Performance Evaluation of 6DoF Localization on Mobile Devices

Chandra Kiran Viswanath Balusu
cbalusu@gmu.edu
George Mason University
Fairfax, Virginia, USA

Bharath Chandra Nimmala
bnimmala@gmu.edu
George Mason University
Fairfax, Virginia, USA

ABSTRACT

The rapid advancement of mobile device technology has driven the need for accurate and efficient localization in three-dimensional (3D) space. Six Degrees of Freedom (6DoF) localization is a promising technology that addresses this challenge by providing precise position and orientation information. In this research project, we aim to evaluate the performance of 6DoF localization on mobile devices such as accuracy and latency. The evaluation of 6DoF (six degrees of freedom) localization on mobile devices is crucial for improving the accuracy and reliability of localization algorithms, which can lead to the development of more effective techniques for 3D space localization. This assessment is particularly important for applications that require precise and dependable localization, ensuring their feasibility and effectiveness. Advancements in this research area can have significant implications for computer vision, robotics, and spatial computing, expanding researcher's understanding of how to enable 3D sensing and spatial awareness in various mobile device applications and settings.

KEYWORDS

6DoF localization, Performance Evaluation, COLMAP, Structure-from-Motion(SfM)

ACM Reference Format:
Chandra Kiran Viswanath Balusu and Bharath Chandra Nimmala. 2023. Performance Evaluation of 6DoF Localization on Mobile Devices. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

1.1 Statement of the Problem, Significance, and Goals

6DoF localization is a technology that enables the identification of an object's position and orientation in three-dimensional space, offering six degrees of freedom, which comprise three for translation and three for rotation. This technology finds applications in diverse fields, such as robotics, virtual reality, and mobile device tracking. 6DoF localization enables robots to navigate more precisely, enhances the immersive experience in virtual reality, and improves location tracking and augmented reality on mobile devices. Evaluating the performance of 6DoF localization[8] on mobile devices involves assessing its accuracy, speed, and reliability. To achieve this, various metrics can be utilized. Some of the metrics used for evaluating the performance of 6DoF localization on mobile devices include Localization accuracy, Latency, Robustness, Power consumption, etc.,

The assessment of 6DoF localization on mobile devices is significant as it has the potential to enhance the precision and dependability of localization algorithms[2] and lead to the creation of more effective techniques for localizing in 3D space. This evaluation is essential for various applications that demand accurate and dependable localization, ensuring that these applications are feasible and effective. Progress in this research area can have far-reaching consequences for fields like computer vision, robotics, and spatial computing, deepening the researcher's comprehension of how to enable 3D sensing and spatial awareness in different mobile device applications and environments.

1.2 Background

A significant amount of research was conducted on 6DoF Localization on mobile devices, especially related to performance evaluation[8] . For example, A study conducted in 2016, "Real-time 6DoF Tracking for Mobile Augmented Reality"[10], introduced a 6DoF tracking system for mobile augmented reality applications, which was evaluated based on its tracking accuracy, latency, and power consumption. Results from the evaluation indicated that the proposed system is appropriate for real-time applications as it offers the necessary tracking accuracy and low latency, with minimal power consumption. Another study conducted in 2017, 6DoF Object Tracking based on 3D Scans for Augmented Reality Remote Live Support"[5], assessed the performance of various 6DoF tracking techniques that could be employed for mobile augmented reality applications. The focus was on comparing feature-based and model-based techniques and evaluating their accuracy, robustness, and computational efficiency.

Evaluating the performance of 6DoF localization[8] on mobile devices involves several methods, and researchers have used different techniques for this purpose. Accuracy of localization is a primary focus, but other factors such as latency, robustness, and power consumption are also evaluated. Evaluation methods can be either quantitative, involving measures like mean error, precision, and recall, or qualitative, relying on user feedback and subjective evaluations. With the advancement of sensors and processing power in mobile devices, the performance of 6DoF localization[8] on such devices is predicted to improve. Future research on 6DoF localization[2] is likely to concentrate on developing more precise and dependable algorithms, along with integrating multiple techniques and sensors to improve the technology's performance. Additionally, efforts may be made to design systems that consume less energy and increase the technology's ability to operate in different environments.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2 SOFTWARE'S USED

