Construction of Unique Buildings and Structures
journal homepage: www.unistroy.spbstu.ru
Строительство уникальных зданий и сооружений

Influence of clock frequency on rendering time

A.P. Marichev [1], E.S. Zalata [2]*, O.S. Grishina [3], G.A. Kukushkina[4], A.E. Fedorec[5]

[1-5]*Peter the Great St. Petersburg Polytechnic University,29 Politechnicheskaya St., St. Petersburg, 195251, Russia*

Article info

scientific article

doi: 10.18720/CUBS.55.7

Article history

Received: 05.04.2017

Keywords

render;
civil engineering;
overclocking;
3D modelling;
Autodesk;
Revit;
Lumion;
NVidia;
Artlantis;
V-ray;

ABSTRACT

One of the most important factors for the successful realization of a construction project is the use of advanced software in the field of architecture. Nowadays there are many different programs for visualization. It is increasingly used to help decision-makers make economically optimal decisions. However, lack of information about interconnection of these programs and computer components leads to incorrect use of the program as well as increasing of time spent for visualization. This paper describes a method of reducing the time spent for rendering a final scene by changing the clock frequency of the computer. The data that have been obtained show changes in the rendering time after using Overclocking process for different visualization programs.

Content

Contact information:

1 +7(921)5870750, lexa-ap94@yandex.ru (Alexey Marichev, Student)
2* +7(921)4078184, zolotka95@mail.ru (Ekaterina Zalata, Student)
3 +7(911)8372393, olgasergeevna2009@yandex.ru (Olga Grishina, Student)
4 +7(911)9761311, kukushkina_ga@mail.ru (Galina Kukushkina, Assistant)
5 +7(964)3642647, toxa.sid@yandex.ru (Anton Fedorec, Student)

1. Introduction

There are many different programs for visualization. With the help of these programs decision-makers choose the best solutions in the construction field. Lack of information about interconnection of these programs and computer components leads to incorrect use of the program as well as increasing of time spent for visualization. The object of the study is time spent for visualization in three programs: Revit, Artlantis, Lumion.

The majority of architectural objects are designed with 3D modeling [1-3]. Thereby 3D technologies play an important role in the construction industry. Software in the field of architectural visualization has a large range however the aim of all these programs is the same - to provide a good quality realistic picture as soon as possible [4, 5].

Despite the relevance of the information about the presentation of construction projects, the amount of literature in this field is extremely small [6, 7]. Most of the publications and manuals, considering the subject, are methodological and explain how to work in three-dimensional environment of software systems [8-12]. Existing reviews of programs for rendering does not consider all the technical capabilities of these programs in details [13, 14]. These reviews also do not consider dependence of a system from external influences [15-17]. At the same time lack of information about interconnection of programs and computer components leads to incorrect use of the product as well as increasing of time spent for visualization [18-20].

The aim of this article is to compare the rendering time before and after using Overclocking process in different visualization programs and using different engines. For comparison of rendering time three visualization programs are chosen: Revit, Artlantis, Lumion. These programs can use such engines as Nvidia Mental ray, Autodesk Raytracer, V-ray, Indigo Render, Octan Render, Maxwell, Artlanis, Lumion.

Tasks to be achieved:

- Calculate the duration of renderings with standard engines
- Calculate the duration of renderings after Overclocking
- Compare the results

2. Methods and Results

For comparison of architectural visualization programs laptop LENOVO IdeaPad Z580 was used.

Specifications:

- Processor - Intel Core I5-3210M
- RAM – 8 GB
- Graphics Card - Nvidia GeForse GT 630M
- Cuda processor on the Fermi architecture

The main purpose of architectural visualization is a graphical representation of the designed buildings or surrounding areas.

There are two main families of visualization engines with different behavior during the rendering process - preconceived and not preconceived. Preconceived method means that you can determine the assumptions that render can do in the process of calculating the pixels. This allows you to specify for the rendering engine whether to do approach or not. Not preconceived method does not provide assumptions.

