

A Strategy for Loading Oblique Photogrammetry Models and Multilayer Basemap Data

Weiye Zhang¹, Zhiqing Tang², Guangyang Zhu¹, Bing Dai¹, Zhuorong Song¹, Jun Li¹

¹College of Electronic Science, National University of Defense Technology, Changsha, China

²The 2nd Surveying and Mapping Institute of Hunan Province, Changsha, China

Email: zhangweiye17@nudt.edu.cn, tangzhiqingcut@163.com, zhuguangyang13@nudt.edu.cn, 530017656@qq.com, songzhuorong18@163.com, lijun@nudt.edu.cn

How to cite this paper: Zhang, W.Y., Tang, Z.Q., Zhu, G.Y., Dai, B., Song, Z.R. and Li, J. (2019) A Strategy for Loading Oblique Photogrammetry Models and Multilayer Basemap Data. *Journal of Computer and Communications*, 7, 1-8. <https://doi.org/10.4236/jcc.2019.77001>

Received: May 8, 2019

Accepted: July 7, 2019

Published: July 10, 2019

Abstract

With the development of drone technology and oblique photogrammetry technology, the acquisition of oblique photogrammetry models and basemap becomes more and more convenient and quickly. The increase in the number of basemap leads to excessively redundant basemap tiles requests in 3D GIS when loading oblique photogrammetry models, which slows down the system. Aiming at improving the speed of running system, this paper proposes a dynamic strategy for loading basemap tiles. Different from existing 3D GIS which loading oblique photogrammetry models and basemap tiles independently, this strategy dynamically loads basemap tiles depending on different height of view and the range of loaded oblique photogrammetry models. We achieve dynamic loading of basemap tiles by predetermining whether the basemap tiles will be covered by the oblique photogrammetry models. The experimental results show that this strategy can greatly reduce the number of redundant requests from the client to the server while ensuring the user's visual requirements for the oblique photogrammetric model.

Keywords

Dynamic Loading of Basemap Tiles, Oblique Photogrammetry Model, Data Visualization, 3D Tile, Cesium

1. Introduction

Considering the convenience of cross-platform and non-installable browser [1]

[2], the application of 3D GIS gradually changes from PC-end to browser. Cesium [3] is a WebGL-based [4] 3D GIS browser-side open-source framework written by JavaScript, which can be applied without any plug-ins. Following the breakthroughs in drone technology and oblique photogrammetry technology, massive data has been injected into 3D GIS.

Recently, there is no personal computer has the ability to satisfy the requirements of loading the basemap tiles and oblique photogrammetry models at a time. Besides, 3D GIS not only need to load oblique photogrammetry models to display the realistic effects, but also need to load the basemap tiles with different functions. However, as the number of the basemap tiles increase, the number of requests sent by the client to server increases, resulting in excessive load on server. Eventually, the speed of the 3D GIS will drop off.

This paper proposes a dynamic strategy for loading basemap tiles to solve the above problem. The main idea of the strategy is that the basemap tiles loading is determined by the height of view and the range of the oblique photogrammetry models.

2. Data Introduction

2.1. Data Acquisition

We have acquired the raw images by drone equipped with sensors in five directions: vertical, front, rear, left and right [5]. Approximately geographical coordinates in each image were recorded during aerial photography. Meanwhile, the ground surveyors provide ground control points with more accurate geographical coordinates. Then, these raw images with geographical coordinates can be converted to oblique photo-grammetry model by Context Capture Master. Finally, in order to obtain the final data in b3dm format, a series of format conversions have been applied.

2.2. Data Storage

As for the storage of oblique photogrammetry models, this paper adopts HLOD data structure [6] [7]. As **Figure 1** shown, from top to bottom, data resolution becomes higher and higher. Regular grid partitioning will be applied to single file which is extremely large. The files in different levels are named by the method shown in **Figure 2**. **Figure 2** showed that data in different levels has different filename length. And the data index of the same level is different. Not only the hierarchical structure but also the geographical extent is recorded by the index file. The extent bounding box is recorded in the order of West, South, East, North, Lower, and Upper (**Figure 3**). Because of the lightweight of JSON data format, the storage format of the index file is JSON.

