Interactive Large Structure N-Body Gravity Simulation for Immersive Learning in Virtual Reality

*** AUTHORS LIST REMOVED ***

Abstract. In this paper, we present our implementation of an interactive Virtual Reality Application that simulates galaxy collisions and star movement. In our work, we have set up different galaxy collision simulations that the user can interact with. In each of these simulations, we allow the user to use Virtual Reality Hand Controllers to insert stars or other stellar objects of selectable mass into the scene at runtime and watch how the n-body simulation evolves. The overall goal of this project is to provide an immersive, interactive and visual experience that will teach people how large stellar objects interact based on Newton's Laws of Gravity. Our studies have shown, that our immersive and interactive approach significantly adds to one's perception of conceptualize Newton mechanics on large scale stellar bodies

Keywords: N - body simulation, galaxy collision, stellar objects, gravity, VR

1 Introduction

Real time n-body simulations[1] of big stellar objects like star clusters, galaxies, or objects that gravitationally interact on stellar magnitude such as black holes or quasars for virtual reality immersive learning[2,3,4,5] have recently become of higher interest due to technological advances and are challenging on several levels.

Understanding the behavior of stellar objects requires effort in abstraction due to their gravitational interaction with all bodies of the universe, given the range of the gravitational force, and due to the huge time scale[6] necessary to observe events of interest. Providing an immersive and interactive experience, such as an n-body simulation must be effective in regards to real time capabilities for a virtual reality experience. Since the gravitational force

adfa, p. 1, 2011.

weakens with the square-distance of two body of mass, rational approximations can be made to increase the simulation frame rate.

Observable stellar body activates can be star dynamics inside a galaxy and the resulting rotation speed around the galaxy center in decency of the distance, galaxy collisions and interactions with other stars or massive gravitational objects such as neutron stars or black holes. In order to understand and experience the trajectory of the mass bodies, observations in real time take usually up to several millions of years, so frame-steps must be in respect to a reasonable time scale. The rotation speed of stars around galaxy centers became of high interest since it was found, that this speed is close to constant and appears independent of the distance to the center[7]. This is an indication of more mass being present in the galaxy than can be observed and is labeled under dark matter[8] due to its invisibility for conventional methods. Providing a slider for increasing the dark matter inside the galaxy body of the simulation can be a significant tool for understanding until the rotation curve (along the radius) matches the observed data. Recent simulations of galaxy collisions of two spiral galaxies lead to a qualitative explanations of known galaxy shapes[9], to be a result of such a collision. Given two spiral galaxies with 100 million stars in average, the task for the simulation is to provide results for each frame in real time. Stellar bodies of big mass like supermassive black holes have a strong impact on the dynamics of nearby galaxies[10] and letting a user experience to be able adding massive bodies into the simulation at any time and place increases the immersion and the learning outcome in understanding movements of big stellar objects.

2 Setup

For our approach, we are using Unity as platform, GPU compute shader for the simulation calculations, and Oculus Rift for the VR experience. Our test run on an Intel i7 with 16GB and an NVidia GTX 1080 Ti graphic card.

3 System

3.1 Methodology

Our task was to simulate interactively single galaxies and its stars movement over time, galaxy collisions and adding/removing stellar objects with a high range selectable mass.

Our galaxy, the Milky Way, has in average number *n* = 10^{11} to 10^{12} million stars and is a spiral galaxy. Astronomy assumes that to be a typical representation of galaxies in the universe[11,12]. For galaxy collisions of two galaxies of approximately the same body, we have a lower boundary of *n* = 2×10^{11} stars.

Gravitational simulations must take all bodies of mass under consideration and consequently, without any approximations, all stars have to calculate the gravitational force to all other stars, resulting in an $O(n^2)$ algorithm. Since the gravitational force is decreasing with the square of the distance

$$F = G\,\frac{m_1 m_2}{r^2}$$

and not linear, we cannot sum up all masses and locations to a virtual average masspoint; *F* being the gravitational force, *G* the gravitational constant, $m_1 m_2$ the masses of the two interacting bodies and r the distance between them. Having $O(n^2)$ on 2×10^{11} stars leaves us with 4×10^{22} operations per frame and is far too much even for our system with 3840 shader units.

