SoftwareX 17 (2022) 100931

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

SfM Flow: A comprehensive toolset for the evaluation of 3D reconstruction pipelines

Davide Marelli *, Simone Bianco, Gianluigi Ciocca

DISCo - Department of Informatics, Systems and Communication, University of Milano-Bicocca, viale Sarca 336, 20126 Milano, Italy

ARTICLE INFO

Article history:
Received 25 February 2020
Received in revised form 2 August 2021
Accepted 1 December 2021

Keywords:
Blender add-on
Structure from Motion (SfM)
3D reconstruction evaluation
Synthetic data generation

ABSTRACT

We present *SfM Flow*, a Blender add-on that provides a toolset for the evaluation of three-dimensional reconstructions obtained form images. 3D reconstruction is increasingly becoming popular and, to date, many techniques are available to perform it. Choosing which technology to use for a specific 3D reconstruction task can be resource and time-consuming. By using this tool it is possible to create images of a virtual 3D scene, perform 3D reconstructions starting from the generated images using Structure from Motion pipelines, and evaluate the accuracy of the obtained 3D reconstruction. The evaluation is carried out comparing the 3D reconstruction with the virtual scene's geometry, that constitutes an exact ground truth, without the need for complex setup and dedicated hardware for the acquisition of a real scene. Furthermore, *SfM Flow* includes support for different lighting, depth of field, and motion blur setups, thus allowing to stress the 3D reconstruction pipelines under common critical conditions.

Code metadata

Current code version	v1.0.3
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_51
Code Ocean compute capsule	N/A
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	Python 3.7, GLSL
Compilation requirements, operating environments & dependencies	Blender 2.80.75+ for Linux, macOS or Windows. Optional: ExifTool 10+
If available Link to developer documentation/manual	https://github.com/davidemarelli/sfm_flow/wiki
Support email for questions	davide.marelli@unimib.it

Software metadata

Current software version	v1.0.3
Permanent link to executables of this version	https://github.com/davidemarelli/sfm_flow/releases/tag/v1.0.3
Legal Software License	MIT License
Computing platforms/Operating Systems	Linux, macOS, Windows
Installation requirements & dependencies	Blender 2.80.75+, ExifTool 10+
If available, link to user manual - if formally published include a reference to the publication in the reference list	https://github.com/davidemarelli/sfm_flow/wiki
Support email for questions	

1. Motivation and significance

During the last years, there has been a growing interest around the use of three-dimensional reconstruction in various fields

* Corresponding author.
E-mail addresses: davide.marelli@unimib.it (Davide Marelli), simone.bianco@unimib.it (Simone Bianco), gianluigi.ciocca@unimib.it (Gianluigi Ciocca).

https://doi.org/10.1016/j.softx.2021.100931

ranging from professional to casual use. The 3D reconstruction process captures the geometry and appearance of a single object or an entire scene. It finds use in many areas such as computer graphics and animation, virtual and augmented reality, video game assets creation, diagnostic imaging, Computer-Aided Design (CAD), virtual tours, and mobile 3D reconstruction apps. Various software, algorithm, and hardware solutions are available to perform 3D reconstructions; among them are popular the passive 3D reconstruction methods that make use of classical bi-dimensional RGB images, without the need of using specific hardware.

In this work, we focus on Structure from Motion (SfM) [1–4], which is a passive 3D reconstruction technique that given a set of images portraying the same scene from different viewpoints, can recover the pose of the cameras used for acquisition, and reconstruct the geometry as a sparse point cloud. The SfM pipeline is made up of different processing steps. Various algorithms are available for each of these steps, and it is thus possible to build different solutions to fit specific needs. Choosing between different methods usually requires performing some test reconstructions of a 3D scene and comparing the results with the geometry ground truth. Such a ground truth should have a high precision and is usually acquired with high-end devices such as 3D scanners.

In the literature, evaluation of SfM pipelines is often made on data specifically acquired to test the designed pipeline and algorithms [5]. The data is not always made available to the research community making it difficult to perform comparisons. Although available datasets exist (e.g. [6]) they may lack some characteristics that are needed to highlight the peculiarities of different 3D reconstruction solutions. The need for ad-hoc real scenes as reference models for the evaluation process is restrictive and not always feasible.