In addition, all engines can be divided into three categories according to the method of computing the final scene:

- CPU
- GPU
- Hybrid

CPU is a central processing unit. Programs use the CPU power. GPU is a graphical processing unit (graphics card). Programs use the power of a separate chip on the graphics card. The third category that uses a hybrid type is based on the use of two processors at once.

As a scene for rendering we use an example which provides company Autodesk when installing Revit 2017.

Revit is a full-featured CAD system offering possibilities of architectural design, design of engineering systems and building structures, as well as the visualization. It provides a high accuracy of ongoing projects. It is

based on the technology of building information modeling - BIM. This system provides a high level of cooperation of specialists in various areas and significantly reduces the number of errors. It allows you to create constructions and engineering systems of any complexity. With the projected model specialists have the opportunity to develop an effective technology of construction and accurately determine the required amount of materials.

First, we make rendering using standard engines Nvidia Mental ray and Autodesk Raytracer. All the settings are set to maximum, image size is 860x470 and depth is 150dpi. Nvidia Mental ray is available for users with a graphics card from Nvidia family. It belongs to the category GPU, preconceived.

Calculation of the scene was completed in 2 hours and 1 minute. During this process, 4.29 GB of RAM was used. Findings are summarized in Table 1.

Table 1. Nvidia Mental ray

Program	Engine	Duration	RAM	CPU/GPU	Type
Revit	Nvidia Mental ray	2:01	4,29 GB	GPU	Preconceived

The engine Autodesk Raytracer is a standard engine for visualizations in AutoCAD, Navisworks and Showcase, which ensures consistency of visualization results for all Autodesk products. When using Autodesk Raytracer visualization tools it is possible to define quality settings.

Calculation of the scene was completed in 2 hours and 17 minutes. During this process, 4.42 GB of RAM was used. Findings are summarized in Table 2.

Table 2. Autodesk Raytracer

Program	Engine	Duration	RAM	CPU/GPU	Type
Revit	Autodesk Raytracer	2:17	4,42 GB	CPU	Preconceived

Next, the add-on for Revit V-ray by Chaos Group was installed. V-ray is a rendering system in which there are several algorithms for calculating global illumination. This system has good render scenes in many areas of visualization because of its flexibility and wide range of instruments.

Calculation of the scene was completed in 2 hours and 18 minutes. During this process, 5.41 GB of RAM was used. Findings are summarized in Table 3.

Table 3. V-Ray

Program	Engine	Duration	RAM	CPU/GPU	Type
Revit	V-Ray	2:18	5,41 GB	CPU	Preconceived

Indigo Render is a program based on the calculation of components of scenes on a physical level. This program uses its own rendering algorithms by which the program can count separate entities or the whole scene. The program has its own library of objects that can improve the interactivity of the scene.

Calculation of the scene was completed in 55 minutes. During this process, 5.39 GB of RAM was used. Findings are summarized in Table 4.

Table 4. Indigo Render

Program	Engine	Duration	RAM	CPU/GPU	Type
Revit	Indigo Render	0:55	5,39 GB	GPU	Preconceived

Octane Render - graphics real-time engine developed by Refractive Software LTD. This engine uses CUDA processor and runs on all GPUs of Nvidia. Unlike the full version that uses a video card, this extension uses the CPU power.

Calculation of the scene was completed in 3 hours and 27 minutes. During this process, 3.27 GB of RAM was used. Findings are summarized in Table 4.

Table 5. Indigo Render

Program	Engine	Duration	RAM	CPU/GPU	Type
Revit	Octan Render	3:27	3,27 GB	CPU	Not preconceived

Artlantis is a program that allows you to make fast high quality visualizations especially for architects and designers. There are two main versions of Artlantis designed for different tasks - Artlanis Studio and Artlanis Render. The article deals with Artlantis Render as it is better suited for static rendering. Artlanis Studio is used for animated presentation of projects.

In the program it is possible to choose one of the two engines for rendering – Maxwell and Artlantis.