As discussed, the gravitational force is non-linear and consequently one cannot make cluster of mass points representing all stars of certain parts of a galaxy to reduce computational time, but the difference of calculating the gravitational force between a star and (a) a mass point or (b) all the stars represented by this masspoint is direct proportional correlated with the area of this cluster and the distance. The more distance between the star and the masspoint, the lesser the resulting error. Taking this under consideration, one would benefit greatly if the designated area of the event gets rasterized in three dimensions with several layers of resolution. Barnes and Hut[13] suggested a hierarchical approach by constructing an octree of a desired maximal resolution. Due to the computational time taken by constructing an octree on the GPU memory every frame as described by Laine and Tero[14], we decided to use buffers representing the 3D area and do the downsampling from higher resolution to lower in our code manually.

3.2 Algorithm

Every star is represented by mass (float), position (3 x float), impulse (3 x float) and an integer for having a linear list later to connect stars in one raster area, having 7 floats and one integer, 32 bytes in sum on a 32 bit system.

We have 7 raster buffers, see figure 1, where the highest resolution is 256^3 and every next buffer contains half the resolution of its predecessor: 128^3, 64^3, 32^3, 16^3, 8^3, 4^3. One buffer element contains the average mass and average position of the masspoint. Having 16 bytes for each masspoint, we need 33.5 MB for the highest resolution buffer, 4.2 MB for the second highest, and in sum 39 MB on the GPU memory for the whole hierarchical raster. To store a list of stars for later faster access of the local stars, the highest resolution buffer has 2 more integers, to hold the index of the actual star of the list as well as to the first star entering the list, needing in additional 17 MB for the first buffer.

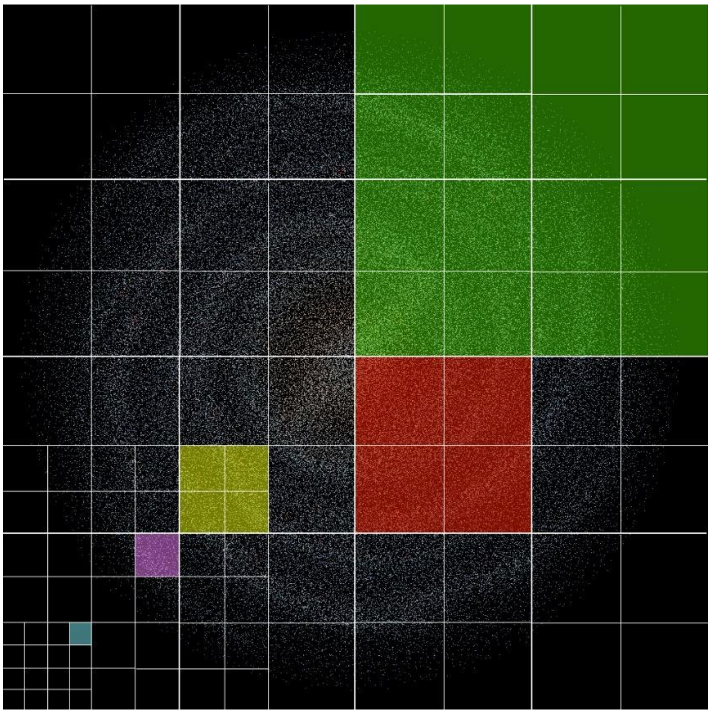

Fig. 1. We use a hierarchical raster buffer to store the average mass and position of the cluster of stars inside the raster elements on seven different levels of resolution, beginning with 256^3 elements, then 128^3 (cyan tile), 64^3 (purple tile), 32^3 (yellow tile), 16^3 (red tile), 8^3 (green tile), and 4^3 elements.

Figure 1 shows our approach on an illustration of the Milky Way. The region gets subdivided into smaller areas.

First we loop over all stars and map their position into the highest resolution raster, adding their mass and position to the average variables of the raster buffer. If the star entering its values to the element of the buffer appears to be the first one, its writing its index to the “first star” and the “actual star” integer. If the “first star” integer already has an entry, its writing its own index to the star “next star” integer of the “actual star” and then overwrites it with its own index.

Then we downsample 8 elements of the buffer to its next lower resolution raster buffer element. Repeating that process for all raster buffer, we obtain masspoints in all different resolutions for the given space.

All our calculations take place on the GPU, having 3840 shader units working parallel on the algorithm.

To calculate the force on each star, we loop through all stars *k* in the local area of the highest resolution buffer, using the linked list we established. Then, we take all first resolution elements around the star (3 x 3 x 3 minus its own element) and repeat that process with the remaining 6 buffers, 3 x 3 x 3 - 1 operations for each, resulting in a sum of (3 x 3 x 3 - 1) x 7 + k which is in comparison with all stars of the given scenery a huge improvement.