For these reasons we propose *SfM Flow*, an add-on for Blender that allows rapid and efficient evaluation and comparison of different 3D reconstruction pipelines. *SfM Flow* allows the generation of datasets to stress the pipelines under different conditions of lighting, multiple camera effects, and scene complexity. The data are generated from synthetic scenes that provide a known geometry and allow an easy, and precise, evaluation of the reconstruction.

To the best of our knowledge, SyB3R [7] is the only similar tool supporting synthetic dataset creation for 3D reconstruction. Recently, a few rendering engines started providing API to interact with them [8,9], mainly for machine learning tasks and none of them is specifically designed for 3D reconstruction. Existing synthetic datasets [10,11] are usually generated using rendering engines and application-specific automation scripts that lack flexibility. To exploit the features of *SfM Flow*, it is usually necessary use multiple software to manually build and render a synthetic dataset, run a 3D reconstruction pipeline and perform a reconstruction evaluation. *SfM Flow* supports the complete workflow in a single package.

2. Software description

SfM Flow is an add-on for Blender that focuses on providing an easy to use toolkit for the evaluation of 3D reconstructions performed starting from images. This add-on covers both the steps required for image generation and for reconstruction evaluation. In order to facilitate the use of the tool, each phase of the evaluation is made available in the user interface as a separate tool. The images are rendered from a custom virtual 3D scene, thus allowing the simulation of a variety of acquisition setups without requiring the use of dedicated hardware for the acquisition of the real scene(s).

2.1. Software architecture

SfM Flow is an add-on for the 3D modeling and rendering software Blender, and thus its architecture is mainly constrained by the coding style of Blender. *SfM Flow* is structured as a python module in which the main components are split into sub-modules. An "operators" sub-module implements the main functionalities as Blender's operators, the "panels" sub-module groups the user interface elements. The "utils" module contains the shared code used across multiple add-on's functionalities. Finally, an asset folder contains the textures used by the add-on during the initialization step to create the concrete looking ground of the scene. In this folder are also present a template flags file used by the Theia [12] 3D reconstruction library and a camera sensor size database used by OpenMVG [13].

Some parts of *SfM Flow* can be easily modified to introduce support for specific use cases.

The scene setup and initialization logic is implemented in the "SFM_OT_init_scene" operator that can be modified to add support for additional setups. The "SFM_OT_animate_camera" and "SFM_OT_animate_sun" classes respectively define the animations for the camera and the sunlight lamp; these can be also extended to provide additional animation types.

SfM Flow currently supports the N-View Match (NVM) and Bundle (.out) 3D reconstruction exchange formats; other formats can be supported by extending the "ReconstructionBase" class. *SfM Flow* automatically discovers and loads implementations located in the same directory of the base class.

SfM Flow supports token substitution for the custom pipeline commands and the Theia flags template file; both can be extended to support additional tokens in the "replace_tokens" function defined in the "run_pipelines.py" file.

2.2. Software functionalities

The main functionalities of *SfM Flow* can be grouped by scope as scene setup, data generation, 3D reconstruction pipeline execution, and 3D reconstruction evaluation. *SfM Flow* provides tools for the complete workflow. With the aim of having an higher flexibility, each group of functionalities also works as standalone. In this section we illustrate the functionalities in the order in which they are used for a complete workflow.

2.2.1. Scene initialization

The first set of tools concerns the 3D scene initialization. These operations support the user during the definition of the 3D scene that will be used for the subsequent steps of data generation and reconstruction evaluation. In this phase, a virtual 3D scene is created by defining the portrayed objects, the environment setting, the lighting setup, as well as the image acquisition viewpoints.

After choosing the main subject of the scene, the environment can be set up in different ways by selecting a scene type and a lighting condition through the "Initialize current scene" functionality. The available scene types are *floor* and *hemisphere*. The first one adds a concrete looking floor to the scene. The latter includes the scene in a smooth hemisphere with the objects placed on its floor. The lighting options available are *sky & sun* and *point lights*. The first one creates a procedural sky and places a sun lamp in the scene to simulate an outdoor setup. The *point lights* option places four point-lights around the scene to provide uniform illumination.

It is also possible to skip one or both the setting of the scene type and the setting of the light, and use an existing or a custom setup. Finally, if the *initialize camera* flag is enabled, the intrinsic camera calibration parameters are set to the default values. In any case, the rendering engine is switched to Cycles

and some render default values are applied. Objects added to the scene by this toolset will be placed in a new collection named "SfM_Environment".