Maxwell Render is the first released rendering system with a so-called physical model. The whole system is based on mathematical equations that describe the behavior of light. For this reason, visualization of objects is made without assumptions. This helps to avoid long setting of the visualization parameters.

Calculation of the scene was completed in 1 hour and 30 minutes. During this process, 4.86 GB of RAM was used. Findings are summarized in Table 6.

Table 6. Maxwell Render

Program	Engine	Duration	RAM	CPU/GPU	Type
Artlantis	Maxwell Render	1:30	4,86 GB	CPU	Not preconceived

Calculation of the scene was completed in 1 hour and 49 minutes. During this process, 4.48 GB of RAM was used. Findings are summarized in Table 7.

Table 7. Artlantis Render

Program	Engine	Duration	RAM	CPU/GPU	Type
Artlantis	Artlantis Render	1:49	4,48 GB	GPU	Preconceived

Lumion - presentation program needed primarily for architects, designers, artists. This product is designed for a user who has no possibility or cannot deal with "heavy" professional rendering programs with complex interface.

Calculation of the scene was completed in 2 minutes. During this process, 5.22 GB of RAM was used. Findings are summarized in Table 8.

Table 8. Lumion

Program	Engine	Duration	RAM	CPU/GPU	Type
Lumion	Lumion	0:02	5,22 GB	CPU+GPU	Preconceived

All the results are combined into a single table 9.

Table 9. Summary table

Program	**Engine**	**Duration, min**	**RAM, GB**	**CPU/GPU**	**Type**
Revit	Nvidia Menal ray	121	4,29	GPU	Preconceived
	Autodesk Raytrcer	137	4,42	CPU	Preconceived
	V-ray	138	5,41	CPU	Preconceived
	Indigo Render	55	5,39	GPU	Preconceived
	Octan Render	207	3,27	CPU	Not preconceived
Artlantis	Maxwell	90	4,86	CPU	Not preconceived
	Artlanis	109	4,48	GPU	Preconceived
Lumion	Lumion	2	5,22	CPU+GPU	Preconceived

Based on the table 9 two figures can be made:

- Time spent for rendering a final scene for each engine;
- RAM usage for each engine.

