High Peformance Algorithms for Real-Time GPGPU Volumetric Cloud Rendering from a Parallel Programming Approach

Carlos Jiménez de Parga[1], Sebastián Rubén Gómez Palomo[2]

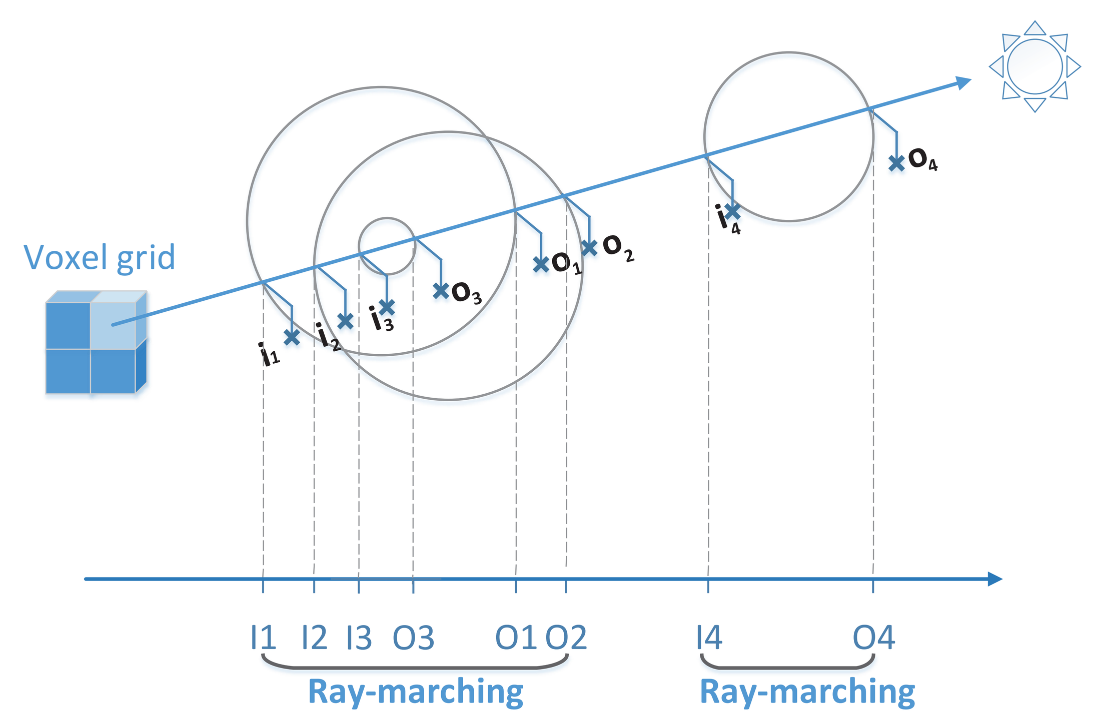
[1]National Distance Education University, UNED, Cartagena.

[2] Department of Software Engineerig and Computer Systems, UNED, Madrid.