The "Animate camera" functionality simplifies the camera animation process by providing tools for the camera path setup, to allow images acquisition portraying the scene from different viewpoints. The type of animation and its duration can be customized; it is also possible to use random camera positions around the computed points to simulate non-perfect acquisition setup. Multiple animations can be chained to obtain a complex acquisition setup besides the possibility to define custom motion paths. Predefined animation types include: helix, hemisphere, circle, and multiple circles moving the target viewpoint upwards. *SfM Flow* also provides a way to see all the acquisition positions of the camera to show at a glance the animation of the camera in the 3D viewport.

Similarly, if the scene is initialized with the *Sky & Sun* lighting setup, it is possible to animate the sun constraining it to follow a semi-circular path around the scene. The sun animation can be randomized to simulate acquisition in different hours of the day.

Thank to the flexibility of Blender, the outcome of each of these operations can be further manually manipulated to create a custom setup to fit specific needs.

2.2.2. Data generation

The second group of tools allows the generation of the synthetic images and additional data necessary for the execution and evaluation of the 3D reconstruction.

A custom rendering tool generates the synthetic images and camera ground truth information. Besides the color profile, the output format for the images can be chosen among JPEG, PNG, BMP, and AVI. Additionally, is possible to apply effects such as motion blur and depth of field (DoF) during image rendering. The motion blur effect can be applied to random frames using user-specified probability and shutter time. For a realistic DoF simulation, the camera focus distance is automatically set to the first object intersection along the camera's look-at direction during the camera animation setup. EXIF metadata are set for each image saved as JPEG or PNG file; please note that this functionality requires ExifTool [14] version 10 or above. The path to the tool is configurable in the *SfM Flow*'s preferences.

The rendering process also generates two Comma-Separated Values (CSV) files. A file named "scene.csv" describes the acquisition setup. The second one ("cameras.csv") contains information about camera position, camera rotation, camera look-at direction, depth of field, motion blur, and sun lighting position (if any) for each image. For further details about the CSV files, refer to the user manual. It is also possible to export the geometry ground truth as a Wavefront OBJ file. The exported file excludes by default the environment objects that are part of the "SfM_Environment" collection. It is also possible to export only the current object selection as geometry ground truth.

2.2.3. Pipeline execution

3D reconstruction pipelines can be run directly from the user interface. *SfM Flow* provides direct support for some incremental SfM pipelines such as COLMAP [2] (version 3.5), OpenMVG [13] (version 1.5), Theia [12] (version 0.8), and VisualSFM [15] (version 0.5.26); commands for other custom pipelines are configurable in the user preferences. Before the execution, the user can specify a reconstruction workspace in which the add-on will create a separate folder for each pipeline where to save the reconstruction and the execution log.

2.2.4. Reconstruction evaluation

The reconstruction evaluation is carried out following the procedure defined in [11]. After importing a reconstruction through *SfM Flow*, it is possible to proceed in the evaluation process by registering the reconstructed point cloud to the scene's geometry. This can be done using either the integrated Iterative Closest Point (ICP) [16] implementation or a registration matrix provided by an external tool. The ICP based algorithm uses by default only 25% of the point cloud for the alignment process and allows a maximum number of iterations defined as 1% of the size of the point cloud. Keep into consideration that the ICP procedure does not adjust the scale of the reconstruction, and the final quality of the registration strongly depends on the initial alignment. For this reason, it is recommended to perform a preliminary manual alignment.

A point cloud filtering functionality takes into account the presence in the reconstruction of points that are not part of the ground truth. This tool can filter the point cloud before the ICP registration, discarding points that are more distant than a given threshold from the ground truth. Once the point cloud has eventually been filtered, the alignment tool performs ICP registration of the remaining points.

