Преглед НЦД 27 (2015), 35–42

Nikola B. Gajić, Dušan B. Gajić, Radomir S. Stanković
Faculty of Electronic Engineering, University of Niš

PERFORMANCE ANALYSIS OF LOCAL AND DISTRIBUTED RENDERING METHODS FOR PREPARING VIDEO CONTENTS

Abstract. The process of digitalization of cultural and historical heritage often requires methods which involve construction of three-dimensional (3D) computer models and their rendering onto computer displays. Algorithms which are used for this purpose demand intense computations. This makes them time-consuming, even with the computing power available on contemporary computers. One of possible solutions for this problem is to distribute computations over multiple computer systems. For the purpose of distributed generation of video contents based on 3D scenes, we apply Autodesk Backburner and perform a comparative analysis of this method and traditional local rendering. We study characteristics of the local and distributed rendering processes in terms of computation times. We measure the influence of scene complexity, determined by the number of polygons, size of textures, and settings of the ray-tracing system, on the run-time of computations. Based on these experiments, we offer several conclusions in the form of recommendations. Besides the complexity of processed scenes, these recommendations are formulated by taking into account the characteristics of used distributed systems, such as network throughput. We believe that the presented information can be useful for practitioners dealing with 3D scenes of different complexity in diverse computing environments.

Keywords. Computer graphics and animation, distributed rendering, performance analysis.

1. Introduction

Modern information technologies offer novel possibilities for better presentation of cultural and historical heritage. Their application creates an enhanced perspective on historical sites and museum exhibitions, through addition of extra information about existing contents or by including completely new virtual contents. For example, computer graphics and animation allow digital reconstruction of long-gone buildings and monuments. Application of augmented reality technologies on mobile devices allows virtual inclusion of these digital restorations into real-world physical environments. However, the intensity of computations involved in rendering of computer graphics, which arises from the complexity of used scenes, leads to prolonged times needed for creation of animations [1].

Distributed (parallel) rendering represents an application of parallel computing which can improve the performance of computations in computer graphics [7]. Since graphics rendering belongs to a class of problems which are inherently and embarrassingly parallel on several computational levels, such as pixels and objects, as well as complete frames, it can benefit dramatically from parallel processing [6]. Therefore, in this paper we discuss the complexity of local and distributed rendering of animations based on three-dimensional scenes featuring models of ancient buildings. Conclusions offered in the paper are aimed at reducing overall times needed for preparing video contents in the process of digitalization of cultural and historical heritage.

The paper is organized as follows. Section 2 considers rendering methods and software technologies used in the rendering process. In Section 3, we describe the experimental environment that is used and present the results of experiments. The paper ends with conclusions and recommendations for the practitioners using distributed rendering of computer graphics.

2. Local and Distributed Rendering

Rendering is the process of creation of a two-dimensional (2D) image from a three-dimensional (3D) model or a collection of 3D models (called a *scene*) [4]. The core of the rendering process is the computation of effects that each geometric primitive in a 3D scene has on each pixel of a 2D screen (a *frame*) [7]. These underlying computations are very intensive, especially when photo-realistic images with fast updates are needed [7]. Animations, based on changes or different views on 3D scenes, require a great number of frames to be rendered (typically 25 or 30 frames per second of animation) [1].

The size of textures and the number of polygons have different influence on rendering times, as it will be discussed later in the paper. In terms of their complexity, 3D scenes can be divided into the following classes:

- Texture-intensive (TI) scenes, i.e., scenes with a small number of polygons and large texture files,
- Polygon-intensive (PI) scenes, i.e., scenes featuring models with a large number of polygons (measured in thousands), and
- Texture-and-polygon-intensive (TPI) scenes, i.e., 3D scenes with complex models with both thousands of polygons and large texture files.

Rendering can be performed *locally*, i.e., on a single computer. In this case, there is no need for data distribution and the direct product of rendering can be either separate images, corresponding to frames, or a complete video file.

