FAST OCCLUSION AND SHADOW DETECTION FOR HIGH RESOLUTION REMOTE SENSING IMAGE COMBINED WITH LIDAR POINT CLOUD

Xiangyun Hu [a,] *, Xiaokai Li [a]

[a] School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road,
Wuhan, CHINA, 430079

Commission VII, WG VII/6

KEY WORDS: Fast, Occlusion, Shadow, High Resolution, Remote Sensing, LiDAR

ABSTRACT:

The orthophoto is an important component of GIS database and has been applied in many fields. But occlusion and shadow causes the loss of feature information which has a great effect on the quality of images. One of the critical steps in true orthophoto generation is the detection of occlusion and shadow. Nowadays LiDAR can obtain the digital surface model (DSM) directly. Combined with this technology, image occlusion and shadow can be detected automatically. In this paper, the Z-Buffer is applied for occlusion detection. The shadow detection can be regarded as a same problem with occlusion detection considering the angle between the sun and the camera. However, the Z-Buffer algorithm is computationally expensive. And the volume of scanned data and remote sensing images is very large. Efficient algorithm is another challenge. Modern graphics processing unit (GPU) is much more powerful than central processing unit (CPU). We introduce this technology to speed up the Z-Buffer algorithm and get 7 times increase in speed compared with CPU. The results of experiments demonstrate that Z-Buffer algorithm plays well in occlusion and shadow detection combined with high density of point cloud and GPU can speed up the computation significantly.

1. INTRODUCTION

The orthophoto is an important component of geographic information system (GIS) database and has been applied in many fields. That it has uniform scale and no relief displacement enables the users to measure distances and areas directly. However, the traditional generation of orthophotos is based on digital elevation models (DEM) which doesn’t take the buildings and any other objects above the terrain into account. Occlusion and shadow effects caused by the abrupt change of surface height are major aspects of information degeneration in orthophotos (Rau *et al.*, 2002). In true orthophotos, the occlusion and shadow should be detected and compensated. Light Detection and Ranging (LiDAR) integrates the Global Navigation Satellite System (GNSS) and Inertial Navigation System (INS) with laser scanning and ranging technologies. It offers a directly method to measure the three-dimensional coordinates of points on ground objects and makes the creation of digital surface models (DSM) very efficient (Disa *et al.*, 2011). Combined with this technology, image occlusion and shadow can be detected automatically.

The Z-Buffer algorithm is one of the most popular methods of occlusion detection (Liang-Chen *et al.*, 2007, Bang *et al.*, 2007). It calculates the distances between the projection centre and object points. The closest object point is visible while others are occluded in one line of sight (Amhar *et al.*, 1998). The shadow detection can be regarded as a same problem with occlusion detection considering the angle between the sun and the camera (Rau *et al.*, 2002). However, this method is computationally intensive (Kato *et al.*, 2010). On the other hand, the resolution of the remote sensing images and LiDAR point cloud is becoming higher and higher, making the large volume of data in remote sensing greater. It has been one of the main challenges in data processing.

With the rapid development of computer hardware, the processing power is growing. Central Processing Unit (CPU) has entered the era of multi-core and it shows strong parallel computing power. And Graphic Processing Unit (GPU) has evolved into highly parallel, multi-threaded, many-core processors with tremendous computational horsepower and a very high memory bandwidth (NVIDIA, 2011). It is much more powerful than CPU in parallel computing capability. Since the release of Compute Unified Device Architecture (CUDA), it has become increasingly convenient and efficient to use GPUs to speed up applications.

In this paper, the method of occlusion and shadow detection for high resolution remote sensing image combined with LiDAR point cloud is discussed and the GPU is introduced to accelerate the algorithm.

