

INTERACTIVE 3D CITY VISUALIZATION FROM STRUCTURE MOTION DATA USING GAME ENGINE

Dany Laksono, Trias Aditya, Gondang Riyadi

Dept. of Geodetic Engineering, Universitas Gadjah Mada, Caturtunggal, Kec. Depok, Kabupaten Sleman, Daerah Istimewa
Yogyakarta 55281, Indonesia.

danylaksono@ugm.ac.id, takmid@gmail.com, gondang@ugm.ac.id

KEY WORDS: LOD1, SfM, 3Dfiers, Land Information, Unity3D, Mapbox

ABSTRACT:

Developing a 3D city model is always a challenging task, whether on how to obtain the 3D data or how to present the model to users. Lidar is often used to produce real-world measurement, resulting in point clouds which further processed into a 3D model. However, this method possesses some limitation, e.g. tedious, expensive works and high technicalities, which limits its usability in a smaller area. Currently, there exists pipeline utilize point-clouds from Lidar data to automate the generation of 3D city model. For example, 3dfier (<http://github.com/tudelft3d/3dfier>) is a software capable of generating LoD 1 3D city model from lidar point cloud data. The resulting CityGML file could further be used in a 3D GIS viewer to produce an interactive 3D city model. This research proposed the use of Structure from Motion (SfM) method to obtain point cloud from UAV data. Using SfM to generate point clouds means cheaper and shorter production time, as well as more suitable for smaller area compared to LiDAR. 3Dfier could be utilized to produce 3D model from the point cloud. Subsequently, a game engine, i.e. Unity 3D, is utilized as the visualization platform. Previous works shows that a game engine could be used as an interactive environment for exploring virtual world based on real-world measurement and other data, such as parcel boundaries. This works shows that the process of generating 3D city model could be achieved using the proposed pipeline.

1. INTRODUCTION

Presenting a 3D city model from real-world data is a challenging task. Various data acquisition methods could be employed to obtain accurate representations of cities components. Data acquisition methods such as UAV-based photogrammetry, LiDAR and ground-based acquisition are often used to obtain 3D data from real-world objects. Development of a 3D city model using LiDAR requires tedious and expensive efforts: Point-clouds obtained from LiDAR data should be filtered, classified and subsequently converted into 3D models (Popovic et al. 2017). However, this approach could be used to obtain detailed building models (e.g. LOD3), preserving the buildings' height and roof shape on the 3D model (Tse, Gold, and Kidner 2008).

For a less-detailed 3D city model focused on geospatial data visualization, the process could be more straightforward. LOD1 buildings could be quickly generated based upon building footprints, digitized from satellite imageries or mosaics from aerial photogrammetry. A 3D modelling software could be employed to extrude the digitized building footprints to form corresponding 3D Models. This approach, however, also possesses some limitations, since the height of the buildings could not be obtained directly and needs to be manually assigned by the operator for each building unit (Aditya and Laksono 2017).

3Dfiers (<http://github.com/tudelft3d/3dfier>) is a software capable of quickly generating LOD1 building models from 2D footprint data and LiDAR point-cloud data in LAS format. The software identifies building footprint as a 2D boundary from GIS data, and further determines individual building heights from the classified LiDAR point clouds. It then automatically extrudes each building footprints into the LOD1 model (i.e. flat roof) based on their corresponding height. Other features such

as water bodies, roads and vegetation could also be generated. The resulting 3D city model could be obtained rapidly and exported to other formats for visualization. Unfortunately, obtaining LiDAR dataset is quite expensive, and therefore is not suitable for smaller areas. In this case, structure from motion (SfM) method is often applied as an alternative to LiDAR to obtain point cloud data (Westoby et al. 2012).

3D data could be presented in an interactive environment using a game engine. The use of game engine, i.e. Unity3D for visualizing geospatial data, is well known (Buyuksalih et al. 2017). Geospatial data from multiple sources could also be integrated, and custom scripts could be utilized to produce interactive visualization of real-world data (Laksono and Aditya 2019).