Finally, the reconstruction evaluation tool generates an evaluation report file in two different formats, text and CSV; the first one is a human-readable version of the latter. Both files contain information about the reconstructed point cloud and the camera poses evaluation. The first part is an evaluation of the geometry in terms of euclidean distance between the reconstructed 3D points and a dense point cloud sampled over the ground truth of the virtual 3D model(s). This evaluation can be performed either on the filtered point cloud or on the full one. Currently, *SfM Flow* provides only the commonly used cloud-to-cloud euclidean distance metric. However, depending on the use case, other metrics such as cloud accuracy, completeness, point density, and roughness may be more suitable. The reconstruction evaluation module is designed to be easily extendable to fit specific needs by implementing other metrics. The second part evaluates camera poses in terms of camera position distance and orientation differences. The position is evaluated using the euclidean distance between the reconstructed camera position and the ground truth one. The orientation difference is computed using the angle of the rotation transformation needed to align the reconstructed camera to its corresponding ground-truth, forcing them to the same position. Finally, the camera orientation is evaluated in terms of the non-oriented angle between look-at vectors of each pair of reconstructed and ground truth camera pose. The camera pose evaluation also reports the percentage of images used by the reconstruction pipeline. All evaluations report minimum, maximum, mean and standard deviation values of each metric considered.

Please note that saving the Blender project or performing undo/redo operations after importing a reconstruction is currently not supported and will possibly lead to errors.

2.2.5. Command line execution

The rendering process of complex scenes is resource-intensive and is often performed on servers. With this scenario in mind, *SfM Flow* comes with additional command line parameters to enable execution from scripts. Such parameters allow rendering with all the possible combinations of effects described in Section 2.2.2 and also post-rendering export of camera poses and scene acquisition setup CSV files ('scene.csv' and 'cameras.csv' files). For further information about the command-line usage, refer to the user manual.

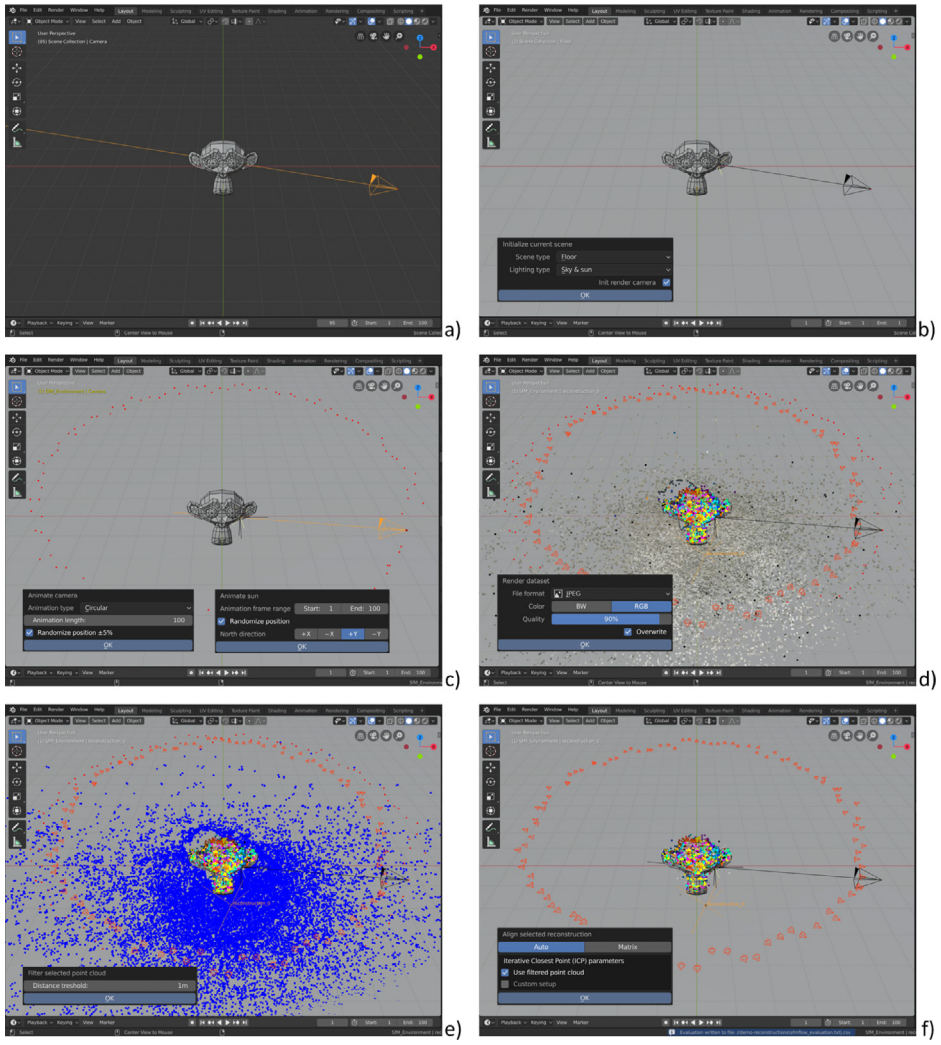


Fig. 1. Illustrative example of the software usage: (a) Initial scene; (b) 3D scene initialization; (c) Camera and sun animation setup; (d) Import 3D reconstruction result; (e) Filter reconstructed point cloud; (f) 3D reconstruction fine alignment.