In a distributed environment, composed of several computer systems connected through a computer network (Figure 1), each computer is considered as a *node*. *Distributed rendering* allows the division of the computational task between several nodes in the computer network and, thus, decreases the total time needed for preparing animations. One of nodes in the network is declared to be the *server*, while other nodes are used as *clients*. Server controls data distribution between nodes and the distributed rendering process, and also takes part in rendering, while clients exclusively perform rendering. The most often used approach in distributed rendering is to assign different frames in the animation to different nodes, since each frame can be generated independently [7]. This approach is also used in the presented research.

Before the rendering process starts, it is necessary to transfer data for the scene to be rendered to each node in the network. These data can be quite large, i.e., the order of GB, especially for texture-intensive scenes. The communication between nodes prevents the speed-up from distributed rendering to scale linearly with the increase of the number of computing nodes. Since it is necessary to transfer complete scene data and assign frames to be rendered by each node before the distributed rendering starts, there is no need for additional communication between nodes during rendering.

Further, rendering in a distributed environment imposes certain limitations on rendering parameters, such as the need to render frames into separate image files. After distributed rendering of all frames is completed, these images need to be connected together into the final animation, which can also be a time-consuming process. Therefore, in the case of the distributed rendering, the total time, needed for producing video files (t_{total}), consists of time for preparing the rendering environment (t_{setup}), time for rendering (t_{render}), and additional time for concatenating frames (t_{concat}) produced by nodes into a single video

$$t_{total} = t_{setup} + t_{render} + t_{concat} \,. \tag{1}$$

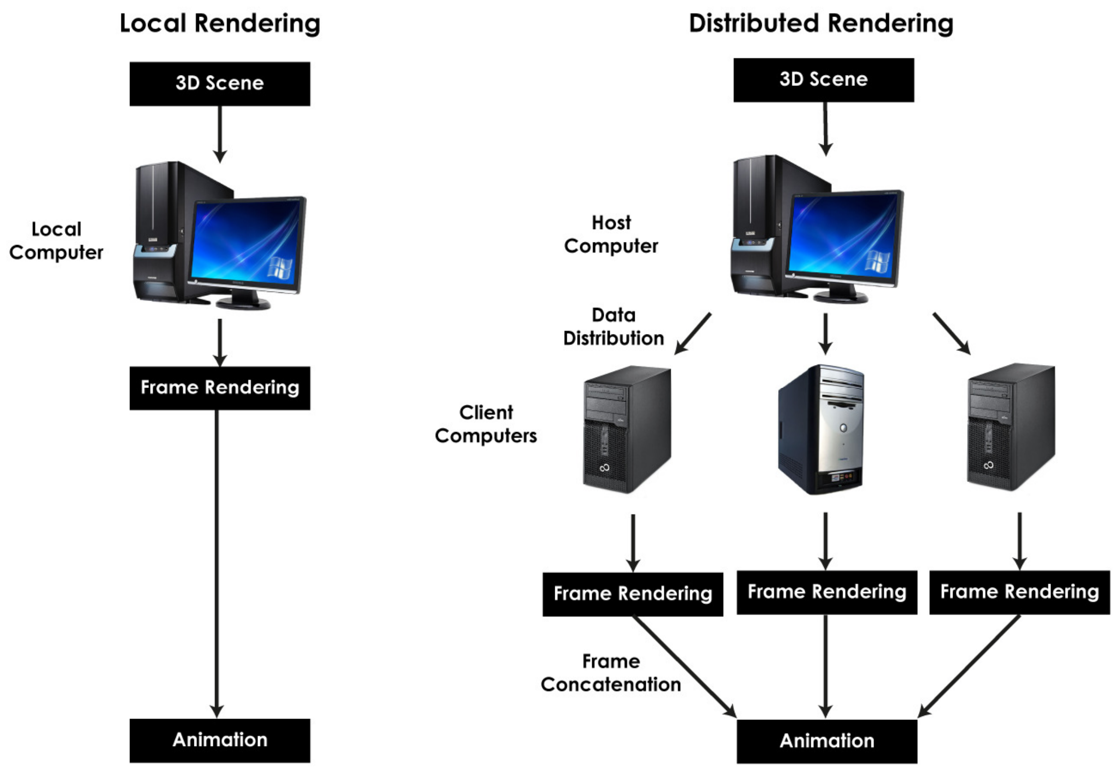

Figure 1: Local and distributed rendering.

