UDC 004.92

E. Malafeev,
V. Sytnikov, ScD.

USING SIMPLIFIED GEOMETRY FOR CONCRETE SITES OF 3D-LANDSCAPE DETALIZATION

Abstract. *Description of approach for selecting the level of detailing for sites of 3D-landscape by analyzing the scene and selecting the desired level of detailing for the tile, allowing to keep the information content and increase the performance of the CPU and GPU. The principles of conservation the resolution of texture for tile that prevents blur or sharpen of the image are given.*

Keywords: *tile, quadtree, levels of detail, texturing, server OSM, analyzer of detail, scale of the rendering, perspective projection, camera, pixel*

Е. А. Малафеев,
В. С. Ситников, д-р техн. наук

ПРИМЕНЕНИЕ УПРОЩЕННОЙ ГЕОМЕТРИИ ДЛЯ ДЕТАЛИЗАЦИИ КОНКРЕТНЫХ УЧАСТКОВ 3D-ЛАНДШАФТА

Аннотация. *Предложен подход к выбору уровня детализации для участков 3D-ландшафта за счет анализа сцены и выбора уровней детализации тайлов, что позволяет сохранить информативность и повысить производительность процессора и видеокарты. Описаны принципы сохранения разрешения текстуры тайла, что предотвращает размытие или избыточную резкость изображения.*

Ключевые слова: *тайл, дерево тайлов, уровни детализации, текстурирование, сервер OSM, анализатор детализации, масштаб рендринга, перспективная проекция, камера, пиксель*

Є. А. Малафеєв,
В. С. Ситніков, д-р техн. наук

ЗАСТОСУВАННЯ СПРОЩЕНОЇ ГЕОМЕТРІЇ ДЛЯ ДЕТАЛІЗАЦІЇ КОНКРЕТНИХ ДІЛЯНОК 3D-ЛАНДШАФТУ

Анотація.*Запропоновано підхід до вибору рівня деталізації для ділянок 3D-ландшафту, за рахунок аналізу сцени і вибору рівнів деталізації тайлів, що дозволяє зберегти інформативність та підвищити продуктивність процесора і відеокарти. Описані принципи збереження оригінального розміру текстури тайла, що запобігає розмиттю або зайвої різкості зображення.*

Ключові слова: *тайл, дерево тайлів, рівні деталізації, текстурування, сервер OSM, аналізатор деталізації, масштаб рендрінга, перспективна проекція, камера, піксель*

Introduction

Traditional approaches for generation of 3D-maps lead to a significant increase in the load on the graphics processor, which leads to development of methods to improve the performance of the GPU or search for alternative approaches to reduce the load by reducing the calculations. It is necessary to apply a simplified geometry to reduce the load on the GPU and CPU when drawing objects of 3D-map [1].

Usually landscape has several levels of detail and is divided into sections called tiles. It should be noted that each subsequent tile is partitioned into four parts (also tiles), etc. Then the initial tile of landscape can be detailed by four subsequent “descendants”, Fig. 1 [1].

The use of simplified geometry is possible after analyzing the scene and selecting the level of detail (LOD) for current tiles. The tasks of analyzer of detail include analysis of the location of the camera, choice of the LOD for each plot of terrain. Decision making should be of high quality and fast to prevent blur of the texture of tile. Blurring occurs when the resolution of the texture on the screen does not correspond to resolution of the original. The elimination of this blur of texture should be attributed to the problem of analyzer of detail for the current scene.

Therefore, the aim is to improve analyzer of detail of the scene.

To choose the appropriate LOD is necessary that analyzer of detail takes into account that plots of the landscape (tiles) depending on the location of the camera (observer) should have a different LOD: closer plots should use tiles with high detail, and remote should use tiles with low detail Fig.1.