- **COLMAP[9]:** COLMAP is a computer vision software package for 3D reconstruction and image-based localization, developed by Johannes L. Schönberger and Jan-Michael Frahm. It can be used for a variety of applications, including robotics, augmented reality, and virtual reality. The software takes a set of 2D images as input and produces a 3D point cloud and a textured mesh as output. It uses feature detection and matching techniques to find correspondences between images, and then estimates the camera poses and the 3D structure of the scene using geometric and photometric constraints. COLMAP supports a wide range of camera models, including pinhole, fisheye, and panoramic cameras, and can handle large datasets with hundreds of thousands of images. It also provides tools for filtering and refining the reconstruction, as well as for exporting the results to other software packages.
- **PyCOLMAP:** PyCOLMAP is a Python package that provides a Python interface for COLMAP, a popular open-source software for computer vision that allows you to perform 3D reconstruction from images. With PyCOLMAP, you can use the COLMAP algorithms from within Python, which allows you to automate and customize the 3D reconstruction process.
- **Hloc[6]:** Hierarchical localization is a technique used in computer vision and robotics to locate an agent (e.g., a robot or a person) in an environment using a pre-built map. The environment is divided into multiple levels of increasing granularity, which reduces the search space and increases accuracy. Building the map involves constructing a 3D model of the environment using sensors, which is divided into levels. Localizing the agent involves matching sensory data with the map at the coarsest level first and refining the estimate at finer levels. Hierarchical localization has applications in various fields, enabling accurate and efficient localization in complex and dynamic environments.
- **FastAPI:** FastAPI is a web framework for building APIs in Python 3.7+ that is fast, modern, and easy to use. It uses asynchronous programming to handle multiple requests at once, and provides built-in support for OpenAPI and JSON Schema for automatic API documentation and testing. FastAPI is intuitive and easy to learn, supporting standard HTTP methods and features like dependency injection, background tasks, and security measures like OAuth2 and JWT tokens. FastAPI is gaining popularity for its high performance and modern features.
- **aiofiles:** aiofiles is a Python library that allows developers to work with files asynchronously, providing a simple API for opening, closing, reading, and writing files, as well as working with file descriptors. It enables file operations to be performed without blocking the event loop, making it useful for applications with multiple simultaneous file operations, such as web applications. aiofiles supports binary and text modes and provides functions for working with file paths and directories. It integrates well with asyncio libraries and frameworks and can improve the performance and efficiency of applications that rely on file I/O operations.
- **aiopath:** aiopath is a Python library that offers an asynchronous interface for path-related operations in file systems. It has a simple API that is similar to the standard os.path API, with functions for joining, splitting, and normalizing paths, and working with file and directory names. It enables path-related operations to be performed asynchronously, making it beneficial for web applications that need to access file system paths simultaneously. aiopath also supports paths that point to remote file systems and can integrate well with other asyncio libraries and frameworks. Overall, aiopath can enhance the performance and efficiency of path-related operations in asynchronous Python applications.
- **uvicorn:** Uvicorn is an asynchronous web server for Python designed to handle high-concurrency HTTP requests and other protocols. It is built on top of the asyncio library and is capable of handling thousands of requests per second with low latency and resource usage. Uvicorn supports features like WebSockets, HTTP/2, and UNIX sockets, and works with other ASGI-compatible frameworks such as FastAPI, Starlette, and Quart. It is easy to use and configure, providing a simple command-line interface and support for configuration via environment variables, command-line options, and configuration files.
- **Unity Engine[3]:** The Unity Engine is a cross-platform game engine widely used for developing video games, simulations, and interactive applications. It supports 2D and 3D graphics with a range of tools for creating and manipulating 3D models, textures, and animations, as well as scripting using C#. Unity provides a powerful editor, libraries, plugins, and third-party extensions. It also supports collaboration and version control for teams working on large-scale projects. Unity's accessibility and versatility have made it popular among indie developers and large game development studios alike.