The objective of this paper is to apply an alternative method to obtain point cloud data containing building heights for rapid visualization of the 3D city. Structure from Motion (SfM) is utilized to obtain point cloud of the study area in Solo City, Central Java. The software also provides classified point clouds to be used as input in 3Dfier. A game engine, i.e. Unity3D, is utilized as a visualization platform which capable of presenting the 3D city model as walkable environment. The pipeline could be reproduced in area where the availability of point-cloud data is limited.

2. RELATED WORKS

The objective of this paper is to apply an alternative method to obtain point cloud data containing building heights and further provide an interactive visualization of the 3D city. Malamboa and Hahn (2010) showed that LiDAR data could be used to produce various Level of Detail (LoD) from building data and produce a CityGML file from LiDAR point cloud. The point cloud from LiDAR could be used to extract buildings, including

roof shapes (Tse et al. 2008). Further, automated extraction of buildings based on elevation data from point cloud is proposed, and further implemented using LiDAR data in the case of Netherland (Kumar, Ledoux, and Stoter 2018). Structure from Motion have shown as a viable alternative to LiDAR point cloud for producing 3D models (Ingwer et al. 2015). Classification of point cloud data from SfM method is shown to be used for producing CityGML of LoD 1 building data. Utilization of SfM for producing point cloud data as source for 3D building extrusion could simplify the otherwise tedious works with LiDAR data (Williams 2018).

Game engine have long been used as an alternative for visualizing 3D geospatial data, although currently no native support for 3D geospatial data in the game engines (Buyuksalih et al. 2017). The recent introduction of Mapbox for Unity opens the possibilities to present geospatial data from real-world measurement into an interactive visualization platform built upon a game engine, e.g. Unity 3D. The interactive visualization could be presented as a game, where users could explore the 3D city on multiple platforms, such as web, Windows or Android application (Laksono and Aditya 2019).

3. METHODOLOGY

3.1 Data Acquisition and Study Area

The pipeline proposed on this paper involves utilization of some software for data acquisition, processing and editing, as well as visualization. The workflow and corresponding software used in the process are illustrated in Figure 1 as follow:

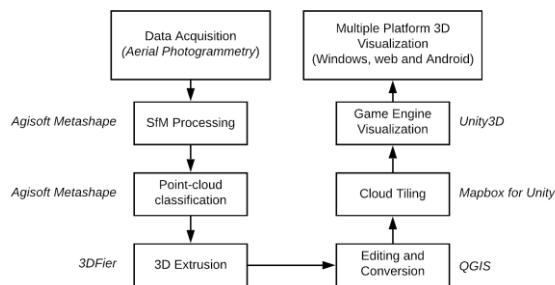


Figure 1. Workflow for 3D visualization

This research is conducted in Solo City, Central Java, Indonesia. The study area is situated in central part of the city, with area of approximately 1 km² around the city centre, which consist of several governmental, residential and historical buildings. Aerial photos of the study area were obtained using DJI Phantom Pro 3 Unmanned Aerial Vehicle (UAV). The data acquisition resulted in ~1300 photos of the study area. Aside from aerial photo, parcel data was obtained from the National Land Agency of Central Java Province, which would be used in the game engine visualization.

3.2 Structure from Motion

Structure from Motion (SfM) is a method capable of obtaining 3D data from a series of overlapping 2D images. SfM could be used to generate 3D points (i.e. Point Cloud) representing surface of objects captured on the images. Software such as Agisoft Metashape (www.agisoft.com) could be used to generate sparse and dense point clouds from series of overlapping photos.



Figure 2. Dense Point Cloud obtained from SfM Processing with Agisoft Metashape

The resulting point cloud (Figure 2) is comparable to point cloud obtained using LiDAR. However, the point cloud obtained from SfM method have different characteristics compared to point cloud obtained from LiDAR, thus additional processing should be employed to produce point cloud which could be used as an input for 3Dfier.