2. OCCLUSION AND SHADOWN DETECTION

Occlusion detection is a problem of visibility analysis (Hablb *et al.*, 2007). And the Z-Buffer algorithm is the most commonly used method. In the algorithm, an image matrix called z-buffer is used to store the distance (Z) between the projection centre and points in the object surface which correspond to the pixels in the image. An index matrix is also needed to denote the visibility of every point. Some points may be projected onto the same pixel. Calculate the distance and compare it with the existing

*huxy@whu.edu.cn; phone 86 27 687-78010; fax 86 27 687-78086

distance in the z-buffer. If the calculated distance is smaller, the previous ground point corresponding to the cell in z-buffer is labelled as invisibility and the distance in the z-buffer is updated with the small distance (Qin *et al.*, 2003). In figure 1, the distances between points in line AC and projection centre O is bigger than those between points in line BD,DC and O, so the points in line AC is invisible. Triangulated irregular network (TIN) needs to be created first, considering that the ground is represented by the discrete LiDAR points. Then project each triangle onto image plane. Shown in Figure 2, rasterize each triangle and use the z-value of point A, B, C to obtain the z-value of each pixel in the triangle by interpolating . At last, update the z-buffer.

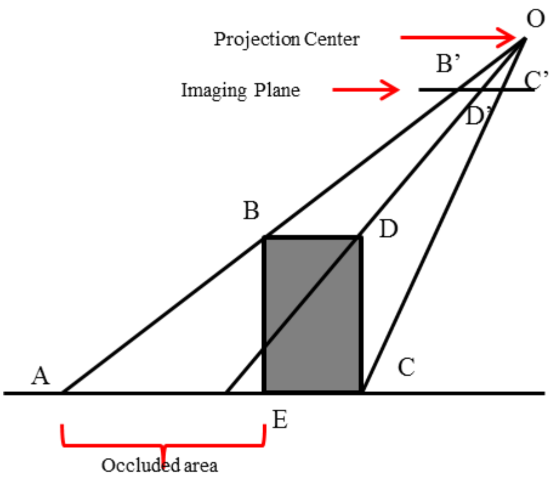


Figure 1. Diagram of Z-Buffer algorithm

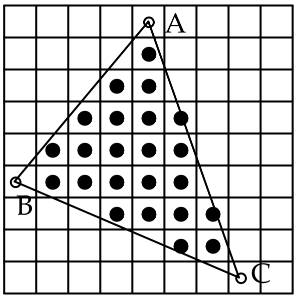


Figure 2. Rasterization of triangle

So, the procedure of the occlusion detection based on LiDAR point is below:

1) Prepare data.
2) Create TIN.
3) Project each LiDAR point to the image plane and calculate the z-value for each point.
4) Rasterize each triangle in image plane and update z-buffer.
5) Compare z-value of each point with the corresponding value in z-buffer and get the index matrix.

Shadow is due to the ground objects blocking the light. From this perspective, it is the same with the problem of occlusion. So it is available to use the same method to detect shadow. And now the position of the projection centre is depend on the zenith angle and azimuth angle of the sun which can be calculated based on the imaging time or just measure the direction of the shadow in the image manually. The projective mode becomes parallel projection (Zhou *et al.*, 2005, Rau *et al.*, 2002).

3. GPU ACCELERATION

The detection of occlusion and shadow is time consuming due to the expensive computation of the visibility analysis and the large volume of data. And in Z-Buffer algorithm, the obtaining or updating of z-buffer is the slowest. GPU is much more powerful than CPU in parallel computing. Two ways of acceleration using GPU are introduced in this part. One is using OpenGL. The other is CUDA.

OpenGL is the most widely used, supported and best documented 2D/3D graphics API in industry. In OpenGL, there is a buffer called *Depth Buffer* which is used to removal the hidden surface. That means the 4) step in the procedure of detection above can be implemented like this: open the *Depth Test* in OpenGL and render the TIN in *GL_FILL* mode to off-screen buffer, then read the data in the *Depth Buffer* which is just what we need (z-value). The rendering is optimized in GPU hardware, so it is very fast (Segal *et al.*, 2012).

Since the release of CUDA, it has become increasingly convenient and efficient to use GPUs to speed up applications. In the step 4), every triangle is rasterized independently. So it is highly suitable for parallel computing. Every triangle is processed by an independent thread. Shown in Figure 3, every triangle is divided into two parts by line 1 which passes through vertex B. Then determine the start and end points (point E and F) of each row and the pixels between them. The coordinates and z-value of point F are interpolated from point A and D. This is the same with point E. Then calculate the z-value between E and F by interpolation from E and F. After that, calculate the next row in the same way. Because points in different triangles may be projected onto the same pixel, *atomic operation* is needed when updating the z-buffer.