In the presented research, *Autodesk 3D Studio Max* was used for creating 3D scenes. *3D Studio Max*, together with *Autodesk Maya*, is currently the dominant 3D computer graphics program used by video game developers, movie studios, and architectural design bureaus. Distributed rendering was performed using *Autodesk Backburner* since it is a standard tool for handling rendering and compiling tasks for a range of software tools developed by Autodesk [3]. It allows sending of blocks of data from individual applications, such as *3D Studio Max*, to many different render nodes at once. *Backburner* is composed of three components (*Manager*, *Server*, and *Monitor*), out of which *Manager* and *Monitor* run on a single machine (chosen as the host), and *Server* runs on all of the systems taking part in the rendering process (one host and three clients in the case of the presented research).

3. Performance Comparison of Local and Distributed Rendering

In order to measure the performance of local and distributed rendering we performed a series of experiments. We first describe the experimental environment and the used test animations. Experimental results are afterwards presented.

3.1 Experimental environment. Specifications of the used computer systems are presented in **Table I**. *Autodesk 3D Studio Max 2012 Design* was used for building 3D models and scenes used in the reported experiments [2]. The rendered frames were connected into animations using video capture and video processing tool called

VirtualDub [10]. To obtain more realistic rendering results, instead of default *3D Studio Max* rendering algorithms, we applied *V-Ray Adv 2.10.01* set of rendering algorithms [5, 8]. *V-Ray* uses advanced rendering techniques, including global illumination algorithms, such as path tracing and photon mapping [1].

Table I
Specification of the experimental environment.

	Single	*Distributed*			
Number of computers	1	4			
CPU	i7-2600K	i7-2600K	i7-2600K	i5-2320	i5-2320
Frequency	3.4 GHz	3.4 GHz	3.4 GHz	3 GHz	3 GHz
Number of cores	4	4	4	4	4
RAM	16 GB	16 GB	16 GB	8 GB	8 GB
Local network	100BASE-TX 100 Mbit				
Operating system	*Windows 7 Ultimate* 64-bit				
3D modeling tool	*Autodesk 3ds Max Design 2012* 64-bit + *V-Ray Adv 2.10.01*				
Distributed processing tool	*Autodesk Backburner 2012*				

3.2 Test animations. During the experiments, computation times were measured with respect to the following two factors:

- Rendering method (local or distributed), and
- Scene complexity, in terms of the three complexity classes discussed in Section 2.

We analyzed animations rendered based on the following three scenes of different complexity, used as the representatives of the complexity classes described in Section 2:

1. *The Southern Church* (Figure 2) – a texture-intensive scene, with a total file size of 2,320 MB, but only 342 polygons and a single light source,
2. *The Horreum* (Figure 3) – a polygon-intensive scene, with 10,441 polygons, one light source augmented by high-dynamic-range (HDR) imaging [9] and small textures, leading to a total file size of only 3.29 MB, and
3. *The Villa with the Peristyle* (Figure 4) – a texture-and-polygon-intensive scene, with 336,523 polygons, a single light source, HDR imaging and large textures, which produces a file size of 238.3 MB.

A sample frame from each of the three animations is shown in Figure 2, Figure 3, and Figure 4, respectively. The scenes were developed as a part of the digital reconstruction of the ancient Roman site of Mediana, the birth place of the Roman emperor Constantine the Great, in the vicinity of the present day city of Niš, Serbia. All animations were rendered in 1920 by 1080 pixels (FullHD) resolution. Notice that, in the case of distributed rendering, additional time after the rendering process is completed is needed to create animations from frames. For the considered animations, these additional times are in the range of 5 to 7 minutes. For each animation, three test runs were performed, for both local and distributed rendering tests.