3. Illustrative examples

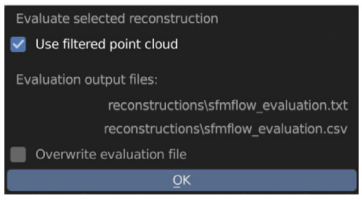
Here below are described the steps for a simple use case: a single object is the subject of the dataset, and COLMAP is used to perform the reconstruction.

1. Create a new project and place an object in the scene. Add a camera to acquire the images that will compose the dataset. An example of initial setup is shown in Fig. 1a.
2. Initialize the scene by adding a ground surface and a lighting setup through the "Initialize current scene" operator (Fig. 1b).
3. Animate the camera to acquire images from different points of view. In Fig. 1c it is shown the case of circular animation around the center of the scene. If necessary, also animate the Sun lamp movement.
4. Render the images using the provided operator. EXIF metadata are added to images if supported by the chosen export file format.
5. Run a 3D reconstruction, select COLMAP from the drop-down list and start it using the "Run 3D reconstruction" operator.
6. Once the reconstruction is ready, import the N-View Match file (NVM) (Fig. 1d) and manually scale and align the reconstructed point cloud to the object(s) in the 3D scene.
7. Filter the reconstructed point cloud using the filter functionality (Fig. 1e) to discard points whose distance is beyond a threshold. Discarded points are shown in blue by default.
8. Use the "Align selected reconstruction" to perform fine registration of the point cloud to the virtual scene. This functionality is visible in Fig. 1f. If the registration process is not satisfying, repeat it with a better manual registration/filtering or use different parameters for fine alignment.
9. Evaluate the reconstructed point cloud using the "Evaluate selected reconstruction" operator. Reconstruction evaluation follows the method defined in [11]; results will be written in file *sfmflow_evaluation.txt*. Evaluation dialog and sample results are visible in Fig. 2.

The functionalities of *SfM Flow* are further explained in Section 2.2.

4. Impact

With *SfM Flow* we provide a fast way to test and stress the 3D reconstruction pipelines using synthetic images, and allow the simulation of a variety of scene setups. By using synthetic data it is possible to overcome the limitations imposed by the need for real 3D models acquisition, which is tricky, requires




```
Project: demo.blend
Evaluation of Reconstruction model 'reconstruction_0' (internal name 'reconstruction_0')
Scene measurement system: METRIC
Scene length unit: METERS
Point cloud evaluation:
   used filtered point cloud: True
   filter threshold: 0.500
   full cloud size: 36034
   evaluated cloud size: 4973 (13.8%)
   discarded points count: 31061
   distance mean: 0.054m
   distance standard deviation: 0.052m
   distance min: 0.002m
   distance max: 0.646m
Camera poses evaluation:
   cameras count: 100
   reconstructed camera count: 100 (100.0%)
   distance mean: 0.248m
   distance standard deviation: 0.016m
   distance min: 0.220m
   distance max: 0.279m
   rotation difference mean: 0.529°
   rotation difference standard deviation: 0.013°
   rotation difference min: 0.504°
   rotation difference max: 0.550°
   look-at direction difference mean: 0.391°
   look-at direction difference standard deviation: 0.137°
   look-at direction difference min: 0.040°
   look-at direction difference max: 0.533°
```

Fig. 2. 3D reconstruction evaluation.

dedicated hardware, and complex setup making it not suitable in many situations. *SfM Flow* is meant to be used by researchers and final users for the evaluation of the existing 3D reconstruction pipelines as well as to provide support for the development of new 3D reconstruction methods. The availability of *SfM Flow* as an add-on for Blender integrates its functionalities with a well known and widely adopted 3D modeling software, making it easily usable by experts of the 3D reconstruction field as well as by newcomers. Finally, *SfM Flow* is written in Python, a popular programming language that encourages its further development including the addition of new functionalities and its adaptation to specific needs. Version 1.0.0 of *SfM Flow* was used to create the IVL-SYNTHSFM-v2 [17] dataset; its preliminary version was previously used to develop the dataset of the research article [11], which also defines the 3D reconstruction evaluation method used in *SfM Flow*.

