# UAV Simulation for Object Detection and 3D Reconstruction Fusing 2D LiDAR and Camera

**Daniel Amigo[1], Jesús García[2], José M. Molina[3], Jorge Lizcano[4]**

**Abstract** Currently it is hard to develop UAV in civil environments, being simulation the best option to develop complex UAV missions with AI. To carry out useful AI training in simulation for real-world use, it is best to do it over a similar environment as the one a real UAV will work, with realistic objects in the scene of interest (buildings, vehicles, structures, etc.). This work aims to detect, reconstruct, and extract metadata from those objects. A UAV mission was developed, which automatically detects all objects in a given area using both simulated camera and 2D LiDAR, and then performs a detailed scan of each object. Later, a reconstruct process will create a 3D model for each one of those objects, along with a geo-referenced information layer that contains the object information. If applied on reality, this mission ease bringing real content to a digital twin, thus improving, and extending the simulation capabilities. Results show great potential even with the current budget specification sensors. Additional post-processing steps could reduce the resulting artefacts in the export of 3D objects. Code, dataset, and details are available on the project page: https://danielamigo.github.io/projects/soco22/

**Keyword** Object reconstruction, LiDAR-camera fusion, UAV simulation, Object detection, AirSim

## Introduction

Unmanned Aerial Vehicles (UAV) are powerful tools capable of autonomously capturing the Earth at a given time. However, at present they are a technology that is neither smart nor robust enough to operate in cities along humans and are thus

[1] GIAA, University Carlos III of Madrid, Madrid, Spain, e-mail: damigo@inf.uc3m.es
[2] GIAA, University Carlos III of Madrid, Madrid, Spain, e-mail: jgherrer@inf.uc3m.es
[3] GIAA, University Carlos III of Madrid, Madrid, Spain, e-mail: molina@ia.uc3m.es
[4] GIAA, University Carlos III of Madrid, Madrid, Spain, e-mail: jorgelizgo@gmail.com

considered potential hazards. They are strongly legislated to restrict its use in habited areas[1], [2]. Eventually this risk will disappear as UAV get smarter. For this reason, work must be pursued in order to operate them safely around humans while performing complex and risky tasks [3].

Software in the loop (SITL) simulation is the easiest way for develop and test complex UAV missions with zero damage. AirSim[4] is one of the most popular UAV simulator. Based on the popular Unreal Engine (UE4), it provides high visual and customization capabilities, enabling to simulate sensors of the UAV such as lidar or cameras, or even to deploy multiple UAVs at the same time for swarm interactions. AirSim is compatible with the highly used for Pixhawk 4 flight controller (PX4) [5]. All these features make it perfect for testing UAV missions with Artificial Intelligence (AI) before applying them on the real world.

A common problem in AI is introducing data into a model that is not close to the data used in training. Using simulation, it is easy to solve this problem for real-world deployment. If the simulated mission environment is like the real world, the data captured by the simulated UAV will be similar too and therefore good for AI training a model that aims to perform in reality [6]. It is possible to fly over photo-realistic environments on AirSim, but those won't be as real environments. Ideally, the environment should be a digital representation of your own real environment, a digital twin. Creating them is a very complex and hard task [7]. It should recreate all static objects and habitual patterns of dynamics actors, such as people, vehicles, or animals. To generate digital twins of any location in the world the only feasible option is automatization using huge amounts of geolocated data.

This work is a further step of [8] and [9], where we detected a specific object and geolocate it to enhance a digital twin. Now using simulation, we first create data with an autonomous flight mission but also automatically create 3D representations of any environment object using only UAVs. With it, a further process can automatically add those objects on the digital twin, improving it for future simulation applications. The proposal tests realistic sensing using low-cost 2D LiDAR and an HD FPV camera. LiDAR sensor is used first to gather information of all the objects in a specific area. The UAV will then use its on-board capabilities to identify them and design a customized mission for each object detected, scanning it in detail with both sensors. After the flight, the 3D object reconstruction process fuses both sensors by coloring each LiDAR detection using the camera data and clean the data to get the object. Finally, it transforms the final point cloud into a 3D object. The knowledge extracted of each object (position, height and the other metadata as the object type generated by image and point cloud classifiers) is also stored in a GIS format for other future uses.