3.3 Point Cloud Classification

3Dfier accept point cloud input which comply to ASPRS standard for LiDAR point classes (e.g. buildings, ground, high vegetation). In order to obtain classified point cloud, the existing dense cloud data obtained from previous step should be processed prior to processing in 3Dfier. The dense point cloud itself should be processed to remove unwanted artefacts (e.g. noise from previous SfM method), and further process the result for classification. Figure 3 shows the classified dense point cloud processed in Agisoft Metashape. The figure shows that the result of automatic point cloud classification still contains misclassified point cloud (e.g. part of trees classified as buildings). The misclassified parts of the point cloud are edited manually to produce more accurate classified point cloud. Finally, the result is exported into LAS format for use in 3Dfier.

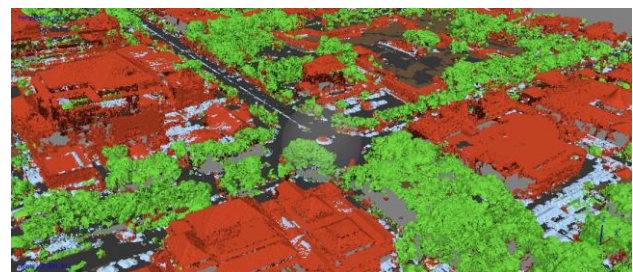


Figure 3. Classified point cloud obtained from Agisoft Metashape

3.4 Building Extrusion

3Dfier input could be categorised into two general categories: LAS point cloud which provide the elevation, and SQLite file which provide building footprints and other city objects. The LAS used is the classified point cloud resulted from Agisoft Metashape. The SQLite format of building footprints and other objects are acquired from digitizing orthophoto from the same aerial photo dataset of the study area using QGIS.

3Dfier processing was executed based on YAML configuration file, which point to the LAS point cloud and digitized SQLite file. The result of 3D extrusion is a CityGML file format, which should be edited prior to visualization in Unity3D (Figure 4).

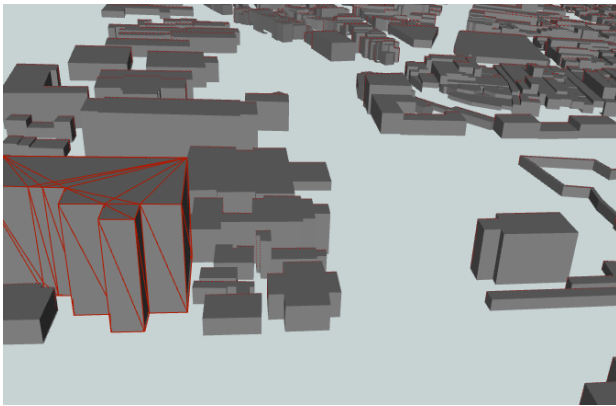


Figure 4. Extruded LoD1 Buildings from 3Dfier



Figure 6. Tiling of parcel data in Mapbox Cloud

3.5 Editing and File Conversion

CityGML from 3Dfier could readily be used in 3D city web visualization, such as 3DCityDB (Kolbe 2009). However, this paper proposed a different approach for visualizing 3D data, which is to utilize a game engine, i.e. Unity3D (Laksono and Aditya 2019). Thus, the CityGML file should be edited prior to visualizing the 3D model in a game engine.

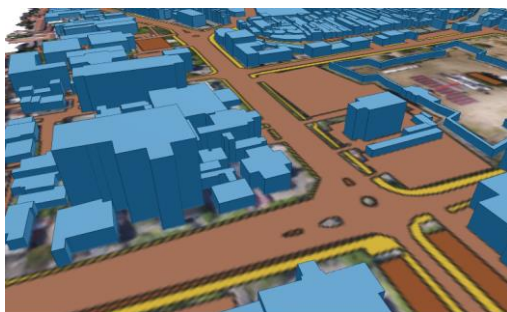


Figure 5. Editing and conversion of CityGML file using QGIS3

QGIS3 is used to perform editing and conversion of 3D data (i.e. CityGML file) as well as other datasets (e.g. parcel data) prior to visualizing into Unity game engine. The editing step is generally flatten-out the 3D data from CityGML and transfer the height attribute to 2D shapefile. QGIS version 3 is able to read CityGML format natively, and further conversion to SQLite could be conducted using only QGIS software. Figure 5 depict the 3D view of QGIS, showing the edited shapefile data and their respective elevation from CityGML 3D model.