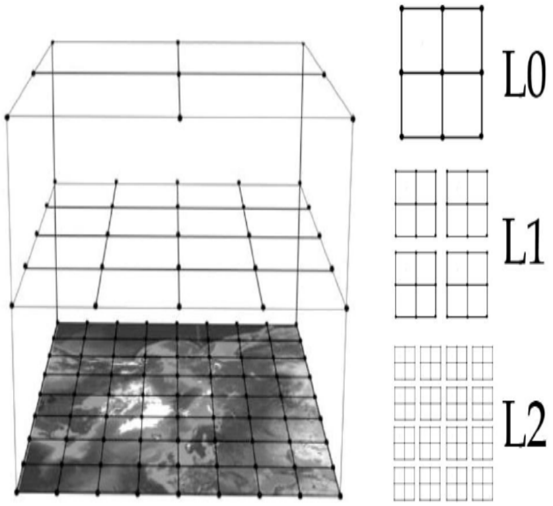


Fig. 1. Representation of the quadtree for tiles: L0 – initial LOD (low, with one tile); L1 – the first LOD (with detail above the previous level, with four tiles); L2 – the second LOD (with detail above the previous level, with sixteen tiles)

Partition of the landscape contributes to the creation of the tree of tiles (quadtree) and the formation of geometry of tiles with different levels of detail for the entire surface with other objects of the world usually in the database Fig.2.

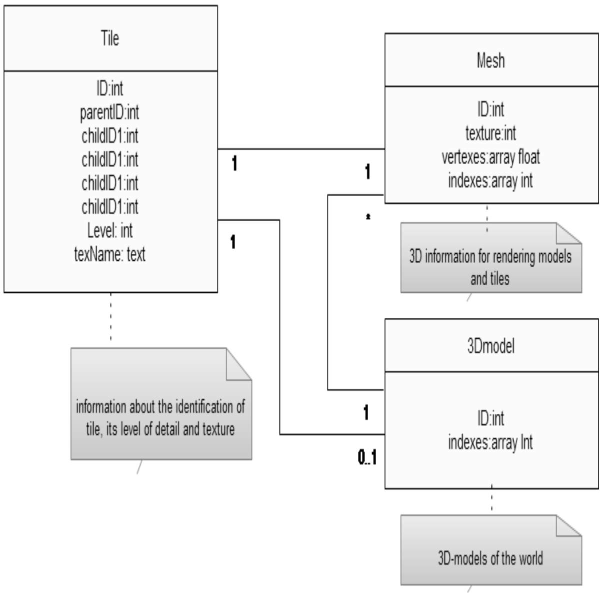


Fig. 2. Relational model of database

In this case the remote parts of the landscape occupy fewer pixels on the screen than the closer areas to the observer; it helps reduce detail without degrading the image quality and improve the performance of GPU and CPU. Algorithm for selecting the desired detail for tiles analyzes the number of points occupied by tile on the screen, Fig. 3.

To determine the number of pixels H, which tile takes on the screen necessary to count the unit of scale of rendering (drawing)

$$d = WHeight / (2 / 0 \cdot tg(fogy / 2.0)),$$

where WHeight – the number of pixels on monitor in the vertical.

From the similarity of triangles, Fig. 3, the sides are in the ratio

$$\frac{d}{dist} = \frac{H}{h},$$

hence the number of pixels H is equal to

$$H = h * d / dist.$$

Knowing the number of pixels on the screen and the resolution of the texture, which is received from the server (e.g., server “Openstreetmap” (OSM) with a resolution of 256x256 pixels) must harmonize the texture resolution of the original with number of pixels H, in order to prevent degrading the image quality on the screen (e.g., blurring or sharpen). To avoid a degrading the image quality on the screen should be set the conditions for transition to another LOD. For example, for tile from server of OSM, which has resolution equal 256x256 pixels, conditions of transitions are satisfied for

min = 256 - 256/2 = 128,
max = 256 +256 / 2 = 384.

Algorithm of analyzer of the detail is shown in Fig. 4 and the result of work Fig. 5.

For example, for texturing of terrain should request information from the server about of textures. To do this, you should know the identifier of tile in OSM, which is formed as *http://[abc].tile.openstreetmap.org/zoom/x/y.png,* where: a, b, c – subdomain of server, zoom – the LOD in a range from 0 to 18; x, y – an identifier of tile in OSM; the obtained image will be in format PNG with resolution of 256 × 256 pixels.