Refer to basemap tiles, Cesium builds a quadtree index in the same way as Bing map and Google Earth based on WMS service [8]. The index of each basemap tiles corresponds to a certain tile extent under the specified hierarchy (**Figure 4**).

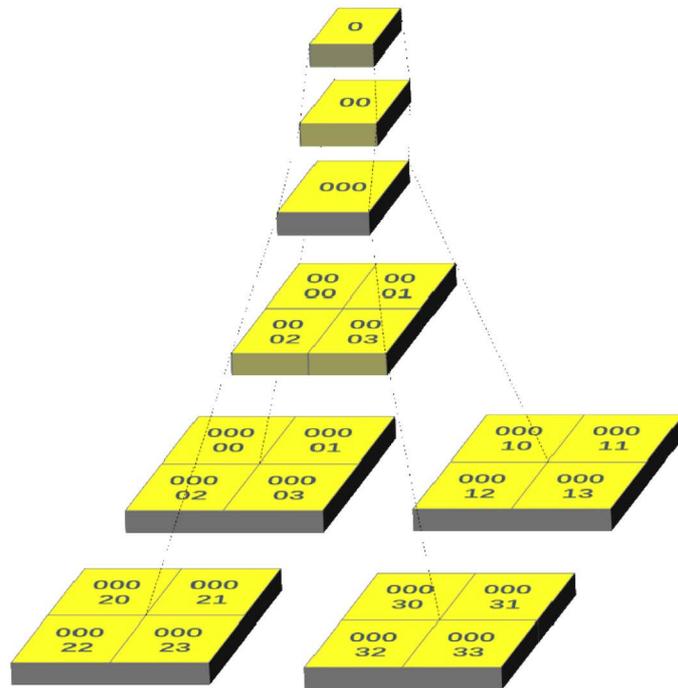


Figure 1. HLOD data structure.