The results obtained are promising. The mission gathering component works as expected, perfectly identifying all objects in the mission, and performing a close and custom scanning for each object with both sensors. The 3D object reconstruction component results are good but could be improved with further post-processes or fusion with other algorithms. In any case, the proposed mission automatically successfully solves an existing problem, easing the generation of digital twins. In conclusion, it has been exemplified how a UAVs work designed in this simulation framework can be successfully developed, reducing the friction when performing it in real drones.

## Related works

This section briefly introduces several studies of other researchers regarding the 3D object reconstruction and of its detection using both point clouds and imagery.

Although there are many sensors [10], not all of them are suitable for use in current UAVs due to their size or weight. Object reconstruction can be performed only using camera data with algorithms as Structure from Motion (SfM). Its visual result is good, but the geometry is not reliable. It can also be done by LiDAR, generating precise geometry but lacking color or texture. Many researchers use drones for large areas reconstruction, but not specifically for object reconstruction. For example, [11] uses LiDAR to precisely map an excavated surface. With this approach no objects are scanned in detail, only a global view, so it is not ideal for digital twins [12]. [13] proposal is the only one found attempting to reconstruct a specific object with a UAV mission. They use a camera and SfM to reconstruct.

Although camera and LiDAR can work separately, it is typical they are fused for these tasks as they have high synergy. The image contains a lot of information with high detail and color, while the LiDAR is composed of lots of individual measurements, highly detailed in shape, but colorless. For example, [14] uses the point cloud from a Terrestrial Laser Scanning to improve the geometry of the point cloud obtained from SfM with a drone flight, obtaining very good results.

Despite the approach or the sensor, the goal is to generate only a 3D representation of an object, so it requires segmentation algorithms for discarding the information that does not belong to the object, as the floor, walls, or other objects. This task can be performed before the main reconstruction, by removing junk information from each raw data, or after generating the 3D point cloud, by detecting specific point cloud points not belonging to the object.

On the other hand, our proposal aims to detect objects in real time and at the end classify them in order to generate additional metadata. The object detection problem is a common task when dealing with point clouds. Clustering and segmentation algorithms can easily discriminate and group them to achieve the desired solution. The object classification approaches use deep learning to train convolutional networks for detecting patterns in images or point clouds relating a specific class. There are few researches, such as [15], that try to combine both in the same procedure. Image-based solutions are widely studied whereas the 3D solutions are relatively unexplored.

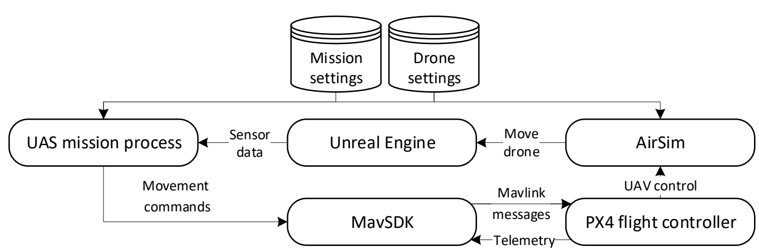



**Figure 1** – Simulation framework interconnection diagram

## Simulation Framework

Several components need to be combined in order to make this realistic simulation work correctly. This section explains the key aspects of each component and its logical connection with the others. A diagram illustrating the interactions between all components is provided on **Figure 1**.

First, it is needed to define the mission to perform; what will the UAV do, with which sensors and where. A JSON file defining the drones' characteristics must be provided to AirSim, the central element of the framework. The UAV mission is designed as fully autonomous, simulating a UAV on-board processor that receives all sensors data and send to the flight controller the movements commands when needed, according to the mission stage. The mission process needs to connect both with AirSim to retrieve sensors data capturing the Unreal Engine environment at a specific time and place, and with MAVSDK library to communicate with PX4, to send those custom movement commands but also to receive telemetry updates to make onboard missions adjustments.

Furthermore, we have the UE4 environment, built around the Cesium plugin for Unreal. It adds a digital twin of the whole Earth, with its texture formed by satellite images and a global elevation model. It also adds a global coordinate system allowing to match the PX4 coordinates with the digital twin, making it possible to simulate UAV flights over real locations on the Cesium virtual Earth. Note that it is only a template, it does not bring the actual objects and dynamic behaviors into the simulator by default.