Identification of the tiles on the map OSM performed in accordance with the forms shown in Fig. 6 and 7.

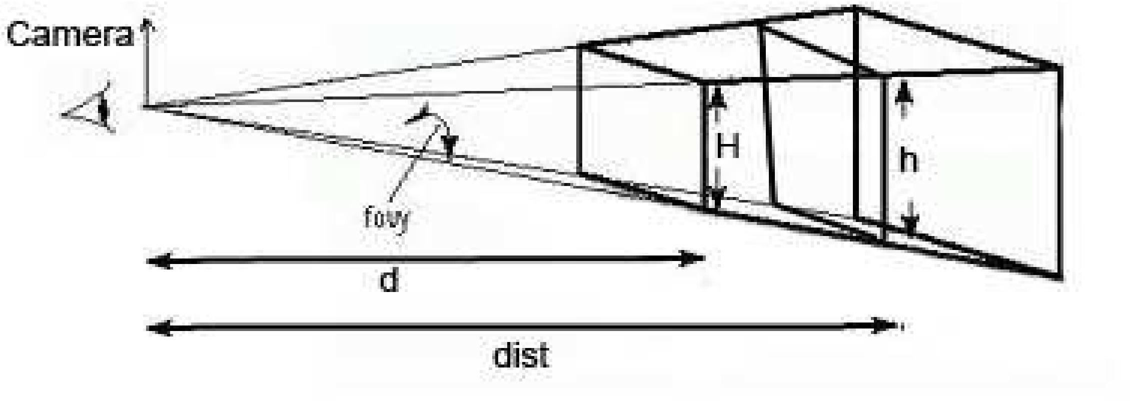


Fig. 3. The tile of landscape in perspective projection:
fovy – field of view of the observer; H – tile size in pixels on the screen; h – tile size in relative units 3D-Api; dist – the distance from the observer (camera) to the tile; d – unit of the scale rendering (drawing) on the screen

