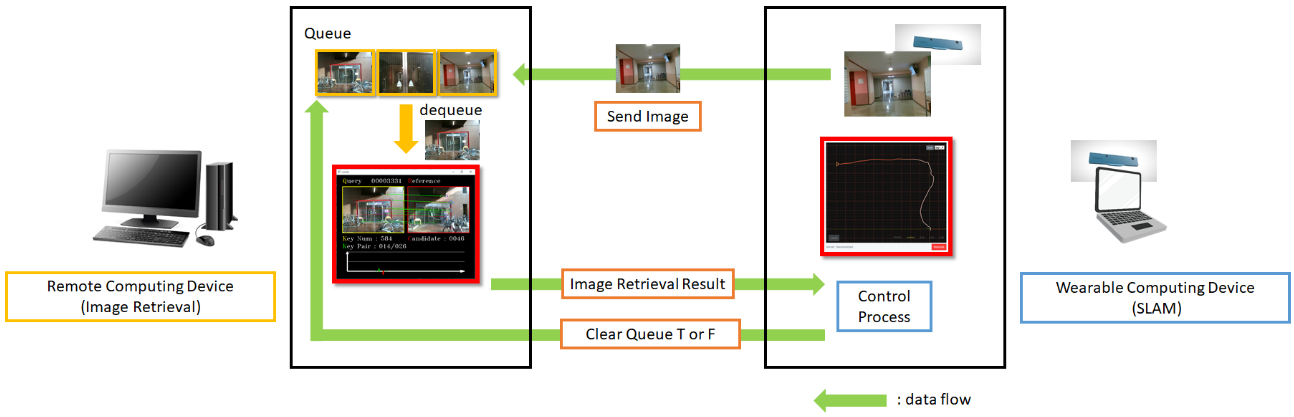


Figure 3. Proposed Architecture

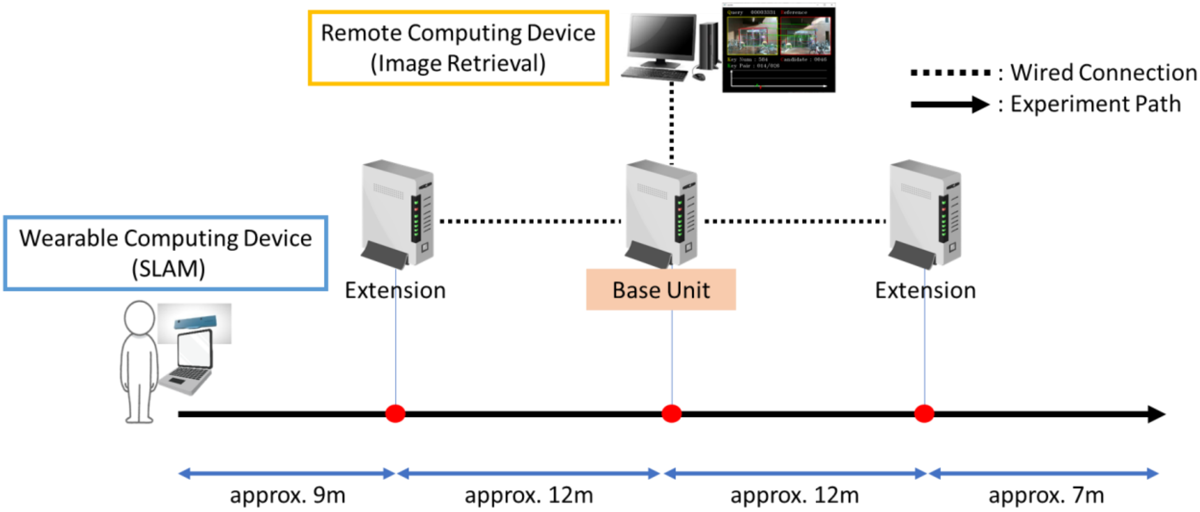


Figure 4. Wireless Experiment Setting

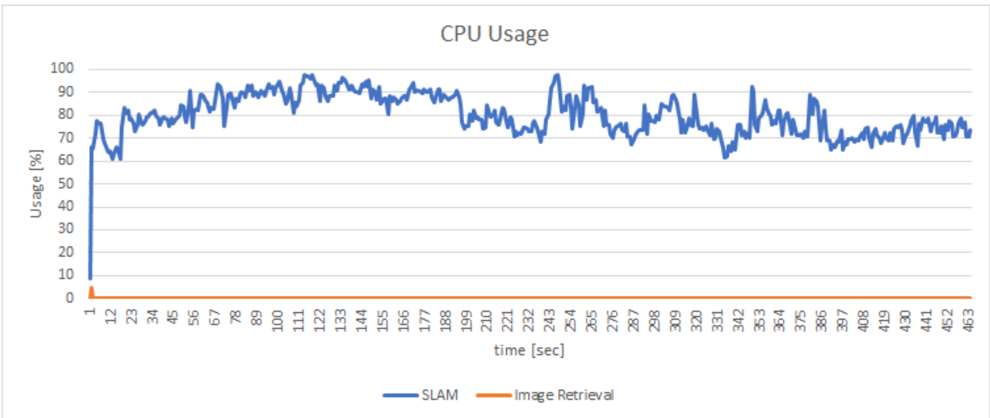