3.6 Cloud Tiling

Game engines such as Unity3D or Unreal Engine does not natively support geospatial data. In order to be used in a game engine, 3D models should be converted into formats which could be read by the engine, such as FBX or OBJ, which has limited support for spatial data. Currently, Mapbox for Unity (www.mapbox.com/unity) provides support for utilizing 3D data in Unity3D game engine.

Using only shapefiles which contains 3D data (such as buildings elevation), Mapbox cloud would then provide a ‘tiling’ of these data for further used in Unity. Other data, such as Parcel shapefile (Figure 6), were also uploaded to Mapbox which convert the file into ‘tilesets’. Mapbox then provided urls which could be used in Unity3D for utilizing each dataset as a game object in the game engine.

3.7 Game Engine Visualization

Game engines such as Unity3D (www.unity3d.com) have been used in various cases for 3D visualization of real-world data (Buyuksalih et al. 2017). The ability to present a simulated environment makes the game engine an ideal platform for presenting 3D cities. The added value from a game engine for visualizing 3D cities is the ability to design interactions and user interface capable of performing various tasks not available in other 3D visualization platforms. For example, a game engine could be used to simulate players in a walkable 3D city, presenting real-world data and interacting with real-world attributes.

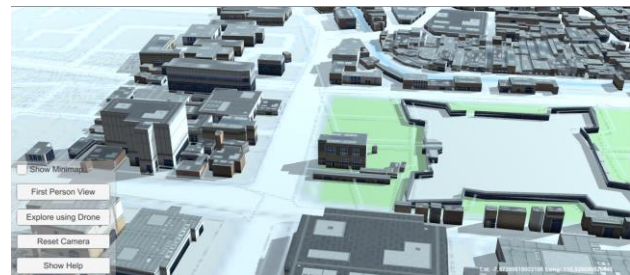


Figure 7. Interactive panel (bottom-left) used to control interactivity in Unity Game Engine

Another advantage of a game engine is its capability to produce multiplatform applications based on a single codebase. The resulting interactive visualization could be installed in desktop or mobile platform, enabling usability of such interactive 3D city model. However, most game engines lack the requirements for handling geospatial data, thus overlaying multiple sources of geospatial data preserving its position is also challenging. Mapbox for Unity (www.mapbox.com/unity) provides the capability to overlay multiple geospatial data into its real-world coordinates on top of slippy map tiles. Thus, it is possible to combine multiple geospatial data (e.g. land parcels) into a 3D city generated from multiple sources.

Laksono and Aditya (2019) utilized custom C# scripts for interactive 3D visualization of geospatial data in Unity3D game engine. The scripts enable exploration of the 3D environment, such as a 3D city model, so that users could walk or fly through a drone around the city. Mapbox for Unity also provides additional layers and C# scripts which could be used to interact with the layers, such as identifying a parcel’s attribute.

4. RESULT AND DISCUSSION

Interactive visualizations on multiple platforms were built using Unity Game Engine. Multiple layers are visible on the interactive visualization, such as parcel boundary, roads and drainage. Custom C# scripts were used to provide navigation and interactivity to the visualization, e.g. for First Person View. The visualization is then deployed to Windows, Web and Android platform.

Parcel data representing boundaries of land ownership from Land Office were overlaid onto the 3D city in Unity game engine to provide with an interactive query on the parcel's attributes. In addition to interactive query, users can benefit from a first-person view and drone view offered through the Unity game engine platform. The use of such 3D city visualization could support the 3D cadastral mapping and valuation application scenario. Targeted users are property seekers who look for information about affordable land and property rights and their restrictions that suit their investment plan.