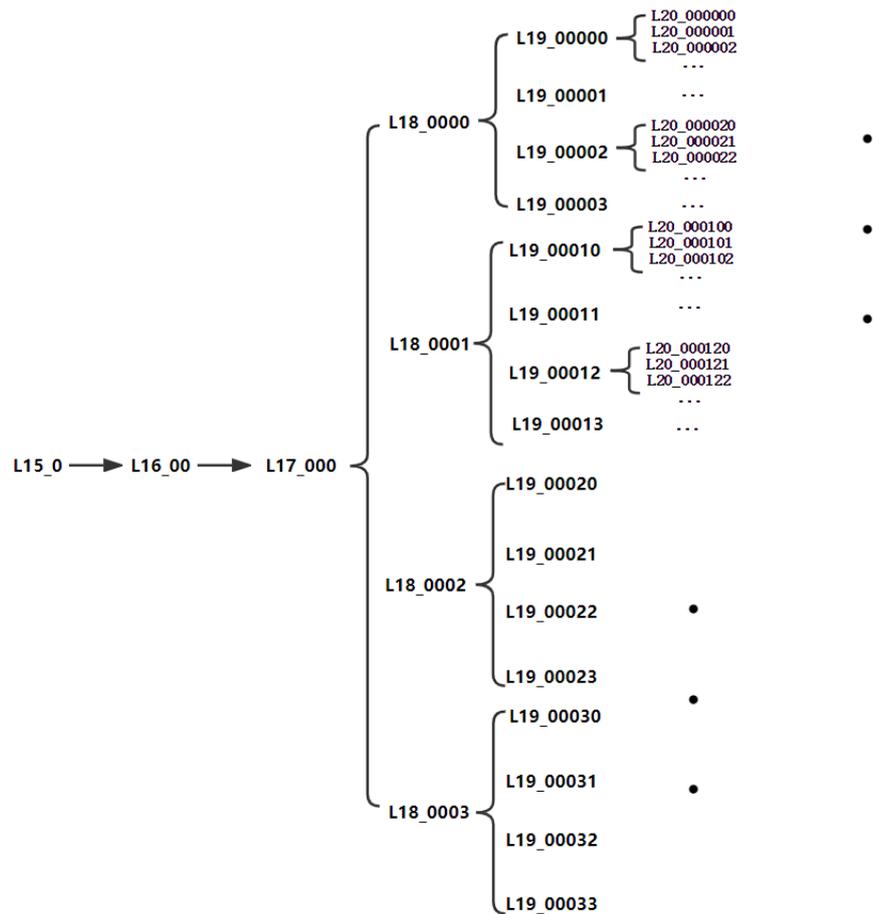


Figure 2. Oblique photogrammetry models file naming rules.

```

"region": [
  1.9719868746636957,
  0.4924953505707456,
  1.9720942154090884,
  0.492574513565812,
  -119.6393737714391,
  161.2211492575291
]

```

Figure 3. The tile range contained in the json index file (west, south, east, north, bottom, top).

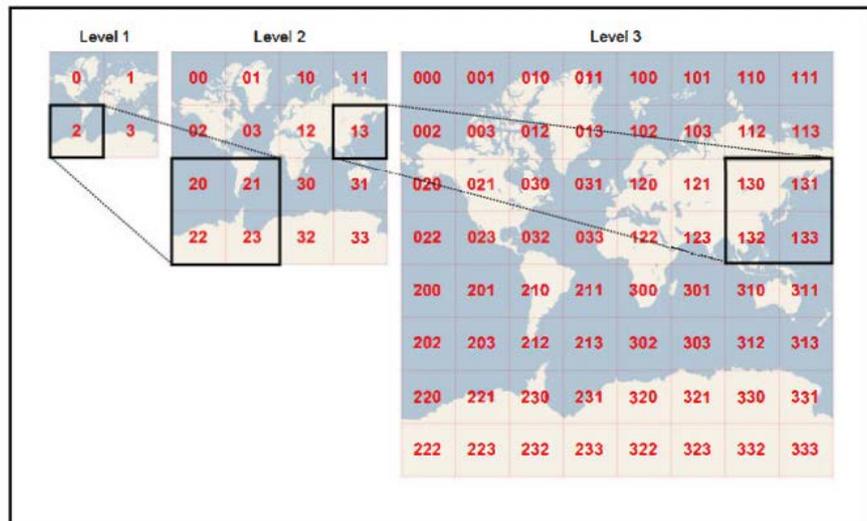


Figure 4. Bing Maps Tile Partition (Source: Microsoft).

3. Strategy Data Loading in Cesium

Cesium adopts the OOC (Out of Core) algorithm [9] for loading massive data. Due to the limited computer memory, the global data loading and processing at once are impossible. Therefore, only the data within the view frustum will be loaded and processed [10] [11].

The loading strategy of basemap tiles and oblique photogrammetry models in Cesium is: the loading of the basemap tiles and the oblique photogrammetry models are decided by the height of view and the scene range. Once the height of view is high and range of scene is large, Cesium loads low resolution basemap tiles only. On the contrary, once the height of view is low and range of scene is small, Cesium loads relatively high resolution basemap tiles and oblique photogrammetry models. Another case is that Cesium will load higher resolution basemap tiles and lower resolution oblique photogrammetry models when the height of view and range of scene are moderate.

However, the whole scene will be covered by the oblique photogrammetry models when the height of view is low and the scene range is small. At the moment higher resolution basemap tiles in covered area don't need to be loaded.