## Proposed process

This section introduces the process to automatically detect and reconstruct 3D objects using LiDAR and camera fusion onboard of a UAV. The process is composed by two main blocks. The first, detailed in **Figure 2**, performs a completely autonomous UAV mission first detecting all objects in a specific area and then performing a custom mission to scan in detail each object. Then, detailed on **Figure 4**, an offline process creates the 3D mesh by colorizing the LiDAR point cloud and discarding all points which are not part of the object. The output is the mentioned 3D mesh of each object but also a GIS layer with object metadata.

The mission parameters must be defined manually, both drone settings with the JSON file explained earlier and the parameters of the algorithms to adjust the mission operation. It also requires the area where to scan for objects.

An illustration of the data gathering component is provided in **Figure 3**. The first step is to connect with AirSim and MAVSDK, and then to take-off. Once it is done, the mission starts, flying directly to the input area at constant altitude. When the UAV arrives at that position, it starts its sensors and starts the first part of the data gathering mission. It performs a sweep capturing all possible objects within the orbit with both LiDAR and camera. Then, the data is processed onboard to identify the objects placement and dimensions.

After this initial sweep, the process performs a LiDAR points analysis. First it merges all point clouds into one and applies the RANSAC algorithm to discard the floor points from the rest. Then, a DBSCAN algorithm is applied to cluster the point cloud in groups, identifying how many objects are and to know its positioning and dimensions. Using each group point cloud the process can design and perform a custom orbit around the object, from top to bottom, so both sensors capture it perfectly. Once the UAV performs the last orbit almost hitting the ground, it comes back to the origin point following the same initial path.