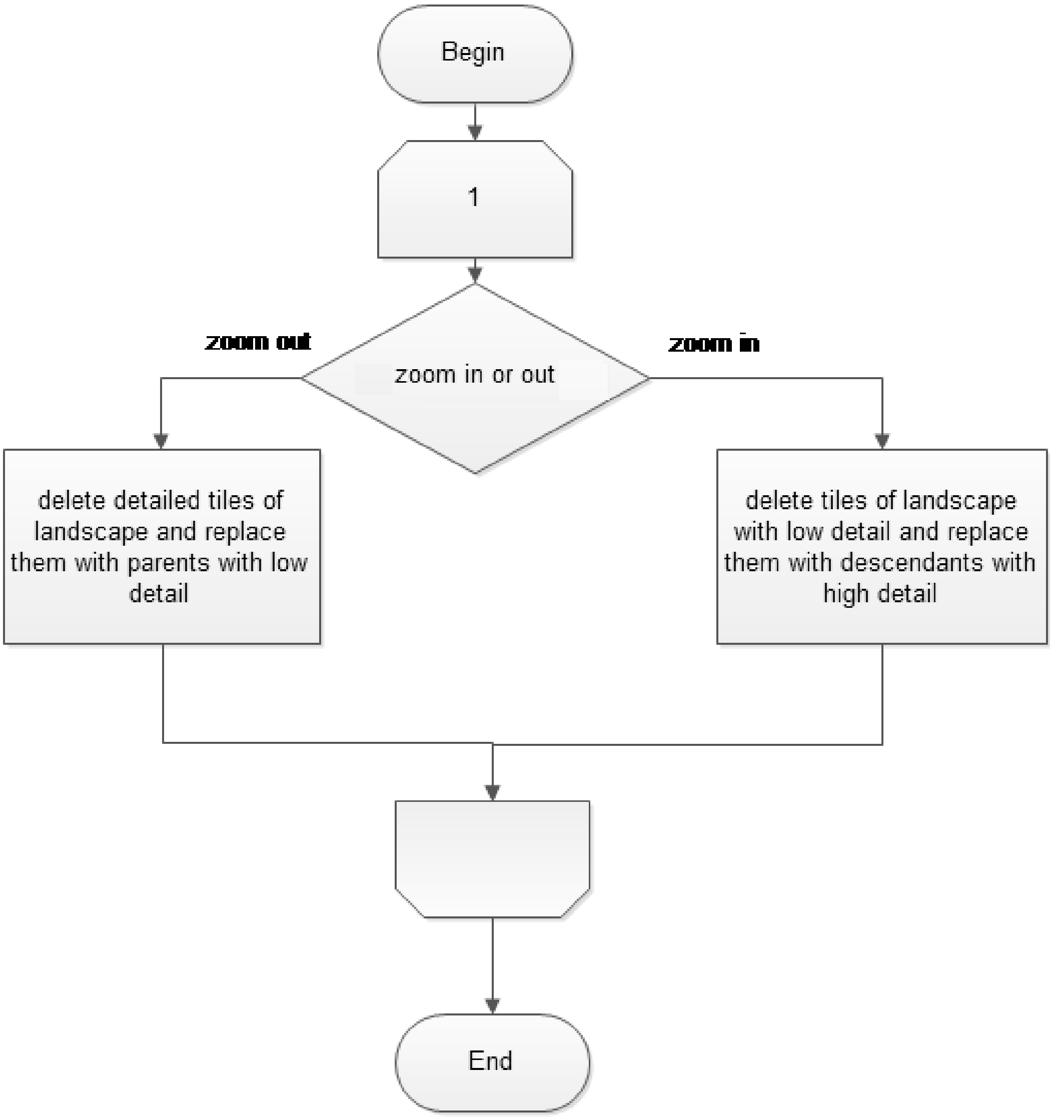
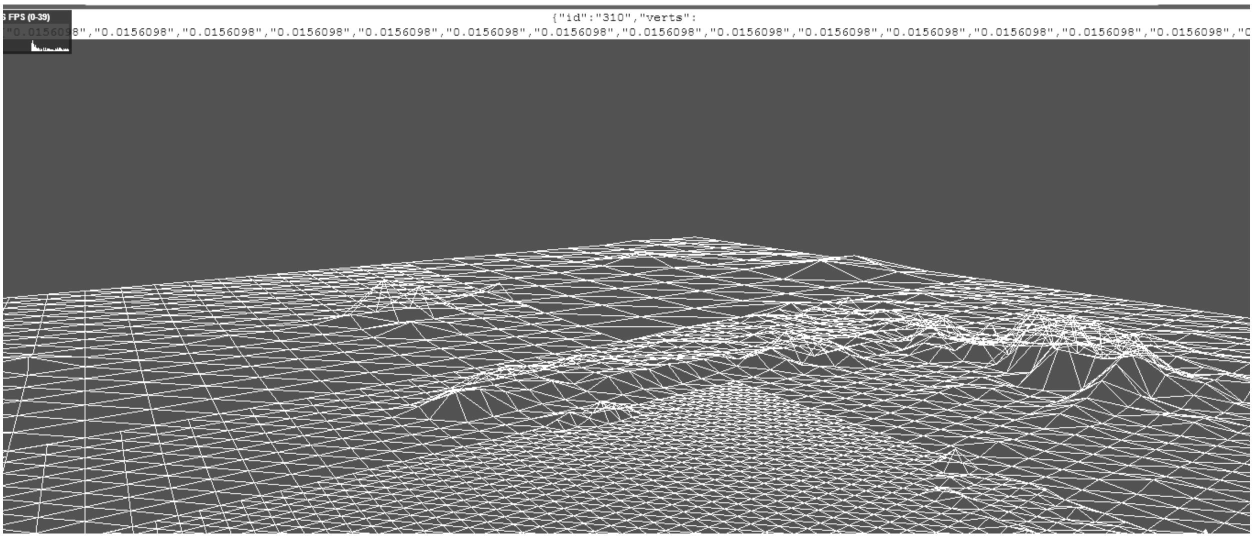
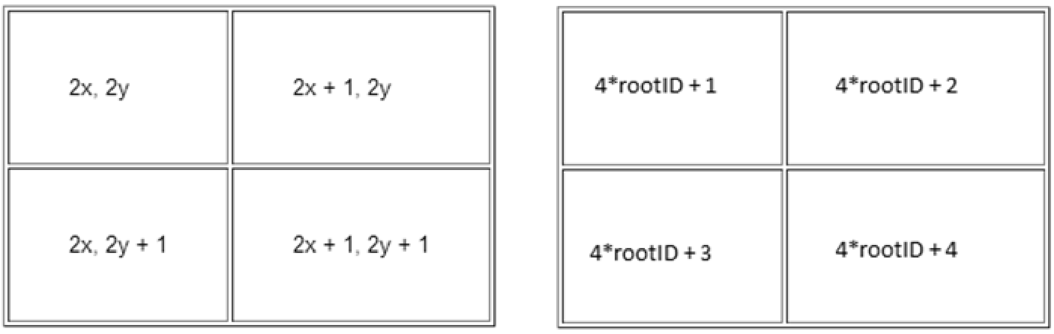