4. Basemap Tiles Dynamic Loading Strategy

When the height of view is low, both the oblique photogrammetry models and basemap tiles require to be requested, loaded and rendered. Nevertheless, if the number of basemap layers is too large, the requests sent by the client increase, the system runs slower. Therefore, the excessive load of server can be relieved by reducing the requests of the basemap tiles from client. When the requests from client decrease, performance of 3D GIS will improve greatly.

4.1. Coverage Area of Oblique Photogrammetry Models

Oblique photogrammetry models and basemap tiles exist in the form of tiles. The range of oblique photogrammetry models is determined by the way of data division. While the range of basemap tiles is determined by specific data partitioning standard. The data division criterion between the oblique photogrammetry models and basemap tiles are different.

In order to speed up scene loading, the covered range of oblique photogrammetry models should be calculated in advance. The geographical range of each oblique photogrammetry model is recorded in the index file. Therefore, covered range in certain area can be calculated by the union of all oblique photogrammetry tiles (Some areas are shown in **Figure 5**).

4.2. Restriction Requests to Basemap Tiles

The strategy for dynamically loading basemap tiles means that the loading of basemap tiles is determined by whether the oblique photogrammetry models are loaded. When the height of view does not satisfy the condition of loading oblique photogrammetry models, only the basemap tiles are loaded; when there are oblique photogrammetry models loading, it is necessary to have some judgements on whether the current basemap tile to be loaded will be covered by the loaded oblique photogrammetry models. The judgement procedures are as follows (**Figure 6**):

- 1) Acquiring the coverage range of each oblique photogrammetry model in current scene;
- 2) Calculating the union of all the ranges in the corresponding area according to the range of each oblique photogrammetry model;
- 3) Acquiring four vertices of the basemap tile that needs to be loaded;
- 4) Calculating whether the four vertices of the basemap tile are include by the coverage of the oblique photogrammetry models;
- 5) Restricting the request of current basemap tile if all four vertices are within

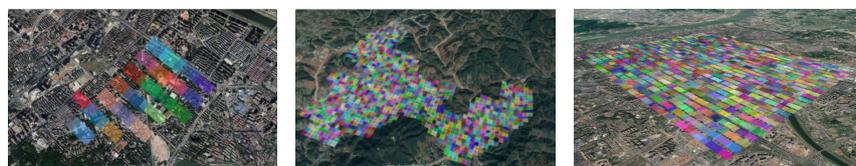


Figure 5. Covered range of B3dm tile.