3 PRELIMINARY TESTING

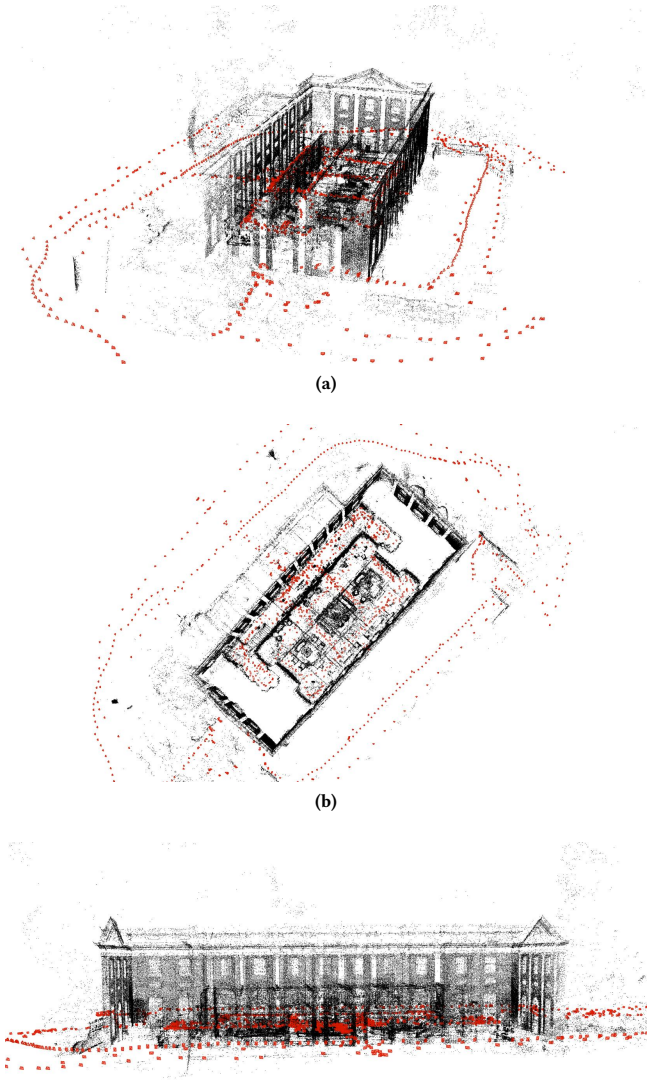
We have conducted our initial testing using the dataset which contains 1273 high-resolution images of the interior and exterior of "Graham" memorial hall at UNC Chapel Hill and generated the spatial map of the hall using COLMAP. The generated map has 407114 points in the point cloud. The generated map can be found in **Figure 1** in page 3.

4 DATASET COLLECTION

We have captured 279 aerial images of the George Mason University Field House located at 4501 University Dr, Fairfax, VA 22030 using DJI mini 2 drone.

5 METHODOLOGY

In order to evaluate the performance of 6DoF localization using Hloc[7] on mobile devices, we need a client-server REST API architecture as compared to mobile devices because of the computing power constraint.



(a)

(b)

(c)

Figure 1: Spatial Map generated from the dataset of Graham Hall.

The design of the REST API server is as follows:

- The REST API server is based on FAST API framework which is written in Python programming language that leverages asyncio concurrency module that increases the speed of the application.
- As we are using Hloc package written in Python, we extended it from the source code by creating a Git submodule in the Git repository.
- In order to facilitate multiple datasets and outputs, the REST API is designed to handle multiple sessions that uses UUID[4] which are stored in the underlying SQLite3[1] to differentiate sessions.The end points can be seen in the **Figure 2**

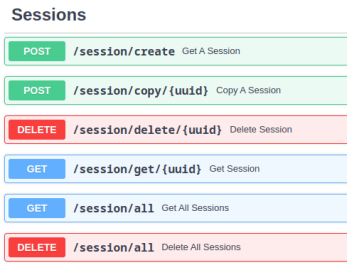


Figure 2: REST API Session Handling Endpoints

- To ensure security to the designed REST API, we require apikey to make calls to the API.
- To upload the dataset, we have an endpoint and there is another endpoint to upload the SfM generated in COLMAP. The end points can be seen in the **Figure 3**