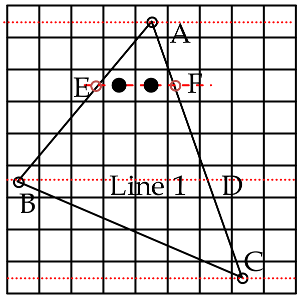
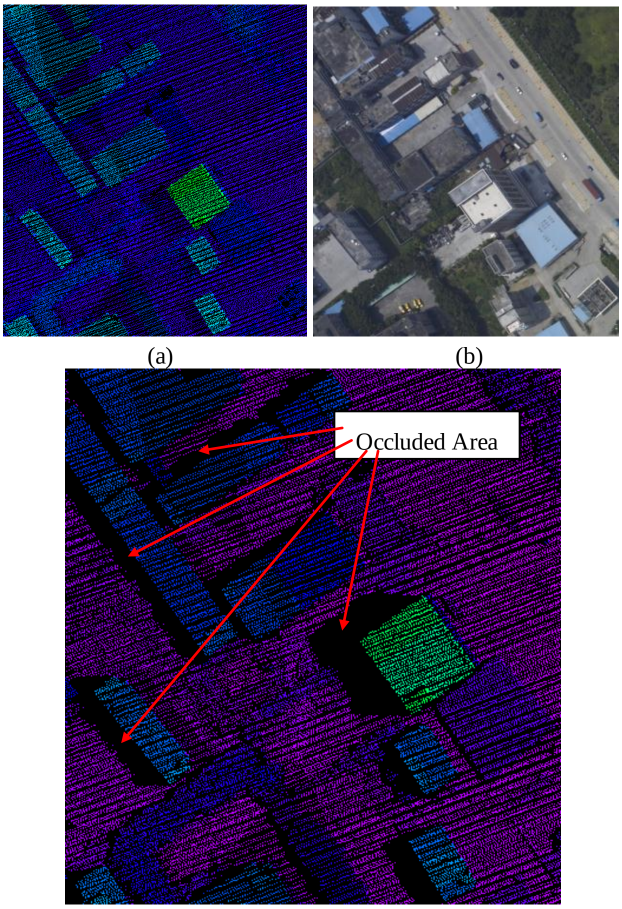


Figure 3. Rasterization of triangle on GPU

4. EXPERIMENTS

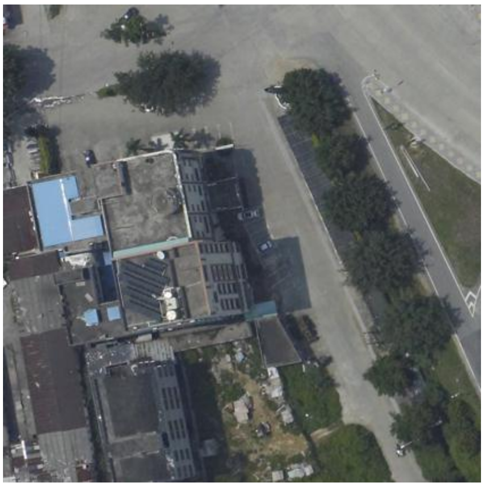
The data in the experiments comes from the city of Huiyan in Guangdong Province in China that contains approximately 1.2 million points with a point density of 1.4 points/m^2 and a 7228*5228 aerial image with the resolution of 0.13m/pixel. Shown in Figure 4, the first image is the original LiDAR point cloud and the second is the original image. The third image shows the hidden area in black. Figure 5 shows the original image and the result of shadow detection. All experiments are done on a PC with Intel(R) Core(TM) i7-920 @ 2.67 Hz CPU with 4.0 GB memory, a NVIDIA GeForce GTX285 GPU with 1.0 GB memory, and Windows 7 Ultimate - 64bit system. Using Z-Buffer to detect the occlusion takes about 24.2 seconds in CPU. The implementation in OpenGL needs about 4.3s. Accelerated by CUDA in GPU, the time is less than 3.4s and we get over 7 times' speedup. It is noteworthy that the creation of TIN takes about 2.1s. So the limit of the running time is not the calculation of the z-buffer any more.