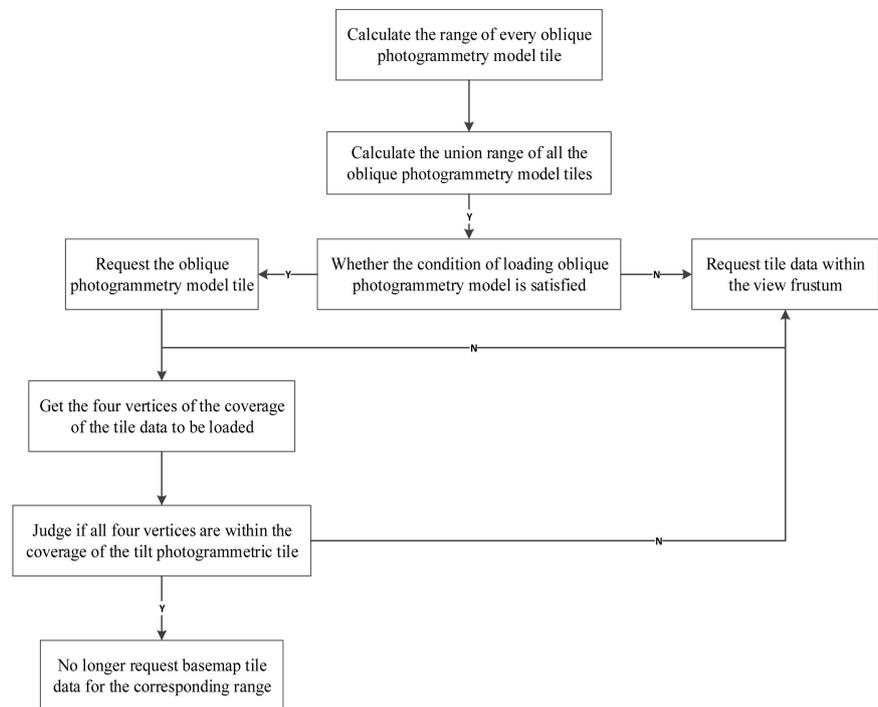


Figure 6. Flowchart loading b3dm oblique photogrammetry models and basemap tiles.

the union range of the oblique photogrammetry models, because the basemap tile will be covered under the oblique photogrammetry models;

6) Requesting the basemap tile if any of the four vertices of the base tile is outside the coverage of oblique photogrammetry models.

5. Experiment

In order to demonstrate the performance of our strategy. In this section, the experiment is carried out based on 1 basemap, 5 layers of basemap and 10 layers of basemap. The number of requests for b3dm oblique photogrammetry models and the basemap tiles was recorded during the experiment.

The experimental data includes the National University of Defense Technology campus, Fenghuang Ancient City and parts of Changsha City. Three groups of experimental loading scenes shown in the **Figure 7**.

In **Figure 8**, we compare the dynamical basemap tiles loading strategy proposed in this paper with Cesium's existing basemap tiles loading strategy. The comparison diagram of the number of basemap tiles requests, b3dm requests and total requests in National University of Defense Technology Campus is shown in **Figure 8(a)**. The same experiments were also carried out for the Fenghuang Ancient city and some areas of Changsha, the experimental results are shown in **Figure 8(b)** and **Figure 8(c)** respectively.

The experimental environment is Intel® Core(TM) i7-7700K@4.20 GHz 64GB RAM, NVIDIA GeForce 1080 MB; 69.0.3497.100 version of Google Chrome, win7 64-bit Ultimate OS.



Figure 7. Scenes of b3dm data loading.