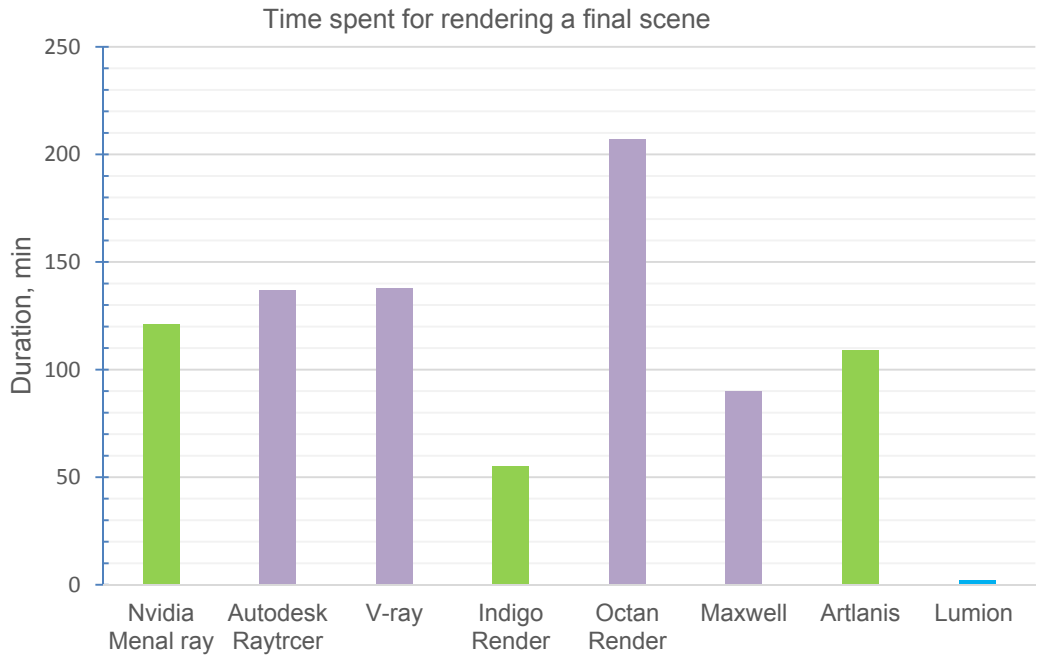

Fig. 1. Time spent for rendering a final scene

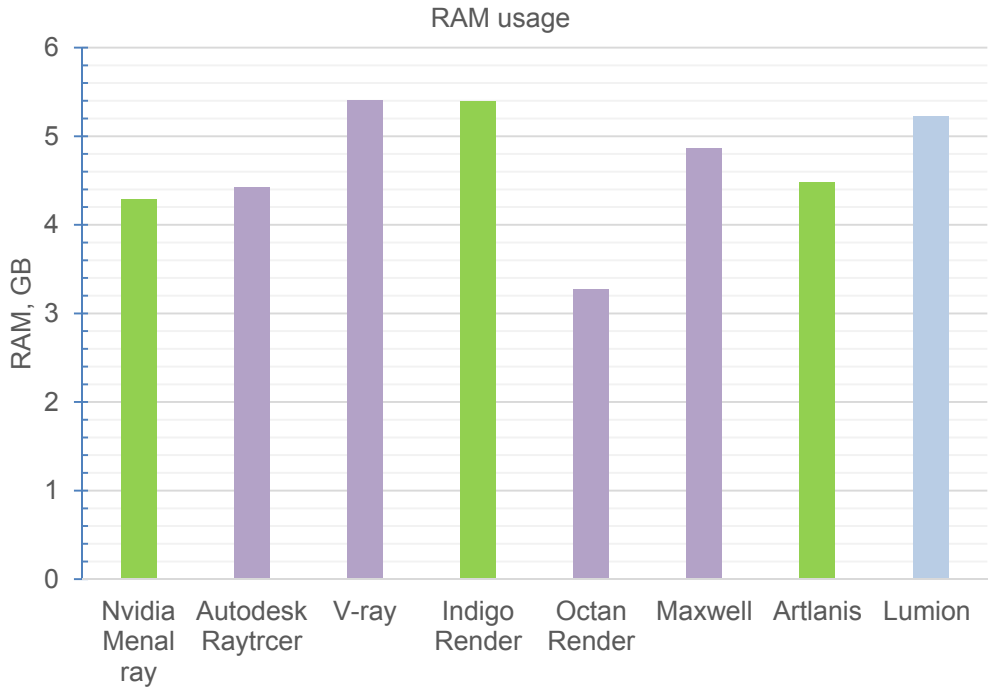

Fig. 2. RAM usage

Figure 2 shows that comparison of Lumion hybrid system with others is beside the purpose, since time values are very different. In general, engines using graphical processing unit (GPU) spend less time for rendering the final scene than engines using central processing unit (CPU).

Overclocking is a process of increasing the frequency (and voltage) of a component in a computer over the regular mode to increase the speed of computer's work. However, this operation can cause very high heat release, power consumption and loud noise. Therefore, to reduce the risk of damage to the computer, frequency was increased only to 50%.

Renders for all the engines were made again and after that were collected in table 10.

Table 10. Summary table after Overclocking

Program	Engine	Duration, min	RAM, GB
Revit	Nvidia Mental ray	62	4,82
	Autodesk Raytracer	102	4,97
	V-ray	103	6,08
	Indigo Render	27	6,06
	Octan Render	155	3,68
Artlantis	Maxwell	68	5,47
	Artlanis	56	5,04
Lumion	Lumion	1	5,87

Figure 3 shows the comparison of time spent for rendering a final scene before and after Overclocking. In hybrid systems (CPU + GPU) the difference is insignificant. Therefore, comparison of time for Lumion is not shown on the figure 3. Values obtained after Overclocking are made with hatching.