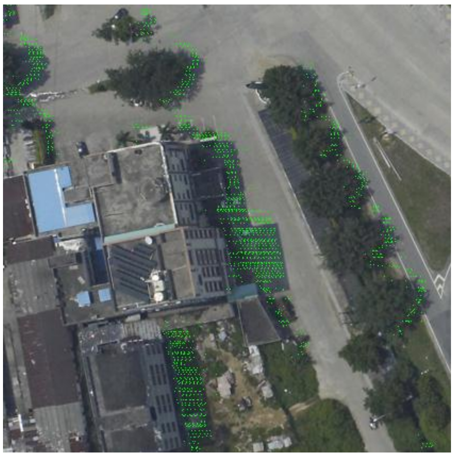
(a) (b)

(c)

Figure 4. Occlusion Detection



(a)



(b)

Figure 5. Shadow Detection

5. CONCLUSION

In this paper, the Z-Buffer algorithm is used to detect the occlusion and the shadow whose detection and compensation is a critical step in the generation of true orthophoto. GPU is introduced to accelerate the process. Experimental results indicate that the fast detection of occlusion and shadow combined with LiDAR point cloud is effective and efficient.

There are still some key point of the algorithm needs to be studied further. Firstly, the precision of the detection result need to be improved. In this paper, only coarse edges are obtained. Secondly, the compensation of the occlusion and shadow using multi-view images is another problem. Thirdly, only NVIDIA's GPUs support CUDA. Open Computing Language (OpenCL) is a better alternative. These three aspects are the direction of our future research.

6. ACKNOWLEDGEMENTS

Thank Guangzhou Jiantong Surveying and Mapping Technology Development Ltd. for providing the experimental data.

REFERENCE

Amhar F., Jansa J. and Ries C., 1998. The generation of true orthophotos using a 3D building model in conjunction with a conventional DTM. *International Archives of Photogrammetry and Remote Sensing*, 32(4): 16-22.

Bang K.I. and Habib A.F., 2007. Comparative analysis of alternative methodologies for true ortho-photo generation from high resolution satellite imagery. *ASPRS 2007 Annual Conference*.

Chen L.C., Teo T.A., Wen J.Y., Rau J.Y., 2007. Occlusion-Compensated True Orthorectification for High-Resolution Satellite Images. *The Photogrammetric Record*, 22(117), pp. 39-52.

Disa N.M., Maarof I., Latif Z.A. and Samad A.M., 2011. LiDAR: A Review On Generating Digital True Orthophoto. *IEEE 7th International Colloquium on Signal Processing and its Applications*, pp. 336-340.

Hablb A.F., Klm E.M. and Klm C.J., 2007. New methodologies for true orthophoto generation. *Photogrammetric Engineering & Remote Sensing*, 73(1), pp.025-036.

Kato A., Moskal L.M., Schiess P., Calhoun D. and Swanson M.E., 2010. True orthophoto creation through fusion of LiDAR derived digital surface model and aerial photos. *International Archives of Photogrammetry and Remote Sensing*, 38(7A): 88-93.

NVIDIA CUDA^{TM}, 2011. NVIDIA CUDA C Programming Guide. NVIDIA, pp.1-1.

Segal M. and Akeley K., 2012. The OpenGL Graphics System: A Specification.

Qin Z., Li W., Li M., Chen Z. and Zhou G., 2003. A methodology for true orhtorectification of large-scale urban aerial images and automatic detection of building occlusions using digital surface model. *IEEE International Geoscience and Remote Sensing Symposium*, 2, pp.729-731.

Rau J.Y., Chen N.Y., Chen L.C., 2002. True Orthophoto Generation of Built-Up Areas Using Multi-View Images. *Photogrammetric Engineering & Remote Sensing*, 68(6), pp.581-588.

Zhou G., Chen W., Kelmelis J.A. and Zhang D., 2005. A Comprehensive Study on Urban True Orthorectification. *IEEE Transaction on Geoscience and Remote Sensing*, 43(9), pp. 2138-2147.