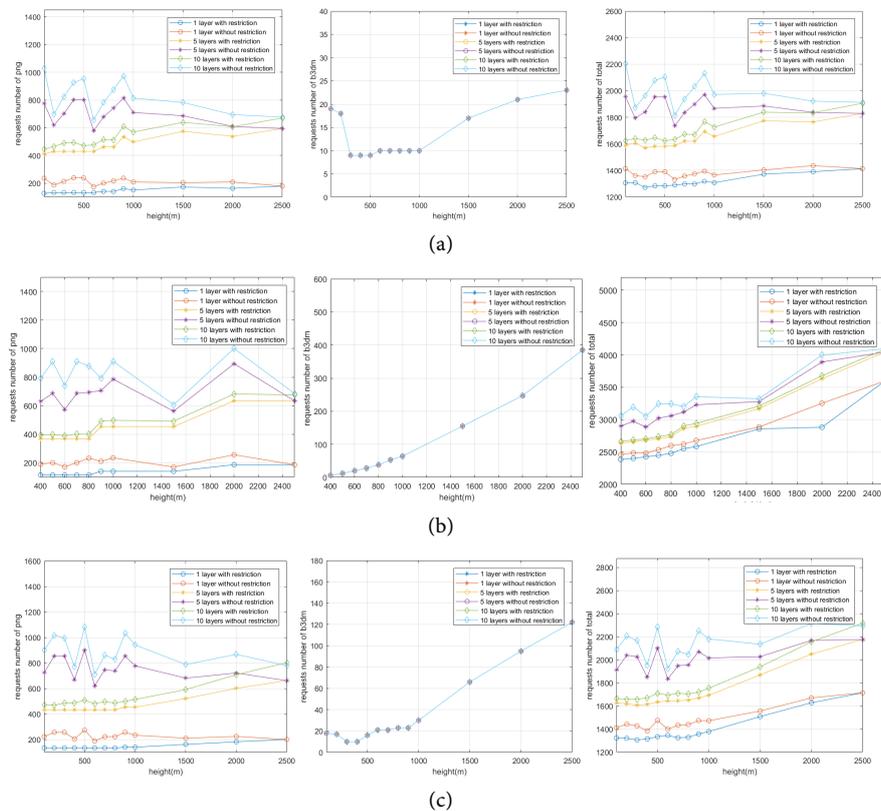


Figure 8. Comparison of request number. (a) National University of Defense Technology Campus; (b) FengHuang Ancient City; (c) Some areas of Changsha.

6. Conclusions

In 3D GIS, loading multilayer basemap and oblique photogrammetry models simultaneously will lead to excessive number of requests from the client, which slows down the speed of 3D GIS. This paper proposes a strategy for basemap tiles dynamical loading. The strategy loads oblique photogrammetry models preferentially, and limits the requests of basemap tiles. In other word, the loading of basemap tiles is affected by the oblique photogrammetry models during the browsing from the users. It is not necessary to load the basemap tiles if they are covered by oblique photogrammetry models. The experiment shows reduction in the number of requests for basemap tiles can greatly improve the performance of 3D GIS.

The requesting, loading and rendering of b3dm oblique photogrammetry models will not be affected by restricting the requests of basemap tiles. Consequently, the strategy proposed by this paper can improve the running efficiency

of the entire system without affecting the overall use of the system

Acknowledgements

Thanks to the 2nd Surveying and Mapping Institute of Hunan Province and Changsha Municipal Bureau of Land and Resources for providing us with oblique photogrammetry model for experiments.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Niu, N., Li, C. and Guo, L. (2013) The Application and Research of Geoprocessing Service in WebGIS Spatial Analysis. *21st International Conference on Geoinformatics*, 1-4. <https://doi.org/10.1109/Geoinformatics.2013.6626054>
- [2] Miao, R., Song, J. and Zhu, Y. (2017) 3D Geographic Scenes Visualization Based on WebGL. *IEEE International Conference on Agro-Geoinformatics*. <https://doi.org/10.1109/Agro-Geoinformatics.2017.8046999>
- [3] Cesium (2016) An Open-Source JavaScript Library for World-Class 3D Globes. <https://cesiumjs.org>
- [4] Feng, L., Wang, C., Li, C., *et al.* (2012) A Research for 3D WebGIS Based on WebGL. *IEEE International Conference on Computer Science & Network Technology*.
- [5] Dong, S. (2017) Application of Oblique Photogrammetry in 3D Modeling. *City Construction Theory Research* (Electronic Edition), China.
- [6] Zhu, Q., Chen, X., Ding, Y., *et al.* (2017) Organization and Scheduling Method of 3D Urban Scene Data Driven by Visual Perception. *Journal of Southwest Jiaotong University*, Chengdu, 869-876.
- [7] Liu, H., Liu, J., Wang, Y., *et al.* (2010) Data Organization and Scheduling of Massive Buildings Combined with Grid Division and Quadtree. *Bulletin of Surveying and Mapping*, China.
- [8] Guo, Y. (2015) A Comprehensive Performance Evaluation of 3D Local Feature Descriptors. *International Journal of Computer Vision*, **116**, 66-89. <https://doi.org/10.1007/s11263-015-0824-y>
- [9] (2006) OpenGIS Web Map Service Version 1.3.0 No. OGC 06-042. Open Geospatial Consortium.
- [10] Cozzi, P.J. (2008) Visibility Driven Out-of-Core HLOD Rendering. Diss. MS Thesis, University of Pennsylvania.
- [11] Song, Z. and Li, J. (2018) A Dynamic Tiles Loading and Scheduling Strategy for Massive Oblique Photogrammetry Models. *IEEE International Conference on Image, Vision and Computing (ICIVC)*. <https://doi.org/10.1109/ICIVC.2018.8492731>