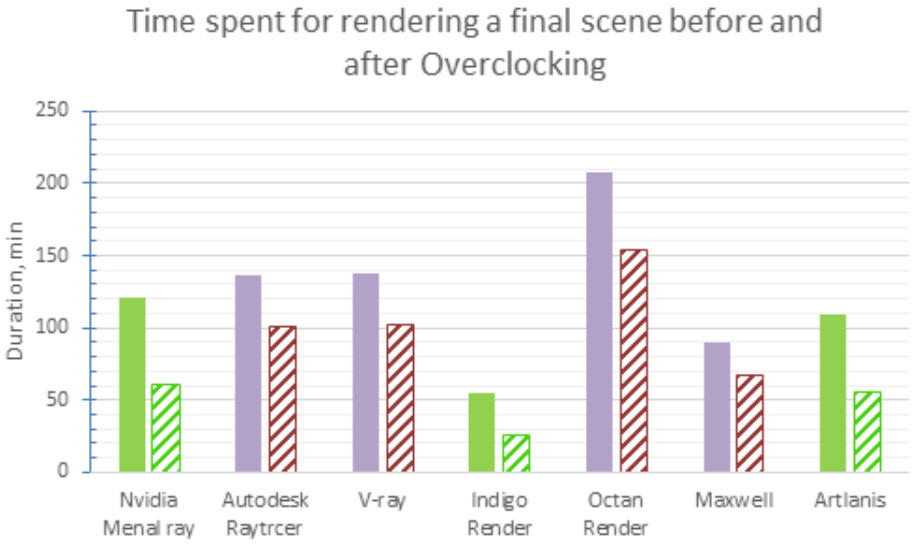

Fig. 3. Time spent for rendering a final scene before and after Overclocking.

3. Discussion

As expected, clock frequency increase reduces rendering time. In addition, it is possible to increase the supply voltage to the components to improve computer performance that means reducing the rendering time. Furthermore, temperature also directly affects computer performance. Since some renderings were carried out without a break and intentional cooling of chips, we should take into account the influence of heat. For ideal conditions it is recommended to keep the temperature of the chips within 50-60 degrees Celsius and 15-20 degrees Celsius for the environment [23]. If we consider new computers with Intel I7 processor and graphics card Nvidia GeForce GTX 1080 then rendering time will be significantly reduced because the technical specifications between new computers and the computer used for research differ a lot.

Most of the publications and manuals, considering the subject, are methodological and explain how to work in three-dimensional environment of software systems [8-12]. Existing reviews of programs for rendering does not consider all the technical capabilities of these programs in details [13, 14]. Authors could not find any articles about changing of clock frequency for better visualization results. To the authors' knowledge, this research is the first which shows clear difference before and after overclocking for visualization programs.

4. Conclusions

As the result, rendering time significantly reduced after increasing the clock frequency for 50 %

Time difference is:

Nvidia Mental ray – decreased by 58 minutes;

Autodesk Raytracer – decreased by 30 minutes;

V-ray – decreased by 37 minutes;

Indigo Render – decreased by 27 minutes;

Octan Render – decreased by 47 minutes;

Maxwell – decreased by 22 minutes;

Artlanis – decreased by 54 minutes.

Thus, increasing of the clock frequency by 50% reduces the rendering time for programs that use the central processing unit (CPU) by 20-25% and for programs that use graphical processing unit (GPU) by 45-50%.References

References

[1] Gindis E. Chapter 30 - Lighting and Rendering. Up and Running with AutoCAD 2017. 2017. Pp. 675-693

[2] Xue J., Yao J., Lu K., Shao L., Rahman M. M. Efficient volume rendering methods for out-of-Core datasets by semi-adaptive partitioning. Information Sciences. Vol.370–371. 2016. Pp. 463-475

[3] Ismail M. A., Mahmud R., Isham S. H. Digital Studio vs. Conventional in Teaching Architectural Design Process. Procedia - Social and Behavioral Sciences. Vol. 64. 2012. Pp. 18-25