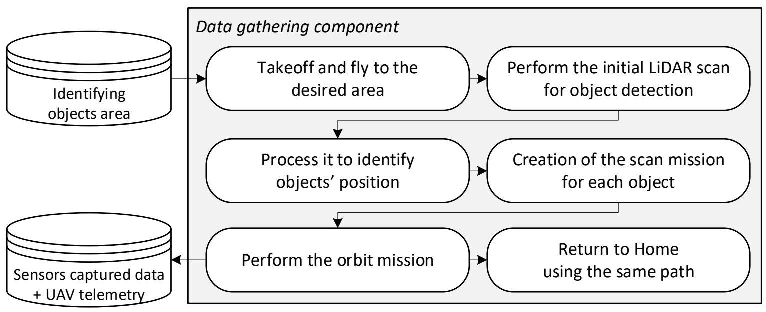



**Figure 2** – Data gathering component

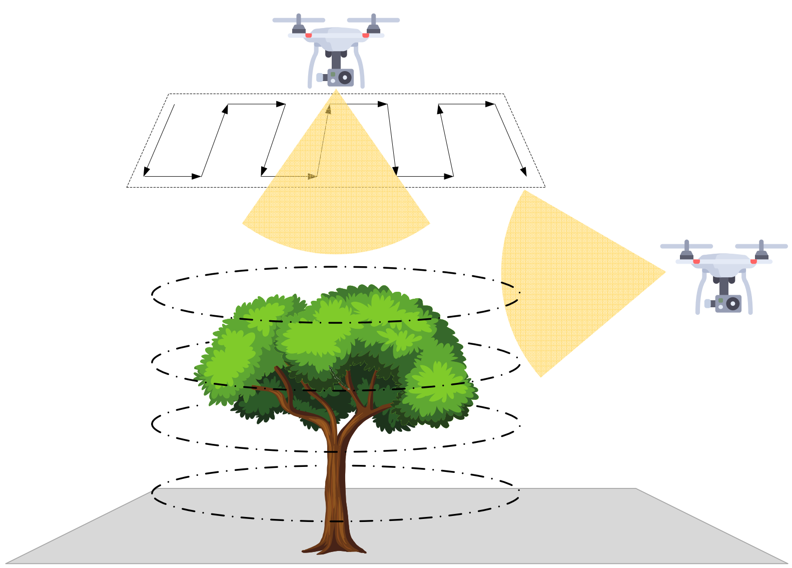


**Figure 3** – UAV mission illustration

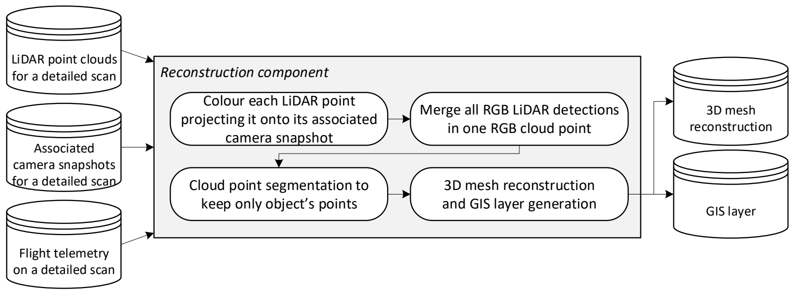



**Figure 4** – Reconstruction process

After all data is captured and the UAV is back, the 3D object reconstruction process is performed. The process gets each capture data at a specific time: a trio formed by the camera snapshot, LiDAR points and the UAV's position and rotation matrix. The aim is to project each one of those LiDAR points into the camera snapshot, obtaining a RGB color for the LiDAR point. To do that, the LiDAR tridimensional point is transformed to the camera coordinate system, and then inserted on the snapshot, using the intrinsic camera parameters.

With all LiDAR points colorized, the next step merges all of them into one point cloud. As done onboard, this detailed point cloud has noise. To discard those non-desired points, it performs first RANSAC algorithm to remove the floor and then DBSCAN to remove other points not from this object.

Finally, the process applies the Poisson Surface Reconstruction algorithm to transform the final colorized point cloud it to a final 3D mesh.

The process at the end also generates a GIS layer with metadata for further uses. Specifically, it adds the object positioning, dimensions, and orientation. It also contains the object type, a useful attribute for all kinds of future processes. It is calculated by applying two classifiers, one image-based, applied to all the snapshots captured during the detailed scan, based on VGG16 and trained with MicroImageNet dataset. The other is a 3D mesh classifier to classify the final 3D object. The classifier is based on PointNet++[16] and trained with CAD data from ModelNet40[17]. If both classifiers conclude the object class is the same, that value is added.

## Demonstration and evaluation

To demonstrate the potential of the developed system, an example mission of the whole process using the presented simulation framework is performed. It contains three objects in the same area: a sculpture, a building, and a car. They are separated from each other for around 10 meters, as illustrated on **Figure 6**(a) and **Figure 7**. Those will be detected using the sensors illustrated at **Figure 5**. They have the following characteristics: The camera (represented in blue) has 1920x1080 resolution, 120º horizontal FOV and it is placed as a FPV view, and the 2D LiDAR (represented in green): 10 rotations per second, 30.000 points per second, 165º FOV.

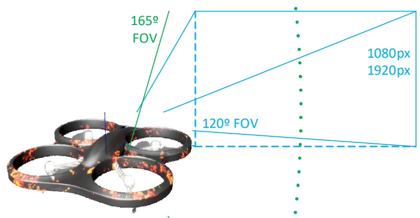
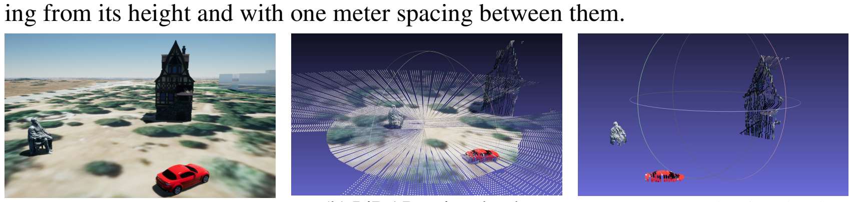



**Figure 5** – LiDAR (green) and camera (blue) installed on AirSim's UAV

First, the drone performs the object detection, going to the specified position, and making an orbit with a radius of 30 meters, obtaining **Figure 6**(b) colorized LiDAR point cloud. Then, the point cloud is processed as explained, obtaining three clusters shown in **Figure 6**(c). It is noteworthy that the building is not completely detected, which could lead to an inaccurate scanning mission design.

After it, the specific scans are performed with orbits at constant altitude, starting from its height and with one meter spacing between them.