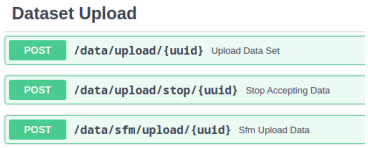


Figure 3: Dataset and SfM Upload Endpoints

- For feature extraction and mapping, we designed two endpoints where feature extraction and matching uses one endpoint and map generation uses another endpoint. The end points can be seen in the **Figure 4**

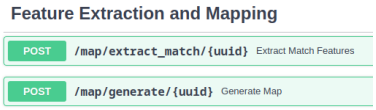


Figure 4: Feature Extraction, Matching and Mapping

- For the Localization of the Query images, there is one endpoint which returns the pose estimations after the localization and the Latency.
- There are few more endpoints that streams the Point cloud to the client which are in the Development phase.

We are using the source code of localization unity application developed by Yiyang Shi, which was provided to us by Nan Wu. We are currently modifying some of our endpoints to work with this application seamlessly.

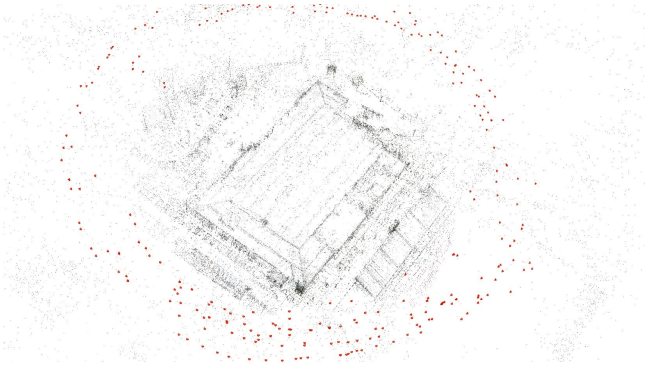
6 RESULTS ACHIEVED TO DATE

Using the dataset of the GMU field house which we collected, we ran the SfM pipeline multiple times with different configurations in order to achieve the best results. The results are as follows:

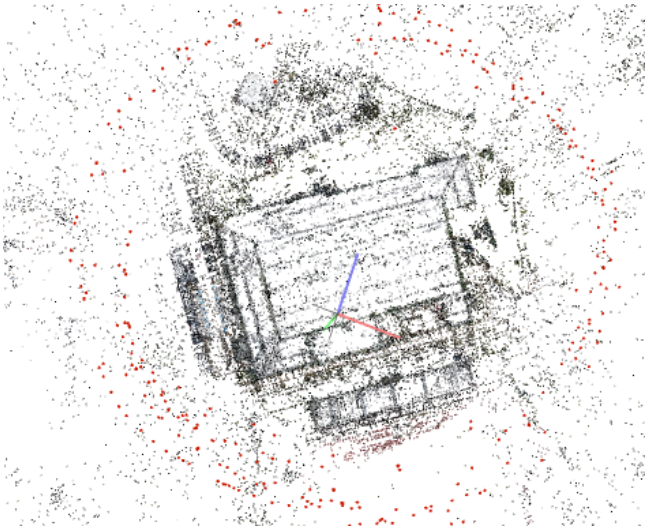
6.1 Spatial Maps Generated Using Different Configurations

The below shown figures are spatial maps that were generated using different configurations. The configurations are mentioned in the figure captions.

Configuration 1: *The below two figures belong to SfM, generated using SuperPoint Aachen for Extraction and SuperGlue Neural Net for Matching with SfM pairs from NETVLAD Global Feature Extractor.*



(a)



(b)

Figure 5: Point Cloud of the Reconstruction with 96971 Points

Configuration 2: *The below figure belong to SfM, generated using Sift Extraction with 16384 maximum features from each image and Exhaustive Matching within the COLMAP.*