Begin
1
zoom out
zoom in or out
zoom in
delete detailed tiles of landscape and replace them with parents with low detail
delete tiles of landscape with low detail and replace them with descendants with high detail
End

Fig. 4. Algorithm of analyzer of the detail



{"id":"310","verts":
"0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","0.0156098","C

Fig. 5. Visualization of landscape with different LOD without texturing

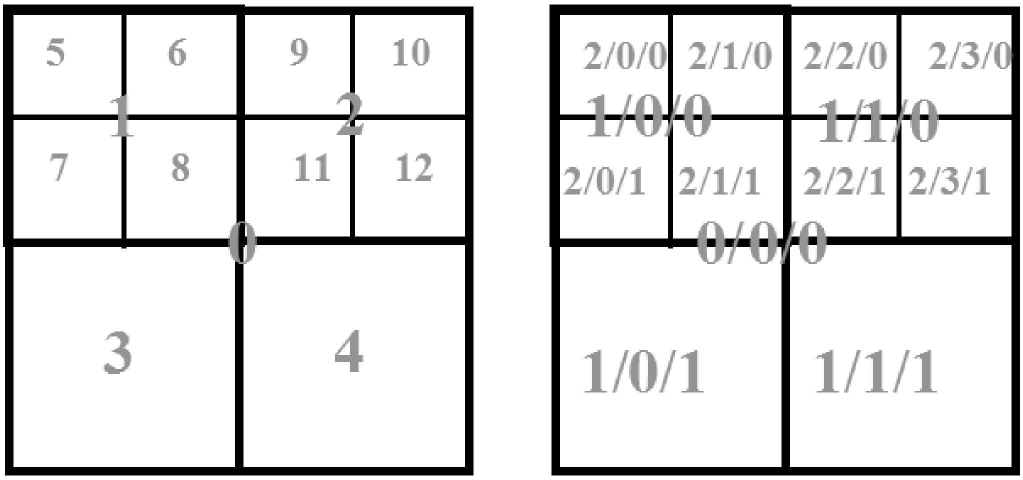


2x, 2y	2x + 1, 2y
2x, 2y + 1	2x + 1, 2y + 1

4*rootID + 1	4*rootID + 2
4*rootID + 3	4*rootID + 4

a b

Fig. 6. Identification of the tiles on the quadtree of the landscape:
a – OSM-formation; b – formation, which is proposed to use



a b

Fig. 7. Example of identification of the tiles on the quadtree:
a – fragments are defined by the formula (Fig. 6 b); b – fragments are defined by zoom / x / y; x, y – according to the formula (Fig. 6 a), zoom – the LOD