3.3 Experimental results. Results of experiments for considered animations are shown in Table II, Table **III**, and Table **IV**, respectively. The summary of the results for distributed and local rendering times for these animations, in terms of the number of frames, is given in Figure 5.

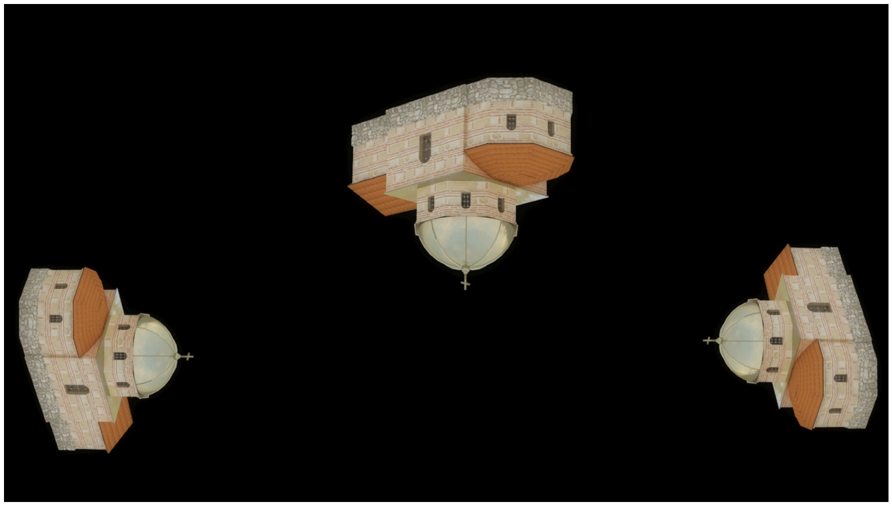

Figure 2: A frame from *The Southern Church* animation.

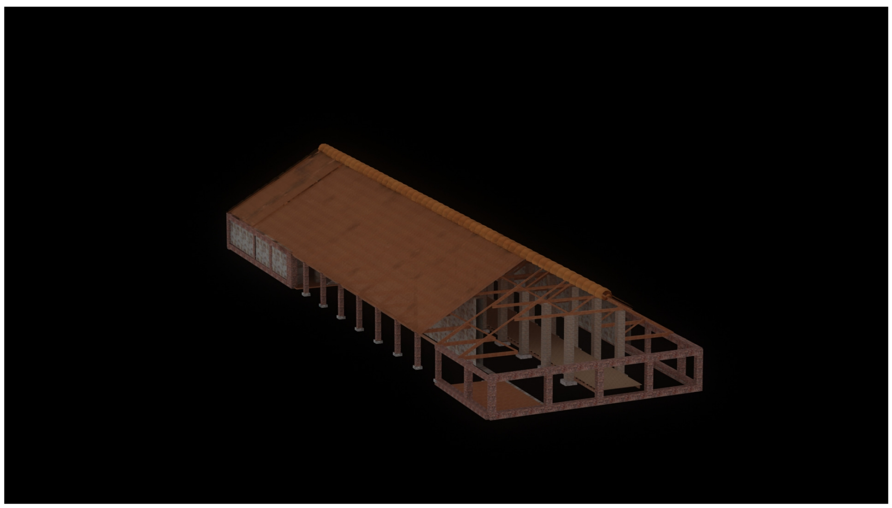

Figure 3: A frame from *The Horreum* animation.

Figure 4: A frame from *The Villa with the Peristyle* animation.

Table II
Experimental results – *The Southern Church* animation.

Number of computers	*Test run*	*Render start*	*300th frame*	*600th frame*	*900th frame*
1	1st	0:00	8:11	15:32	25:32
	2nd	0:00	8:23	15:43	25:45
	3rd	0:00	8:09	15:42	25:42
4	1st	4:44	9:25	12:34	15:06
	2nd	4:46	9:38	12:40	15:32
	3rd	4:38	9:18	12:27	14:55

Table III
Experimental results – *The Horreum* animation.