If a star is too close to the border of the highest resolution rasterbuffer, it would only interact with the next masspoint, not with a possible close star. Such error can be corrected by increasing the amount of rasterbuffers to cover all border areas, but is subject to future implementation.

Still, 10^{11} stars are a huge computational problem, not only for computational time but also for the GPU memory. Even known dwarf galaxies such as NGC 5457 have a star amount of approximately 100 million stars, hence being problematic in real time as well.

We decided to reduce our galaxies to a maximum of one million stars, letting our stars represent small clusters of stars and believe this approach to still resemble real interaction of galaxies. With *n* = 10^6, we need 32 MB for the star buffer and have in average a local group of stars of *k* = 3 for one galaxy filling the entire simulation area.

3.3 Constructing the Galaxy

To construct a galaxy of stars, they need to be distributed according to observation (cluster galaxies, spiral galaxies, etc) and rotating around its center to satisfy

$$v_0 = \sqrt{G\frac{m_1 m_2}{r}}$$

which can be calculated once the positions are set, and v0 being the initial velocity in spiral direction, r the distance, G the gravitational constant and m1m2 the masses of the two stars. To have a realistic distribution, we developed a program to read from an image file and calculate possible star locations. Having a galaxy image see figure 2, we assume brighter areas to have a dense star population. Our algorithm takes the pixel brightness as well as the overall brightness of its surroundings into consideration to determine density, but also the thickness of the galaxy, see results in figure 2.

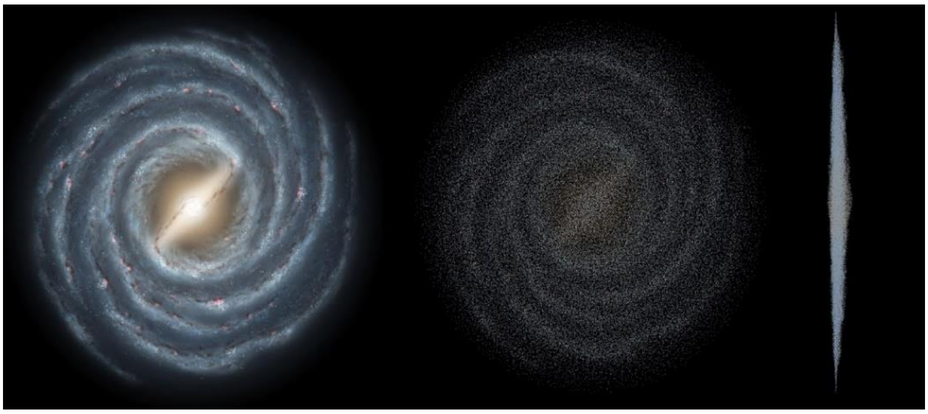

Fig. 2. On the left side an illustration of NGC 6814, scanned and after stars placed into position, the rendered 3D image of top down view is next to the right, followed by a side view of the galaxy.

4 Simulation Results

Our approach runs in real time on an Oculus Rift. At program start, the user can choose between several scenarios involving different types of galaxies and collisions, see figure 3: Single galaxies or pairs of such for collision, galaxies which resembles real ones or hypothetical ones to increase the user experience such as pure spirals or rectangular shapes of star clusters.