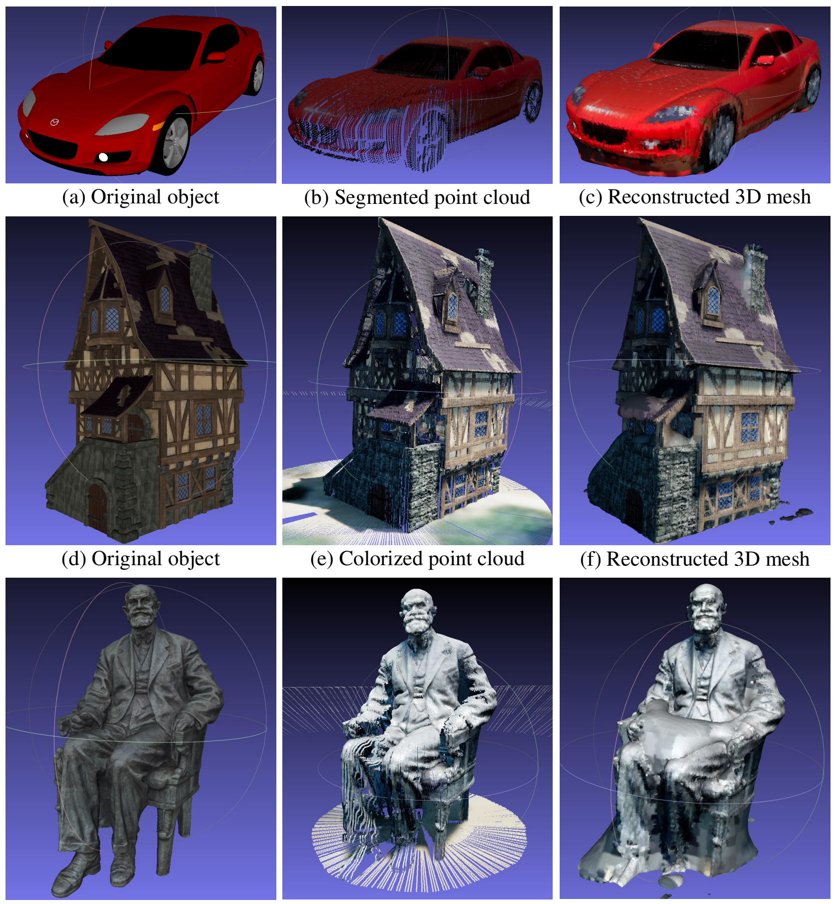
(a) Environment image (b) LiDAR point cloud (c) Segmented point cloud

**Figure 6** – Data gathering object detection example


| | |
|---|---|
| TypeObject | car |
| (Derived) | |
| (Actions) | |
| id | 0 |
| Height | 2.011823000000000 |
| Altitude | 900.004000000000019 |
| Yaw | 14.420000000000000 |
| Pitch | 0 |
| Roll | 0 |
| TypeObject | car |
| ImAsBuildi | 0.12222 |
| ImAsCar | 1.000000000000000 |
| ImAsSculpt | 0.02 |
| 3DAsBuildi | 0.02 |
| 3DAsCar | 0.67545165 |
| 3DAsSculpt | 0.02 |

**Figure 7** – Orbits and detected objects in GIS (left). Metadata of car GIS layer (right)

The results for each object are illustrated at **Figure 8**. A visual analysis reveals gaps in the rows of 2D LiDAR points, which may cause trouble to the mesh generation algorithm. Also, in (c) it is noticeable how the meshes are illuminationdependent, as the car's front has sunlight reflected in the snapshots which is then transferred to the 3D model. In (e) it is observed that LiDAR data is missing in the chimney, although it is fairly well solved in (f). In contrast, in the roof behind the stairs, Poisson Surface Reconstruction generates a kind of imaginary bubble. The same happens in (i). Finally, the bottom of all 3D models is less detailed. This may be justified by the rough cropping of the ground, causing the scan mission not to perform lower orbits.



(a) Original object (b) Segmented point cloud (c) Reconstructed 3D mesh

(d) Original object (e) Colorized point cloud (f) Reconstructed 3D mesh

(g) Original object (h) Colorized point cloud (i) Reconstructed 3D mesh

**Figure 8** – Demonstration 3D mesh generation

## Conclusions and perspectives

This work proves the potential of designing autonomous drone missions using a realistic simulation framework, which can facilitate the development of complex tasks with AI prior to their actual operation.

The data gathering components work well, detecting and performing a scanning mission using only onboard processing. However, the proposed solution is still at an early stage, and for actual deployment it should be more robust, for example avoiding obstacles during the mission or considering complex cases as whether the object is close to a wall, denying the drone the acquisition of data in such a perspective.

In addition, the proposed 3D reconstruction process works well but improvements are possible. Further work will be applied to quantify the results with different metrics for this colorizing LiDAR points approach. It should also be compared with alternatives such as SfM or deep learning approaches such as NERF, the best quality results may be obtained by combining several of them. Anyway, this way is a good alternative, as it requires low computational resources.