Number of computers	*Test run*	*Render start*	*300th frame*	*600th frame*
1	1st	0:00	12:43:04	23:54:15
	2nd	0:00	12:55:27	22:10:32
	3rd	0:00	12:42:53	23:23:33
4	1st	1:07	7:05:16	13:18:35
	2nd	1:32	7:13:07	14:32:23
	3rd	1:14	6:58:07	13:35:03

Table IV
Experimental results – *The Villa with the Peristyle* animation.

Number of computers	*Test run*	*Render start*	*150th frame*	*300th frame*
1	1st	0:00	34:26:33	61:42:33
	2nd	0:00	34:56:21	58:55:12
	3rd	0:00	33:47:23	60:27:33
4	1st	3:48	8:20:03	18:01:00
	2nd	3:45	8:34:23	18:17:32
	3rd	3:47	8:12:03	17:53:36

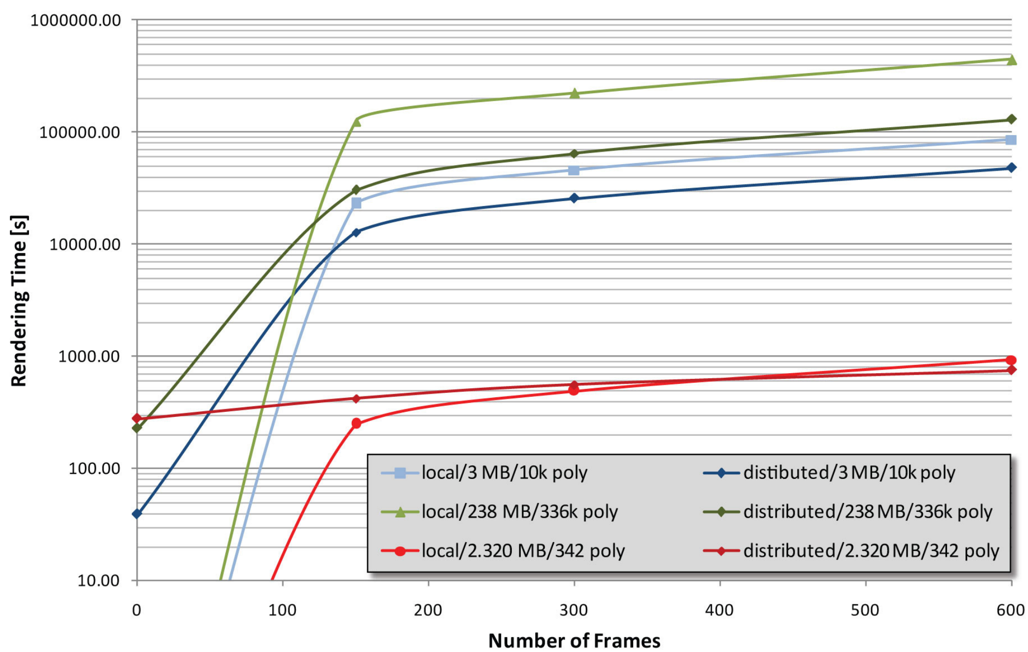

Figure 5: Distributed and local rendering times for 3D scenes of different complexity as the function of the number of frames.

4. Recommendations for Practitioners

The experimental results presented in Table II, Table **III** and Table **IV**, as well as in Figure 5 , lead to the following conclusions.

Scenes in class TI (texture-intensive), such as our first animation, *The Southern Church*, have large textures, but few polygons. In this case, times needed for data transfer before the distributed rendering process starts, as well as additional times for connecting separate frames into a video file, are of the same order as distributed rendering times. As a consequence, there is almost no speed up when distributed rendering is used instead of local rendering.

Scenes from the other two classes – PI (polygon-intensive), such as *The Horreum*, and TPI (texture-and-polygon-intensive), such as *The Villa with the Peristyle*, have less complex textures, but much higher polygon count. For animations based on scenes with these properties, distributed rendering allows speed ups which are almost linear in the number of computer systems used for rendering.