Figure 8. Bird-eye view of the generated visualization

Here, users (e.g. future investors) can walk and inspect buildings and underlying parcel boundaries and their corresponding attributes interactively (Figure 8). Land right types and values, as well as building permits, can be explored by users in a walkable 3D city neighbourhood either via first-person or drone mode interactions. The result is a 3D interactive visualization of Solo City area in multiple platforms (i.e. Desktop, web and Android).

5. CONCLUSION

This paper provides an alternative workflow for rapidly visualizing 3D city model in an interactive environment. Utilizing SfM on aerial photo is more suitable for smaller area compared to LiDAR, while providing 3D models similar to those obtained using LiDAR. Automatic extraction of 3D building, such as utilizing 3Dfier, could be used to further speed the production of 3D model. Lastly, game engine such as Unity3D could be used to visualize a 3D City model while combining multiple sources of geospatial data. One of the advantages of using a game engine is to produce multiplatform visualization using a single codebase.

However, this proposed pipeline still contains some limitations. For example, in order to provide 3D model of buildings into the Unity3D game engine, a conversion to 2D data should be performed first. This of course could be accepted where the 3D models are LoD1 or extruded buildings and would not work for higher Level of Details. In the future, direct supports of CityGML or other 3D geospatial data format are expected, thus reducing the bottleneck of visualizing geospatial data in a game engine such as Unity3D.

ACKNOWLEDGEMENT

This research is supported by Indonesian Ministry of Research and Technology and Higher Education. The parcel data from Solo City area is provided by the National Land Agency of Solo, Central Java.

REFERENCES

- Aditya, T. and D. Laksono. 2017. "LOD 1 - 3D CityModel for Implementing Smartcity Concept." *ACM International Conference Proceeding Series* (2):136–41.
- Buyuksalih, I., S. Bayburt, G. Buyuksalih, A. P. Baskaraca, H. Karim, and A. A. Rahman. 2017. "3D Modelling And Visualization Based On The Unity Game Engine – Advantages And Challenges." *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-4/W4(October):161–66.
- Ingwer, Patrick, Fabian Gassen, Stefan Püst, Melanie Duhn, Marten Schällicke, Katja Müller, Heiko Ruhm, Josephin Rettig, Eberhard Hasche, Arno Fischer, and Reiner Creutzburg. 2015. "Practical Usefulness of Structure from Motion (SfM) Point Clouds Obtained from Different Consumer Cameras." P. 941102 in *Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2015*. Vol. 9411.
- Kolbe, Thomas H. 2009. "Representing and Exchanging 3D City Models with CityGML." Pp. 15–31 in *3D geo-information sciences*. Springer.
- Kumar, K., H. Ledoux, and J. Stoter. 2018. "Dynamic 3D Visualization of Floods: Case of the Netherlands." Pp. 83–87 in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. Vol. 42.
- Laksono, Dany and Trias Aditya. 2019. "Utilizing A Game Engine for Interactive 3D Topographic Data Visualization." *ISPRS International Journal of Geo-Information* 8(8):361.
- Malamboa, LM and M. Hahn. 2010. "LiDAR Assisted CityGML Creation." in *3rd Summer Applied Geoinformatics for Society and Environment*. Vol. 211.
- Popovic, Dragana, Miro Govedarica, Dusan Jovanovic, Aleksandra Radulovic, and Vlado Simeunovic. 2017. "3D Visualization of Urban Area Using Lidar Technology and CityGML." P. 042006 in *IOP Conference Series: Earth and Environmental Science*. Vol. 95.
- Tse, Rebecca, Christopher Gold, and Dave Kidner. 2008. "3D City Modelling from LIDAR Data." Pp. 161–75 in *Lecture Notes in Geoinformation and Cartography*.
- Westoby, M. J., J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 2012. "'Structure-from-Motion' Photogrammetry: A Low-Cost, Effective Tool for Geoscience Applications." *Geomorphology* 179:300–314.
- Williams, C. 2018. "Crowd-Sourced Cadastral Geospatial Information: Defining a Workflow from Unmanned Aerial System (UAS) Data to 3D Building Volumes Using Opensource Applications."