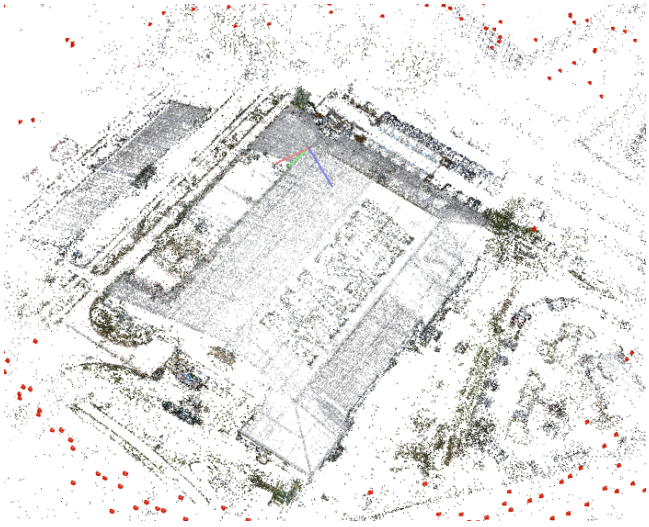


Figure 6: Point Cloud of the Reconstruction with 167275 Points

Configuration 2: *The below figure belong to SfM, generated using Sift Extraction with 24576 maximum features and Exhaustive Matching within the COLMAP.*

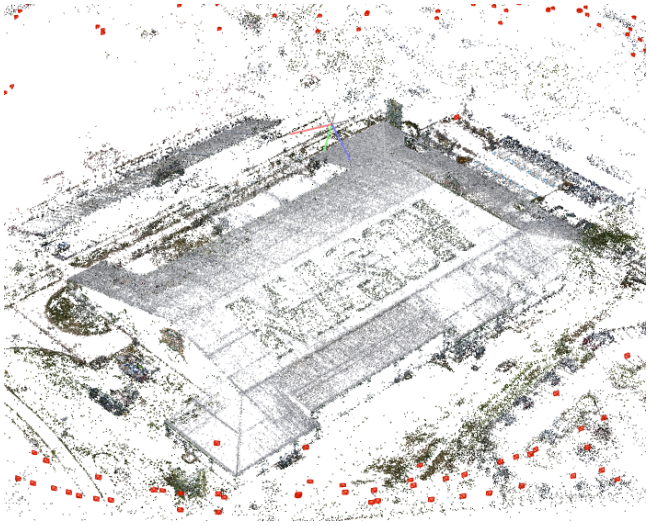
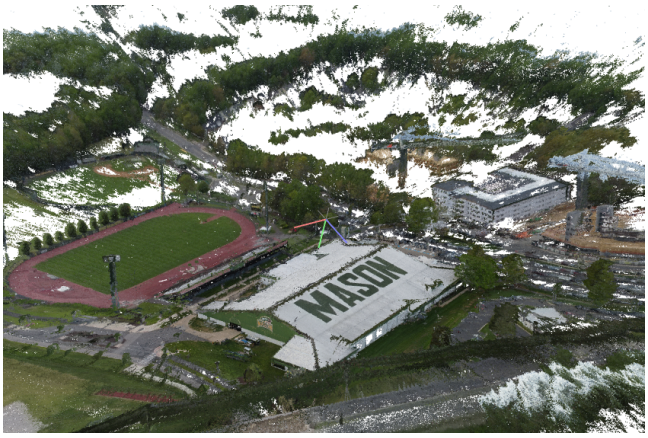


Figure 7: Point Cloud of the Reconstruction with 307829 Points

From the above three cases we can say that SIFT is more robust when compared to Superpoint Extraction and moreover when the number of features that are being extracted increases, we were able to see a more dense point cloud each time. SIFT Extraction and Matching took less time compared to different configurations.
Now that we have a sparse SfM with cameras that have intrinsic values, we then proceeded with the Dense Reconstruction.