After comparing the rendering times in Table II with the times in Tables Table **III** and Table **IV**, it is also clear that using textures for representing details in 3D scenes should, whenever possible, be preferred over polygons, as it leads to much shorter rendering times. However, in the case of the distributed rendering, times needed for transferring large textures through the computer network can make a significant share of the total processing times. This influence can be lowered by using faster Gigabit *Ethernet* networks, instead of the 100 Mbit network used in the presented experiments. Therefore, when deciding whether to use local or distributed rendering, all of the previously discussed factors (size of textures, number of polygons, processing power available on different systems, and computer network speed) should be carefully taken into account.

From Figure 5, it can be seen that the sharp rise in rendering times ends after around 150 frames in all considered cases. This fast initial rise is the consequence of time needed to prepare rendering environment, which is independent of the total number of animation frames. As a consequence, the rendering setup times represent a larger share of the total processing time for shorter animations. When rendering more than 150 frames, the increase of rendering time becomes directly proportional to the increase of animation length in frames.

We can conclude that the distributed rendering is particularly recommendable in the cases when:

- Animations are created from scenes with high number of polygons and small textures. When objects on the scene use large textures, time needed for data transfer between nodes over the computer network must be taken into consideration when deciding whether to perform local or distributed rendering.

- Animations are composed of at least 150 frames. For shorter animations, time needed for the setup of the distributed rendering environment is longer than the reduction of processing time ($t_{setup} > t_{render}$). Therefore, for short animation (less than 5 or 6 seconds, depending on the frame), local rendering should be preferred over the distributed rendering.

5. Conclusions

In this paper, we presented a performance comparison of the local and distributed rendering methods, used for generation of video contents. The presented conclusions were drawn using animations with 3D reconstructions of buildings from the ancient Roman site of Mediana. For the purposes of distributed rendering, we used *Autodesk Backburner*. We showed that the intense use of textures leads to much shorter processing times than in the case when polygons are used for representing details of objects. We also demonstrated that distributed rendering is not advantageous in every setting and offered recommendations for practitioners considering the decision whether to perform rendering locally or in a distributed environment, based on factors such as size of textures, number of polygons, animation length, and computer network speed.

Acknowledgements

The research reported in this paper is partly supported by the Ministry of Education and Science of the Republic of Serbia, projects ON174026 (2011-2014) and III44006 (2011-2014).

References

[1] T. Akenine-Moller, E. Haines, and N. Hoffman, *Real-time Rendering*, 3rd edition, CRC Press, 2008.
[2] Autodesk, Inc., *3DS Max*, http://www.autodesk.com/products/autodesk-3ds-max/overview, last accessed December 15, 2013, 10:08 PM CET.
[3] Autodesk, Inc., *Autodesk Backburner*, http://usa.autodesk.com/adsk/servlet/index?id=12810296&siteID=123112, last accessed November 27, 2013, 11:20 PM CET.
[4] A. Chalmers, E. Reinhard, and T. Davis, *Practical Parallel Rendering*, CRC Press, 2002.
[5] Chaos Group, *V-Ray for 3DS Max*, http://www.chaosgroup.com/en/2/vray.html, last accessed November 27, 2013, 11:20 PM CET.
[6] J. F. Hughes, A. van Dam, M. McGuire, D. Sklar, J. D. Foley, S. K. Feiner, and K. Akeley, *Computer Graphics: Principles and Practice*, 3rd edition, Addison-Wesley, 2013.
[7] S. Molnar, M. Cox, D. Ellsworth, and H. Fuchs, *A sorting classification of parallel rendering*, IEEE Computer Graphics and Applications, 14: 4(194), 23–32.
[8] M. Pharr, G. Humphreys, *Physically Based Rendering: From Theory to Implementation*, 2nd edition, Morgan Kaufmann, 2010.
[9] G. Ward, *The hopeful future for high dynamic range imaging*, Invited Paper, Proc. 2007 Society of Information Display Symposium, Long Beach, California, USA, May, 2007.
[10] VirtualDub, available from: http://www.virtualdub.org/, last accessed December 11, 2013, 10:40 PM CET.

nikola.b.gajic@gmail.com
dusan.b.gajic@gmail.com
radomir.stankovic@gmail.com