Other future developments must be seeking for a more robust object type classifier. Also, in using the object type in the 3D mesh generation, maybe as a post-processing. Lastly, to incorporate the 3d meshes automatically into the digital twin.

## Acknowledgement

This research was partially funded by public research projects of Spanish Ministry of Science and Innovation, references PID2020-118249RB-C22 and PDC2021-121567-C22 - AEI/10.13039/501100011033, and by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors, reference EPUC3M17.

## References

1. Grigoropoulos, N., Lalis, S.: Flexible Deployment and Enforcement of Flight and Privacy Restrictions for Drone Applications. In: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 110–117. IEEE, Valencia, Spain (2020)
2. Hildmann, H., Kovacs, E.: Review: Using Unmanned Aerial Vehicles (UAVs) as Mobile Sensing Platforms (MSPs) for Disaster Response, Civil Security and Public Safety. Drones. 3, 59 (2019). https://doi.org/10.3390/drones3030059

3. Kakaletsis, E., Symeonidis, C., Tzelepi, M., Mademlis, I., Tefas, A., Nikolaidis, N., Pitas, I.: Computer Vision for Autonomous UAV Flight Safety: An Overview and a Vision-based Safe Landing Pipeline Example. ACM Comput. Surv. 54, 1–37 (2022). https://doi.org/10.1145/3472288
4. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. ArXiv170505065 Cs. (2017)
5. Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 6235–6240. IEEE, Seattle, WA, USA (2015)
6. Alvey, B., Anderson, D.T., Buck, A., Deardorff, M., Scott, G., Keller, J.M.: Simulated Photorealistic Deep Learning Framework and Workflows to Accelerate Computer Vision and Unmanned Aerial Vehicle Research. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 3882–3891. IEEE, Montreal, BC, Canada (2021)
7. Cinar, Z.M., Nuhu, A.A., Zeeshan, Q., Korhan, O.: Digital Twins for Industry 4.0: A Review. In: Calisir, F. and Korhan, O. (eds.) Industrial Engineering in the Digital Disruption Era. pp. 193–203. Springer International Publishing, Cham (2020)
8. Amigo, D., Pedroche, D.S., García, J., Molina, J.M.: Automatic Individual Tree Detection from Combination of Aerial Imagery, LiDAR and Environment Context. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., and Corchado, E. (eds.) 16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021). pp. 294–303. Springer International Publishing, Bilbao, Spain (2022)
9. Amigo, D., Pedroche, D.S., García, J., Molina, J.M.: Automatic context learning based on 360 imageries triangulation and 3D LiDAR validation. In: 2021 24th international conference on information fusion (FUSION). p. 8 (2021)
10. Ghamisi, P., Gloaguen, R., Atkinson, P.M., Benediktsson, J.A., Rasti, B., Yokoya, N., Wang, Q., Hofle, B., Bruzzone, L., Bovolo, F., Chi, M., Anders, K.: Multisource and Multitemporal Data Fusion in Remote Sensing: A Comprehensive Review of the State of the Art. IEEE Geosci. Remote Sens. Mag. 7, 6–39 (2019). https://doi.org/10.1109/MGRS.2018.2890023
11. Ribeiro, L.G.: 3D Reconstruction of Civil Infrastructures from UAV Lidar point clouds. 71
12. Siebert, S., Teizer, J.: Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. Autom. Constr. 41, 1–14 (2014). https://doi.org/10.1016/j.autcon.2014.01.004
13. Mentasti, S., Pedersini, F.: Controlling the Flight of a Drone and Its Camera for 3D Reconstruction of Large Objects. Sensors. 19, 2333 (2019). https://doi.org/10.3390/s19102333
14. Luhmann, T., Chizhova, M., Gorkovchuk, D.: Fusion of UAV and Terrestrial Photogrammetry with Laser Scanning for 3D Reconstruction of Historic Churches in Georgia. Drones. 4, 53 (2020). https://doi.org/10.3390/drones4030053
15. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum PointNets for 3D Object Detection from RGB-D Data. ArXiv171108488 Cs. (2018)
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. ArXiv170602413 Cs. (2017)
17. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A Deep Representation for Volumetric Shapes. ArXiv14065670 Cs. (2015)