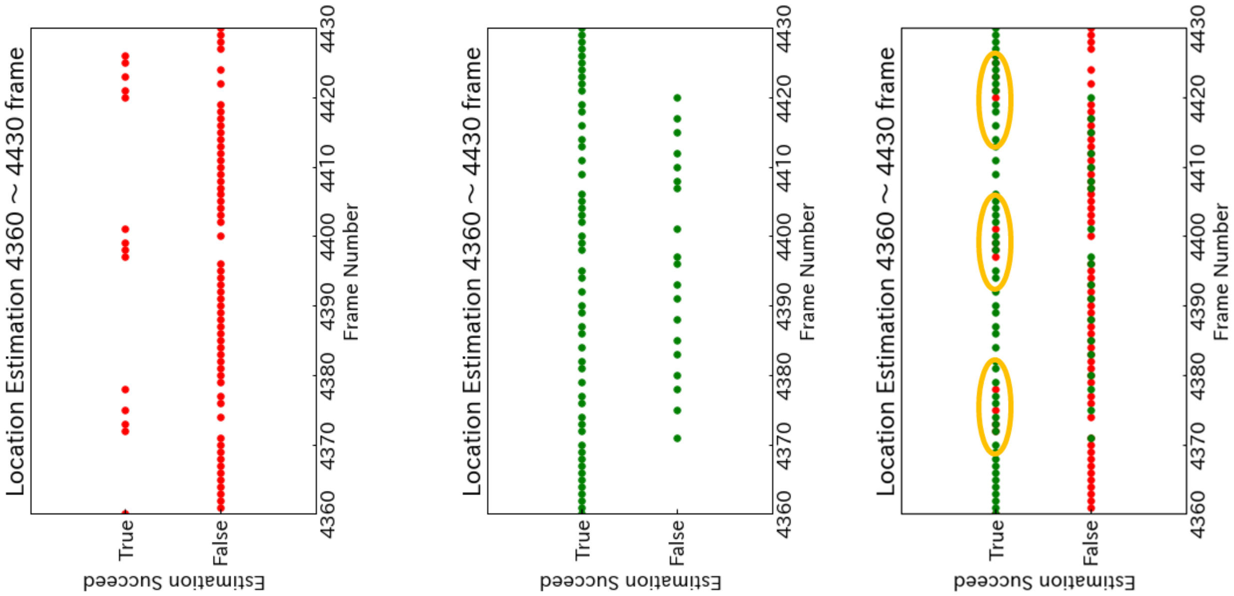


Figure 5. CPU Usage of Image-Retrieval and SLAM on Single Computer

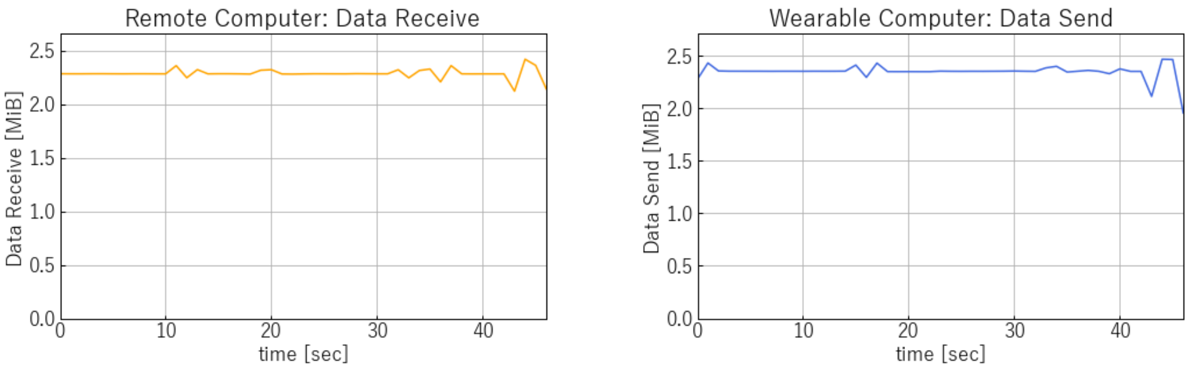
(a) Frame: 300 (b) Frame: 2400 (c) Frame: 4400 (d) Frame: 8900