[4] Vysotskiy A., Makarov S., Zolotova Y., Tuchkevich E., Features of BIM Implementation Using Autodesk Software. Procedia Engineering. Vol. 117. 2015. Pp. 1143-1152

[5] Kleijn M., Hond R., Martinez-Rubi O., A 3D spatial data infrastructure for Mapping the Via Appia. Digital Applications in Archaeology and Cultural Heritage. Vol. 3. Issue 2. 2016. Pp 23-32

[6] Steen J., Boardman T., Chapter 7 - Revit and mental ray. Rendering with mental ray and 3ds Max (Second Edition). 2010. Pp 214-240

[7] Cheng-Ting .C, Chun-Ping C., Chien-Cheng C., BIM-enabled Power Consumption Data Management Platform for Rendering and Analysis of Energy Usage. Patterns Procedia Engineering. Vol. 118. 2015. Pp. 554-562

[8] Johansson .M,Roupé R., Bosch-Sijtsema P. Real-time visualization of building information models (BIM). Automation in Construction. Vol. 54. 2015. Pp. 69-82

[9] Shoubi M.V., Shoubi M.V., Bagchi A., Barough A.S., Reducing the operational energy demand in buildings using building information modeling tools and sustainability approaches. Ain Shams Engineering Journal. Vol. 6. Issue 1. 2015. Pp. 41-55

[10] Smith B.L. Chapter 11 - Introduction to mental ray 3ds Max. Design Architectural Visualization. 2012. Pp. 275-283

[11] Steen J., Boardman T., Chapter 4 - mental ray® and materials. Rendering with mental ray and 3ds Max. 2010. Pp. 76-109

[12] Shakaev V., Shabalina O., Kamaev V., Chickerur S., LOW-LEVEL API AGNOSTIC RENDERING INTERFACEFOR BRIDGING OPENGL AND DIRECT3D. Innovacionnye informacionnye texnologii. № 1. 2014. Pp 243-252.

[13] Sugimoto Y., Ino F., Hagihara K., Improving cache locality for GPU-based volume rendering. Parallel Computing. Vol. 40. Issues 5–6. 2014. Pp 59-69

[14] Rossi M., Marini D., Rizzi A., Methods and application for photorealistic rendering and lighting of ancient buildings. Journal of Cultural Heritage. Vol. 5. Issue 3. 2004. Pp 291-300

[15] Lu D., Zhu D., Wang Z., Texture-Based Multiresolution Flow Visualization. Jisuanji Yanjiu yu Fazhan. 2015. T. 52. № 8. Pp. 1910-1920.

[16] Maia L., Mêda P., Freitas J.G., BIM Methodology, a New Approach - Case Study of Structural Elements Creation Procedia. Engineering. Vol. 114. 2015. Pp. 816-823

[17] Künzel H., Criteria Defining Rain Protecting External Rendering Systems. Energy Procedia. Vol. 78. 2015. Pp. 2524 2529

[18] Laura D., Cristian B., Simonetta F., Ornella L.R., Alessandro R., Assessing Color Rendering in a 3d Setup. Energy Procedia. Vol. 78. 2015. Pp. 705-710

[19] Mukhina K., Bezgodov A., The Method for Real-time Cloud Rendering. Procedia Computer Science. Vol. 66. 2015. Pp. 697-704

[20] Wang L., Hongfei L., Zhiqiang M., A Method of Real-time Rendering for Subsurface Scattering Based on BRDF. Procedia Engineering. Vol. 29. 2012. Pp 1565-1572

[21] Linwei Z., Mei Y., Gangyi J., Xiangying M., Songyin F., Ting L., A New Virtual View Rendering Method based on Depth Map for 3DTV3DTV; virtual view rendering; depth; false contour; holes filling; image restoration. Procedia Engineering. Vol. 15. 2011. Pp 1115-1119

[22] Jian Y., Jing-Xiong C., Xiao-Qing C., He-Ping T., Modeling and Rendering of Real-time Large-scale Granular Flow Scene on GPU. Procedia Environmental Sciences. Vol. 10. Part B. 2011. Pp 1035-1045

[23] Thomas R.K., Thomas M.T., GPU for AutoCAD. Computer-Aided Design and Applications Computer-Aided Design and Applications. Vol. 4. 2013. Pp. 853-862.