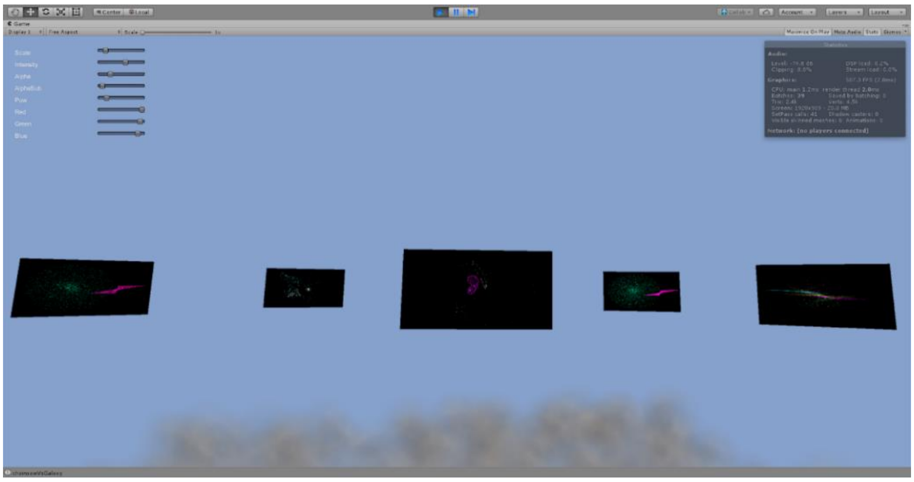

Fig. 3. Start screen of our simulation in unity. The user is in VR, seeing a menu of 5 different scenes he can choose from, from left to right: Galaxy collision of 2 conceptual galaxies, single real galaxy, single concept galaxy, collision of two real galaxies and collision of a conceptual and a real galaxy.

Inside the scene, the simulation is paused by default and the user can start/stop the simulation any time or navigating through the scene. Additional stars can be inserted and the mass of this new stars can be selected up to the mass of supermassive black holes, which are represented by a dot to leave them visible for the user.

The gravitational constant can the increased as well, to observe its effect on the rotation speed of stars, until the galaxy rotation speed curve matches real observation, concluding the mass percentage missing inside the galaxy filled with dark matter.

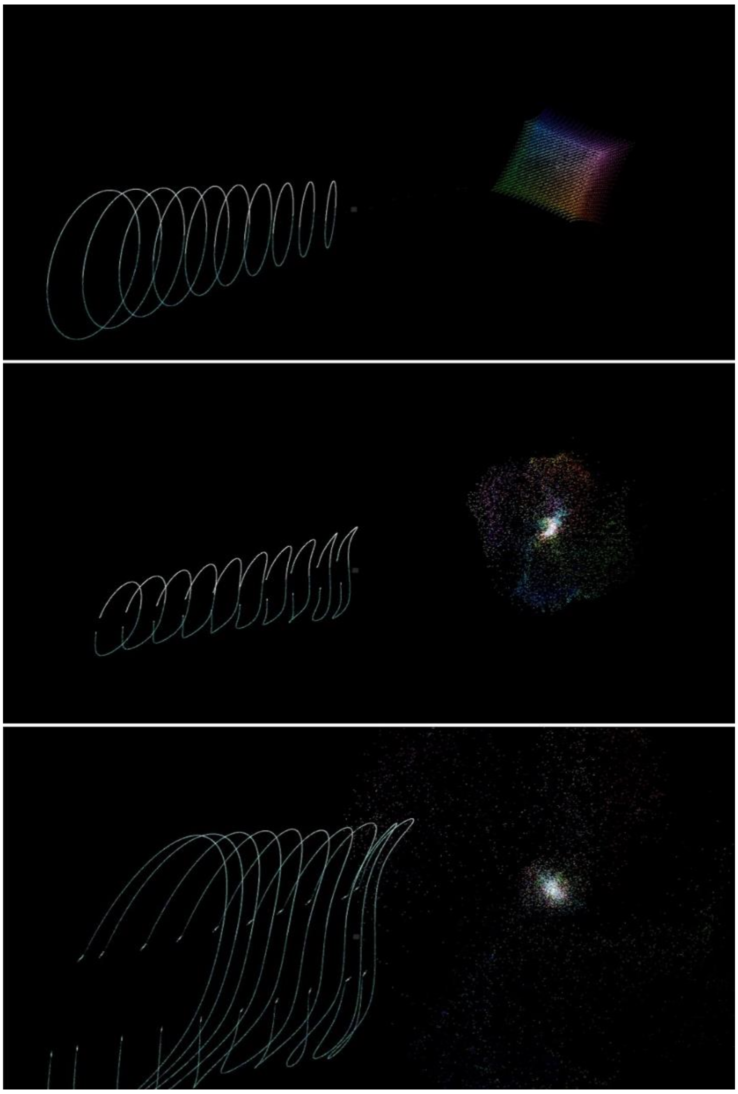

Fig. 4. Evolution of the simulation with 2 conceptual galaxies for better visualization in a two dimensional image, from top to bottom: In the first image, we see two conceptual galaxies, one being shaped as cube, the other one as several rings with different density and masses periodically over each ring. In the next image, we can see the uneven distribution of the rings start to take effect by splitting them up, while the cubed shaped galaxy implodes. Further on, in the third image, the rings bend towards the mass of the cube galaxy while the stars of the cubed one, which started to pass the center at the second image which high speed, are still expanding, though at the center a cluster of stars concentrate.