Figure 6. Video Frames on Long Path (Wired Experiment)



(a) Image-Retrieval (b) SLAM (c) Image-Retrieval + SLAM

Figure 7. Successful Frame Numbers (4360 – 4430 frame, wired experiment)

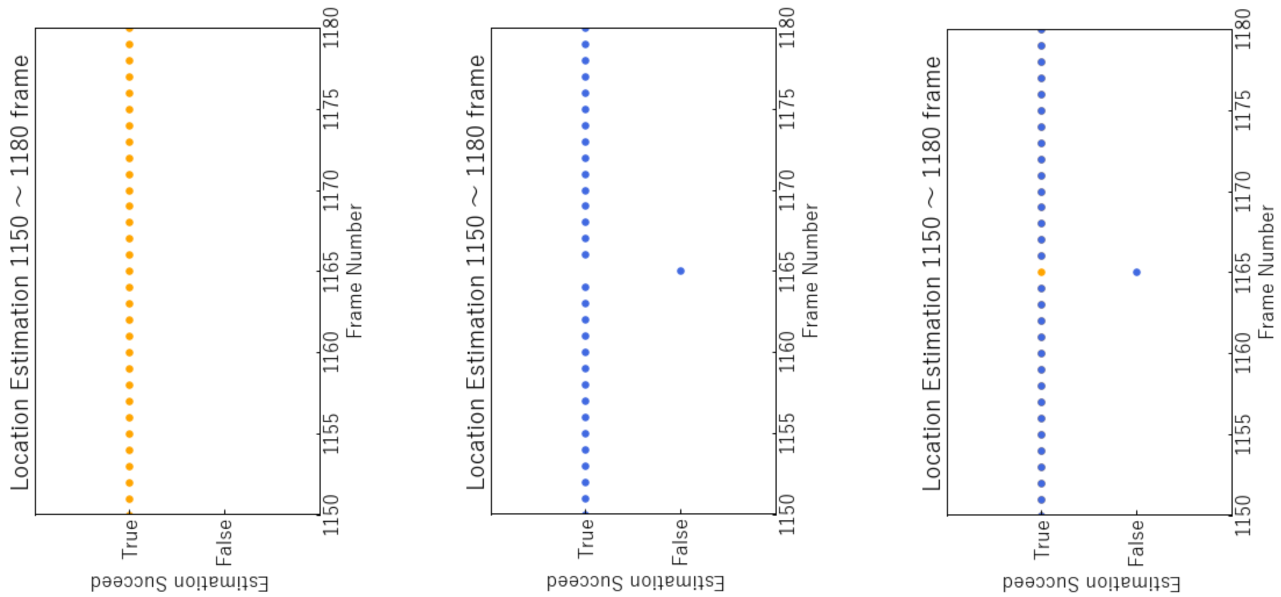


(a) Remote Computing Device (b) Wearable Computing Device

Figure 8. Data Traffic (Wireless Experiment)

(a) Frame: 10 (b) Frame: 100 (c) Frame: 430 (d) Frame: 800

Figure 9. Video Frames on Short Path (Wireless Experiment)



(a) Image-Retrieval (b) SLAM (c) Image-Retrieval + SLAM

(b) Figure 10. Estimation Succeed (1150 – 1180 frame, wireless experiment)

5. CONCLUSION

Even when the location estimation was unsuccessful at the SLAM side, the other process, the image retrieval approach, helped to offer the location information to the user. Thus the proposed system can realize the hybrid location estimation by using two computing devices, each of which executes the image retrieval and SLAM respectively.

This work was supported by KAKENHI 17H01773 and 18H03480.

REFERENCES

[1] Chahine G., Pradalier C., "Survey of Monocular SLAM Algorithms in Natural Environments," 15th Conference on Computer and Robot Vision, 345-352 (2018).
[2] Gouveia B, Portugal D, Silva D, Marques L, "Computation Sharing in Distributed Robotic Systems: A Case Study on SLAM," IEEE Transactions on Automation Science and Engineering, 12(2), 410-422 (2015).
[3] Hugh D.W., Tim B., "Simultaneous localization and mapping: part I," IEEE Robotics and Automation Society, 13(2), 99-110 (2006).
[4] Kazuho K., Itaru K., Yoshinari K., "Image Based Location Estimation for Walking Out of Visual Impaired Person," Studies in health technology and informatics, 2017(242), 709-716 (2017).
[5] "Intel®RealSense™Camera ZR300 | Intel®Software", Intel®Software Developer Zone, https://software.intel.com/en-us/realsense/zr300 (23 October 2019)
[6] "Intel®RealSense™for Linux - SLAM Library: Developers Guide," Intel®RealSense™for Linux – SLAM Library, https://software.intel.com/sites/products/realsense/slam/developer_guide.html (23 October 2019)