Влияние тактовой частоты на скорость рендера

А.П. Маричев[1], Е.С. Залата[2*], О.С. Гришина[3], Г.А. Кукушкина[4], А.Е. Федорец[5]

[1-5]Санкт-Петербургский политехнический университет Петра Великого, 195251, Россия, г. Санкт-Петербург, Политехническая ул., 29.

ИНФОРМАЦИЯ О СТАТЬЕ	ИСТОРИЯ	КЛЮЧЕВЫЕ СЛОВА
doi: 10.18720/CUBS.55.7	Подана в редакцию: 05.04.2017	Рендер; оверклокинг; 3д моделирование; Autodesk; Revit; Lumion; NVidia; Artlantis; V-ray;

АННОТАЦИЯ

Одним из наиболее важных факторов успешной реализации строительного проекта является использование передового программного обеспечения в области архитектуры. В настоящее время существует множество различных программ для визуализации. Они все чаще используются для того, чтобы помочь лицам, принимающим решения, выбрать экономически оптимальный вариант. Однако отсутствие информации о взаимосвязи этих программ и компьютерных компонентов приводит к неправильному использованию самих программ, а также к увеличению времени, затрачиваемого на визуализацию. В этой статье описывается способ сокращения времени, затраченного на рендеринг финальной сцены, путем изменения тактовой частоты компьютера. Полученные данные показывают изменения во времени рендеринга после изменения тактовой частоты процессора для разных программ визуализации.

Контакты авторов:

1 +7(921)5870750, lexa-ap94@yandex.ru (Маричев Алексей Павлович, студент)
2* +7(921)4078184, zolotka95@mail.ru (Залата Екатерина Сергеевна, студент)
3 +7(911)8372393, olgasergeevna2009@yandex.ru (Гришина Ольга Сергеевна, студент)
4 +7(911)9761311, kukushkina_ga@mail.ru (Кукушкина Галина Андреевна, ассистент)
5 +7(964)3642647, toxa.sid@yandex.ru (Федорец Антон Евгеньевич, студент)

Marichev A.P., Zalata E.S., Grishina O.S., Kukushkina G.A., Fedorec A.E. Influence of clock frequency on rendering time/
Маричев А.П., Залата Е.С., Гришина О.С., Кукушкина Г.А., Федорец А.Е. Влияние тактовой частоты на скорость рендера ©

Литература

[1] Gindis E. Chapter 30 - Lighting and Rendering. Up and Running with AutoCAD 2017. 2017. Pp. 675-693

[2] Xue J., Yao J., Lu K., Shao L., Rahman M. M. Efficient volume rendering methods for out-of-Core datasets by semi-adaptive partitioning. Information Sciences. Vol.370–371. 2016. Pp. 463-475

[3] Ismail M. A., Mahmud R., Isham S. H. Digital Studio vs. Conventional in Teaching Architectural Design Process. Procedia - Social and Behavioral Sciences. Vol. 64. 2012. Pp. 18-25

[4] Vysotskiy A., Makarov S., Zolotova Y., Tuchkevich E., Features of BIM Implementation Using Autodesk Software. Procedia Engineering. Vol. 117. 2015. Pp. 1143-1152

[5] Kleijn M., Hond R., Martinez-Rubi O., A 3D spatial data infrastructure for Mapping the Via Appia. Digital Applications in Archaeology and Cultural Heritage. Vol. 3. Issue 2. 2016. Pp 23-32

[6] Steen J., Boardman T., Chapter 7 - Revit and mental ray. Rendering with mental ray and 3ds Max (Second Edition). 2010. Pp 214-240

[7] Cheng-Ting .C, Chun-Ping C., Chien-Cheng C., BIM-enabled Power Consumption Data Management Platform for Rendering and Analysis of Energy Usage. Patterns Procedia Engineering. Vol. 118. 2015. Pp. 554-562