5. Conclusions

We presented *SfM Flow*, a toolkit for the evaluation of Structure from Motion 3D reconstruction techniques. *SfM Flow* includes tools for the setup of a 3D virtual scene, the generation of the images, the execution of reconstruction pipelines, and the evaluation of the obtained results in terms of camera poses and geometry accuracy. Moreover, the tools included in *SfM Flow* can render images with a combination of different lighting camera effects to simulate critical conditions, stress the reconstruction pipelines, and assess their performances. Blender was chosen as the modeling and rendering software based on its popularity and its extension capabilities; this allowed us to provide an easy to use toolkit that encourages its usage and the integration of other functionalities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Partially supported by FooDesArt: Food Design Arte - L'Arte del Benessere, CUP (Codice Unico Progetto - Unique Project Code): E48I16000350009 - Call "Smart Fashion and Design", cofunded by POR FESR 2014–2020 (Programma Operativo Regionale, Fondo Europeo di Sviluppo Regionale - Regional Operational Programme, European Regional Development Fund).

References

[1] Özyeşil O, Voroninski V, Basri R, Singer A. A survey of structure from motion. Acta Numer 2017;26:305–64.
[2] Schönberger JL, Frahm J-M. Structure-from-motion revisited. In: Conference on computer vision and pattern recognition; 2016. p. 4104–4113.
[3] Wu C. Towards linear-time incremental structure from motion. In: 3D vision-3DV 2013, 2013 international conference on. IEEE; 2013, p. 127–34.
[4] Snavely N, Seitz SM, Szeliski R. Photo tourism: exploring photo collections in 3D. In: ACM transactions on graphics (TOG). 25, (3):ACM; 2006, p. 835–46.
[5] Stathopoulou EK, Remondino F. Open-source image-based 3D reconstruction pipelines: Review, comparison and evaluation. In: 6th international workshop lowcost 3d–sensors, algorithms, applications; 2019. p. 331–338.
[6] Schonberger JL, Hardmeier H, Sattler T, Pollefeys M. Comparative evaluation of hand-crafted and learned local features. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 1482–1491.
[7] Ley A, Hänsch R, Hellwich O. Syb3r: A realistic synthetic benchmark for 3d reconstruction from images. In: European conference on computer vision. Springer; 2016, p. 236–51.
[8] Qiu W, Zhong F, Zhang Y, Qiao S, Xiao Z, et al. Unrealcv: Virtual worlds for computer vision. In: Proceedings of the 25th ACM international conference on multimedia; 2017. p. 1221–1224.
[9] Unity Technologies. Unity perception package. 2020, https://github.com/Unity-Technologies/com.unity.perception.
[10] Broekman A, Gräbe PJ. PASMVS: A perfectly accurate, synthetic, path-traced dataset featuring specular material properties for multi-view stereopsis training and reconstruction applications. Data Brief 2020;32:106219. http://dx.doi.org/10.1016/j.dib.2020.106219, URL https://www.sciencedirect.com/science/article/pii/S2352340920311136.
[11] Bianco S, Ciocca G, Marelli D. Evaluating the performance of structure from motion pipelines. J Imaging 2018;4(8). http://dx.doi.org/10.3390/jimaging4080098.
[12] Sweeney C. Theia multiview geometry library: Tutorial & reference. 2015, http://theia-sfm.org.
[13] Moulon P, Monasse P, Marlet R. OpenMVG. An open multiple view geometry library. 2013, https://github.com/openMVG/openMVG.
[14] Harvey P. ExifTool by phil harvey: Read, write and edit meta information. 2003, https://www.sno.phy.queensu.ca/~phil/exiftool/ (Accessed: Nov 26, 2019).
[15] Wu C. VisualSFM: A visual structure from motion system. 2011, http://ccwu.me/vsfm/.
[16] Besl PJ, McKay ND. Method for registration of 3-D shapes. In: Sensor fusion iv: control paradigms and data structures, Vol. 1611. International Society for Optics and Photonics; 1992, p. 586–606.
[17] Marelli D, Bianco S, Ciocca G. IVL-SYNTHSFM-v2: a synthetic dataset with exact ground truth for the evaluation of 3D reconstruction pipelines. Data Brief 2019;105041.