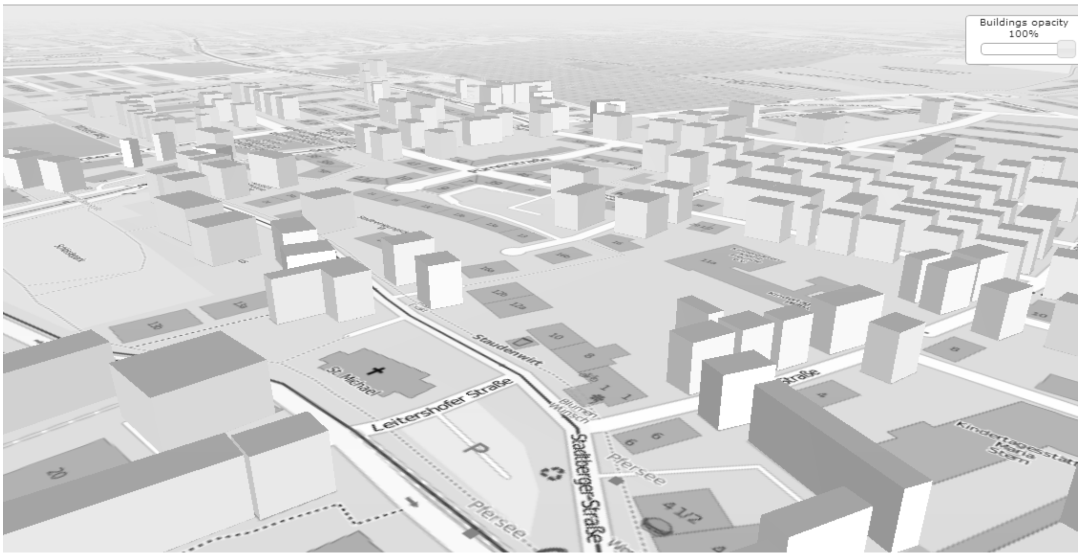


Fig. 8. Texturing of landscape with different levels of detail

Conclusions

Thus, these calculations allow us to improve the operation of the analyzer of detail by matching the number of pixels of the original texture of tile which is coming from the map server that makes extensive use of the simplified geometry in remote areas of the image. In addition, the flexibility of the algorithm allow to minimize the transition to another resolution of texture of the server and appears universality conditions for transition to neighboring levels, which improves performance video card and processor.

References

1. Snook G. Real-Time 3D Terrain Engines Using C++ and DirectX 9, (2003), *Charles River Media,* Hingham, 374 p. (In English).

2. Nitschke B. Professional XNA Programming, (2008), *Wiley Publishing*, Indianapolis, 329 p. (In English).

3. Polack T. Focus on 3D Terrain Programming, (2002), *Course Technology PTR,* Cincinnati, 232 p. (In English).

4. Cozzi P., and Ring K. 3D Engine Design for Virtual Globes, (2011), *A K Peters/CRC Press,* Cincinnati, 520 p. (In English).

5. Zerbst S. 3D Game Engine Programming, (2004), *Course Technology PTR,* Cincinnati, 896 p. (In English).

6. Stephen M., and Hope H. Landscape Modeling: Digital Techniques for Landscape Visualization, (2001), *McGraw-Hill Professional Publishing,* Bernburg, 352p. (In English).

7. Jackson D., and Simpson R. Deity digital earth: Virtual Nations: Data Cities, (2012), *Newtown, N.S.W.: DCity Pty Ltd*, Newtown, 382 p. (In English).

8. Ian D. Visualization in Landscape and Environmental Planning: Technology and Applications, (2005), *Taylor and Francis,* Boston, 320p. (In English).

9. James L., and Mark S. Digital Land: Integrating Technology into the Land Planning Process, (2007), *Wiley Publishing,* Inc-Indianapolis, 240 p. (In English).

10. Cantrell B. Modeling the Environment: Techniques and Tools for the 3D Illustration of Dynamic Landscapes, (2012), *Wiley Publishing, Inc,* Indianapolis, 288 p. (In English).

Received 10.04.2014

Malafeev Evgenii,
M.Sc, Department of Computer Systems of Odessa National Polytechnic University,
Ukraine.
E-mail: jikero@mail.ru

Sytnikov Valery Stepanovich, Doctor of Technical Sciences, professor, Head of the Department of Computer Systems of Odessa National Polytechnic University,
tel: +38-048-705-85-61.
E-mail: sitnvs@mail.ru