The Dense Reconstruction that we generated has 42207561 colored points



(a)

(b)

(c)

Figure 8: Dense Point Cloud of the Reconstruction with 42207561 colored points

6.2 Latencies Observed While Localizing New Query Images

We took 4 Query Images of the Field House and Passed it to REST API for evaluation

(a) 56.4KB

(b) 335.7KB

(c) 1.6MB

(d) 3.6MB

Figure 9: Query images that were used

Table 1: Latencies Observed while Localizing New Query Images

Query size	Extractor	Matcher	Latency
56.4 KB	SIFT	NN-Mutual	15005.703ms
56.4 KB	SuperPoint	SuperGlue	19587.808ms
335.7 KB	SIFT	NN-Mutual	12885.49.ms
335.7 KB	SuperPoint	SuperGlue	27846.842ms
1.6 MB	SIFT	NN-Mutual	14341.984ms
1.6 MB	SuperPoint	SuperGlue	30920.418ms
3.6 MB	SIFT	NN-Mutual	18486.24ms
3.6 MB	SuperPoint	SuperGlue	48983.89ms

7 TECHNICAL CHALLENGES FACED

- As our image dataset was huge, initially we had few issues regarding the computation power as our PC was unable to process and generate a spatial map of the dataset. We solved this problem by running it in an advanced PC with high computational power
- During the dataset collection, we had to choose a building in George Mason University which doesn't have much glare and other environmental factors such as trees, moving things etc., This process took us multiple attempts and initially, we captured images of Eaglebank Arena and collected the image dataset, but the spatial map generated was incorrect because of the symmetric nature of the arena. Then we moved to capture and collect the dataset of GMU Field house, which we think is one of the best building in George Mason University free of glare and other environmental factors.
- During performance evaluation, we got few metrics which are incorrect/inaccurate. So, we had to perform the performance evaluation multiple times which were time-consuming

8 FUTURE WORK

- Currently, we are working on calculating the Accuracy of the generated spatial map.
- We are also developing our mobile application using Unity. Initially, we had less knowledge about the Unity application development, so this took us some time to learn about the framework and start the application development process.
- After the application is developed, we will integrate our mobile application with the REST API for the localization process.
  We will complete the aforementioned project works before the final project submission deadline.

9 CONCLUSION

This research project has focused on evaluating the performance of 6DoF localization on mobile devices, with a particular focus on measuring accuracy and latency. Through our research, we feel that this evaluation plays a crucial role in a variety of applications that require precise and reliable localization, ensuring the practicality and efficiency of these applications.

10 ACKNOWLEDGEMENTS

We thank Dr. Bo Han and Nan Wu from the department of Computer Science at George Mason University. We would also like to thank our classmates and friends for their support and advice given on this research project. Finally, we would like to thank George Mason University for supporting us with available resources.

REFERENCES

[1] Grant Allen and Mike Owens. 2010. SQLite Design and Concepts. (01 2010). https://doi.org/10.1007/978-1-4302-3226-1_5
[2] Ronald Clark, Sen Wang, Hongkai Wen, Niki Trigoni, and Andrew Markham. 2016. Increasing the efficiency of 6-DoF visual localization using multi-modal sensory data. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. 973–980. https://doi.org/10.1109/HUMANOIDS.2016.7803390
[3] Afzal Hussain, Haad Shakeel, Faizan Hussain, Nasir Uddin, and Turab Ghouri. 2020. Unity Game Development Engine: A Technical Survey. *University of Sindh Journal of Information and Communication Technology* 4 (10 2020).
[4] Paul Leach, Michael Mealling, and R. Salz. 2005. A Universally Unique IDentifier (UUID) URN Namespace. (01 2005).
[5] Jason Rambach, Alain Pagani, Michael Schneider, Oleksandr Artemenko, and Didier Stricker. 2018. 6DoF Object Tracking based on 3D Scans for Augmented Reality Remote Live Support. *Computers* 7, 1 (2018). https://doi.org/10.3390/computers7010006
[6] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2018. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. *CoRR* abs/1812.03506 (2018). arXiv:1812.03506 http://arxiv.org/abs/1812.03506
[7] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2019. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*.
[8] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. 2018. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*. IEEE Computer Society, United States, 8601–8610. https://doi.org/10.1109/CVPR.2018.00897 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018 ; Conference date: 18-06-2018 Through 22-06-2018.
[9] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
[10] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. 2010. Real-Time Detection and Tracking for Augmented Reality on Mobile Phones. *IEEE Transactions on Visualization and Computer Graphics* 16, 3 (2010), 355–368. https://doi.org/10.1109/TVCG.2009.99