5 Learning Experience

This section is already written, but as long as we do not have ethics approval, we are not allowed to discuss the data. We hope to be able to include the text in the next 10 days (as from 3/1/2018).

6 Conclusion future work

In our work, we have shown that qualitative, immersive and interactive real time gravitational force simulation on large stellar structures in virtual reality are possible under several approximations and abstractions on parallel algorithms for compute shader GPU. We provided a tool for constructing a galaxy from an image file and tested subjects on the impact and learning outcome when interacting with our simulation, concluding that such immersive virtual reality experience has significant impact due to the nature of this field of study, being mostly taught in a static and abstract manner, given the time scale and complexity of this stellar events. There is room for further improvement to make the simulation more precise and effort will be made to include more stars per galaxy to match real galaxies in future. The next step is to implementing the Lorentz transformation in dependency of the speed of the user while traversing the simulation, in order to warp the perception of space according to relativity.

7 Acknowledgements

The equation to compute star movement and the text files with predefined galaxies were received from:
http://courses.cs.tamu.edu/sueda/CSCE441/2017S/labs/L17/

The following link was used to learn how to render particles in Unity using the compute shader: https://github.com/antoinefournier/XParticle

The following links were used to learn how compute shaders in Unity work: http://www.emersonshaffer.com/blog/2016/5/11/unity3d-compute-shader-introduction-tutorial, http://kylehalladay.com/blog/tutorial/2014/06/27/Compute-Shaders-Are-Nifty.html

The camera system was modeled in reference to the following link: https://learnopengl.com/Getting-started/Camera

8 References

1. Portegies Zwart, Simon F.; Belleman, Robert G.; Geldof, Peter M.: “High-performance direct gravitational N-body simulations on graphics processing units” ELSEVIER, 11/2007

2. Learning science in virtual reality multimedia environments: Role of methods and media. By Moreno, Roxana,Mayer, Richard E. Journal of Educational Psychology, Vol 94(3), Sep 2002, 598-610

3. Bailenson, J., Yee, N., Blascovich, J., Beall, A., Lundblad, N., & Jin, M.: “The use of immersive virtual reality in the learning sciences: digital transformations of teachers, students, and social context”, The Journal of the Learning Sciences, 17, 102-141, 2008.

4. R. L. Jackson, E. Fagan: “Collaboration and learning within immersive virtual reality” CVE '00, pages 83-92

5. Z. Merchant, E. T.Goetz, L. Cifuentes, W. Keeney-Kennicutt, T. J.Davis: “Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis”, Computers & Education, vol 70, 2014

6. N. C. Hearn, S. A. Lamb: “A High-Speed Galaxy Collision With Episodic Star Formation”, Arp 119,Volume 551, 2001

7. E. Corbelli, P. Salucci: "The extended rotation curve and the dark matter halo of M33". Monthly Notices of the Royal Astronomical Society. 311 (2), 2000

8. Hammond, Richard: “The Unknown Universe: The Origin of the Universe, Quantum Gravity, Wormholes, and Other Things Science Still Can't Explain.”, Franklin Lakes, NJ: Career Press, May 1, 2008

9. Volker Springel[1] and Lars Hernquist, “Formation of a Spiral Galaxy in a Major Merger”, The Astrophysical Journal Letter, 2005

10. Schödel, R.:"A star in a 15.2-year orbit around the supermassive black hole at the centre of the Milky Way". Nature. 419 (6908): 694–696 et al. (2002).

11. Hubble, E.P. (1936). The realm of the nebulae. Mrs. Hepsa Ely Silliman memorial lectures, 25. New Haven: Yale University Press.

12. "Hubble and Galaxy Zoo Find Bars and Baby Galaxies Don't Mix". Science Daily. 16 January 2014.

13. J. Barnes and P. Hut: “A hierarchical O(N log N) force-calculation algorithm”. Nature, 324(4):446-449, December 1986

14. Laine, Samuli; Tero Karras: "Efficient Sparse Voxel Octrees – Analysis, Extensions, and Implementation", IEEE Transactions on Visualization and Computer Graphics 17 (8), 1048-1059

15. Schombert, J. M.; Pildis, R. A.; Eder, J. A.; Oelmer, A., Jr. (1995). "Dwarf Spirals". The Astronomical Journal. 110: 2067–2074.