[8] Johansson .M,Roupé R., Bosch-Sijtsema P. Real-time visualization of building information models (BIM). Automation in Construction. Vol. 54. 2015. Pp. 69-82

[9] Shoubi M.V., Shoubi M.V., Bagchi A., Barough A.S., Reducing the operational energy demand in buildings using building information modeling tools and sustainability approaches. Ain Shams Engineering Journal. Vol. 6. Issue 1. 2015. Pp. 41-55

[10] Smith B.L. Chapter 11 - Introduction to mental ray 3ds Max. Design Architectural Visualization. 2012. Pp. 275-283

[11] Steen J., Boardman T., Chapter 4 - mental ray® and materials. Rendering with mental ray and 3ds Max. 2010. Pp. 76-109

[12] Shakaev V., Shabalina O., Kamaev V., Chickerur S., LOW-LEVEL API AGNOSTIC RENDERING INTERFACEFOR BRIDGING OPENGL AND DIRECT3D. Innovacionnye informacionnye texnologii. № 1. 2014. Pp 243-252.

[13] Sugimoto Y., Ino F., Hagihara K., Improving cache locality for GPU-based volume rendering. Parallel Computing. Vol. 40. Issues 5–6. 2014. Pp 59-69

[14] Rossi M., Marini D., Rizzi A., Methods and application for photorealistic rendering and lighting of ancient buildings. Journal of Cultural Heritage. Vol. 5. Issue 3. 2004. Pp 291-300

[15] Lu D., Zhu D., Wang Z., Texture-Based Multiresolution Flow Visualization. Jisuanji Yanjiu yu Fazhan. 2015. T. 52. No. 8. Pp. 1910-1920.

[16] Maia L., Mêda P., Freitas J.G., BIM Methodology, a New Approach - Case Study of Structural Elements Creation Procedia. Engineering. Vol. 114. 2015. Pp. 816-823

[17] Künzel H., Criteria Defining Rain Protecting External Rendering Systems. Energy Procedia. Vol. 78. 2015. Pp. 2524 2529

[18] Laura D., Cristian B., Simonetta F., Ornella L.R., Alessandro R., Assessing Color Rendering in a 3d Setup. Energy Procedia. Vol. 78. 2015. Pp. 705-710

[19] Mukhina K., Bezgodov A., The Method for Real-time Cloud Rendering. Procedia Computer Science. Vol. 66. 2015. Pp. 697-704

[20] Wang L., Hongfei L., Zhiqiang M., A Method of Real-time Rendering for Subsurface Scattering Based on BRDF. Procedia Engineering. Vol. 29. 2012. Pp. 1565-1572

[21] Linwei Z., Mei Y., Gangyi J., Xiangying M., Songyin F., Ting L., A New Virtual View Rendering Method based on Depth Map for 3DTV3DTV; virtual view rendering; depth; false contour; holes filling; image restoration. Procedia Engineering. Vol. 15. 2011. Pp. 1115-1119

[22] Jian Y., Jing-Xiong C., Xiao-Qing C., He-Ping T., Modeling and Rendering of Real-time Large-scale Granular Flow Scene on GPU. Procedia Environmental Sciences. Vol. 10. Part B. 2011. Pp 1035-1045

[23] Thomas R.K., Thomas M.T., GPU for AutoCAD. Computer-Aided Design and Applications Computer-Aided Design and Applications. Vol. 4. 2013. Pp. 853-862.

Marichev A.P., Zalata E.S., Grishina O.S., Kukushkina G.A., Fedorec A.E. Influence of clock frequency on rendering time. Construction of Unique Buildings and Structures. 2017. 4(55). Pp. 50 -66

Маричев А.П., Залата Е.С., Гришина О.С., Кукушкина Г.А., Федорец А.Е. Влияние тактовой частоты на скорость рендера, Строительство уникальных зданий и сооружений, 2017, №4 (55). С. 50-66.