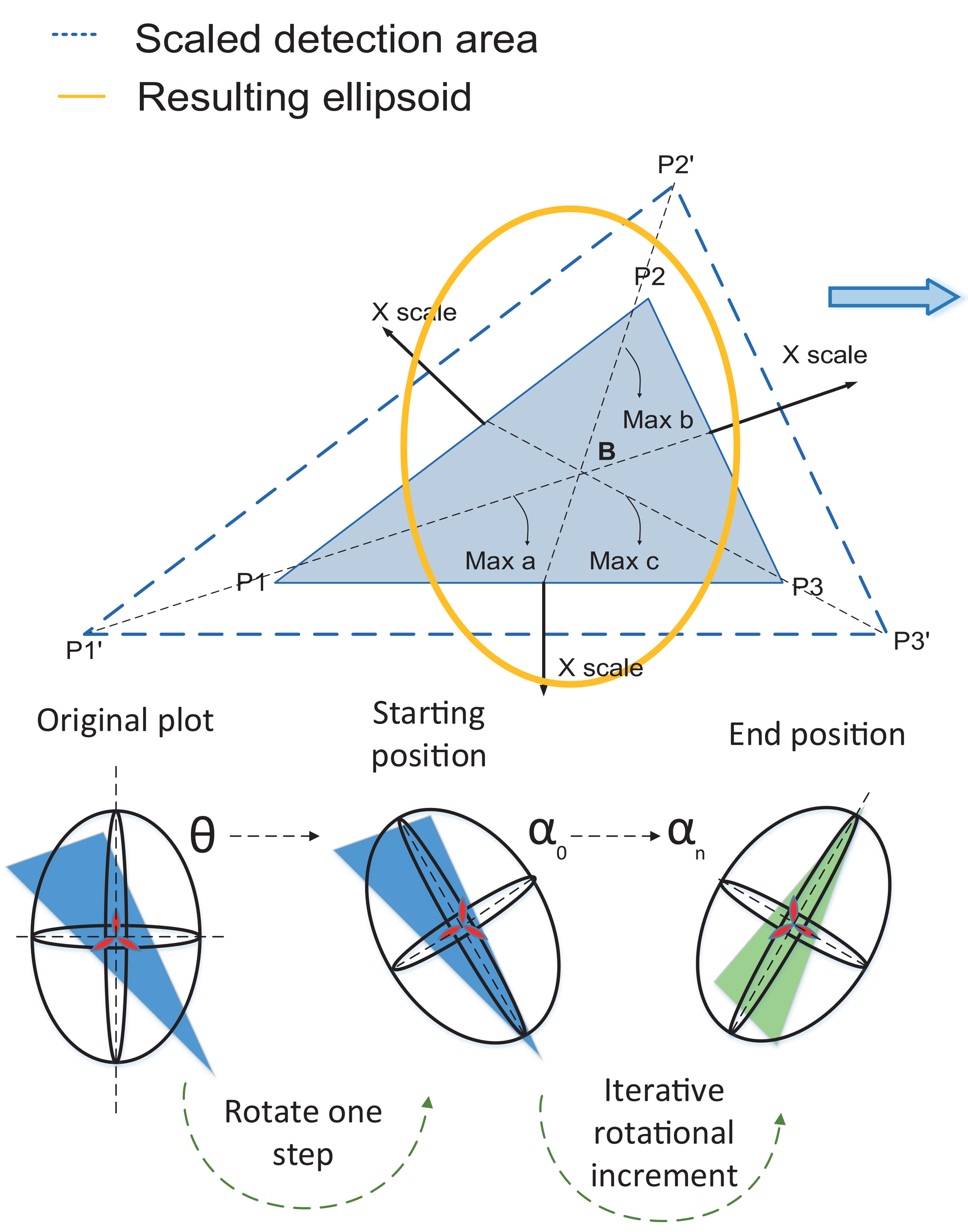
ABSTRACT AND OBJECTIVES

Real-time rendering of realistic clouds is an essential feature in landscape generation for flight simulators, computer games and virtual reality environments. However, the computation cost of ray-marching in volumetric scenarios prevents achieving the desired performance without dedicated ray-tracing graphics cards. The objective of this research is providing cutting edge software based on efficient algorithms and parallel computing through Compute Unified Device Architecture (CUDA) to achieve an optimum balance between realism and performance. Starting from a cloud primitive, this method uses enhanced mathematics and physical abstract models for real-time cloud rendering with improved geometry. The empiric benchmarks and render quality results indicate that our methods stick to the initial hypothesis and they are suitable for use in the entry level graphics industry.

MATERIALS AND METHODS



The No-Duplicate-Tracing algorithm (NDT) speeds-up light pre-computation X2



Pre-processing

$$B = \left(\sum_{i=1}^{3}\frac{x_i}{3}, \sum_{i=1}^{3}\frac{y_i}{3}, \sum_{i=1}^{3}\frac{z_i}{3}\right)$$

$$P_i' = (P_i - B)\cdot scale + B$$

$$radius_a = \|(B - P_1')\|$$
$$radius_b = \|(B - P_2')\|$$
$$radius_c = \|(B - P_3')\|$$

$$if \begin{cases} max(radius_a) & dir\vec{T}ria = P_1' - B \\ max(radius_b) & dir\vec{T}ria = P_2' - B \\ max(radius_c) & dir\vec{T}ria = P_3' - B \end{cases}$$

$$if \begin{cases} max(radius_a) & dir\vec{E}llip = (radius_a + B_x, B_y, B_z) - B \\ max(radius_b) & dir\vec{E}llip = (B_x, radius_b + B_y, B_z) - B \\ max(radius_c) & dir\vec{E}llip = (B_x, B_y, radius_c + B_z) - B \end{cases}$$

Rodrigues' Rotation Formula

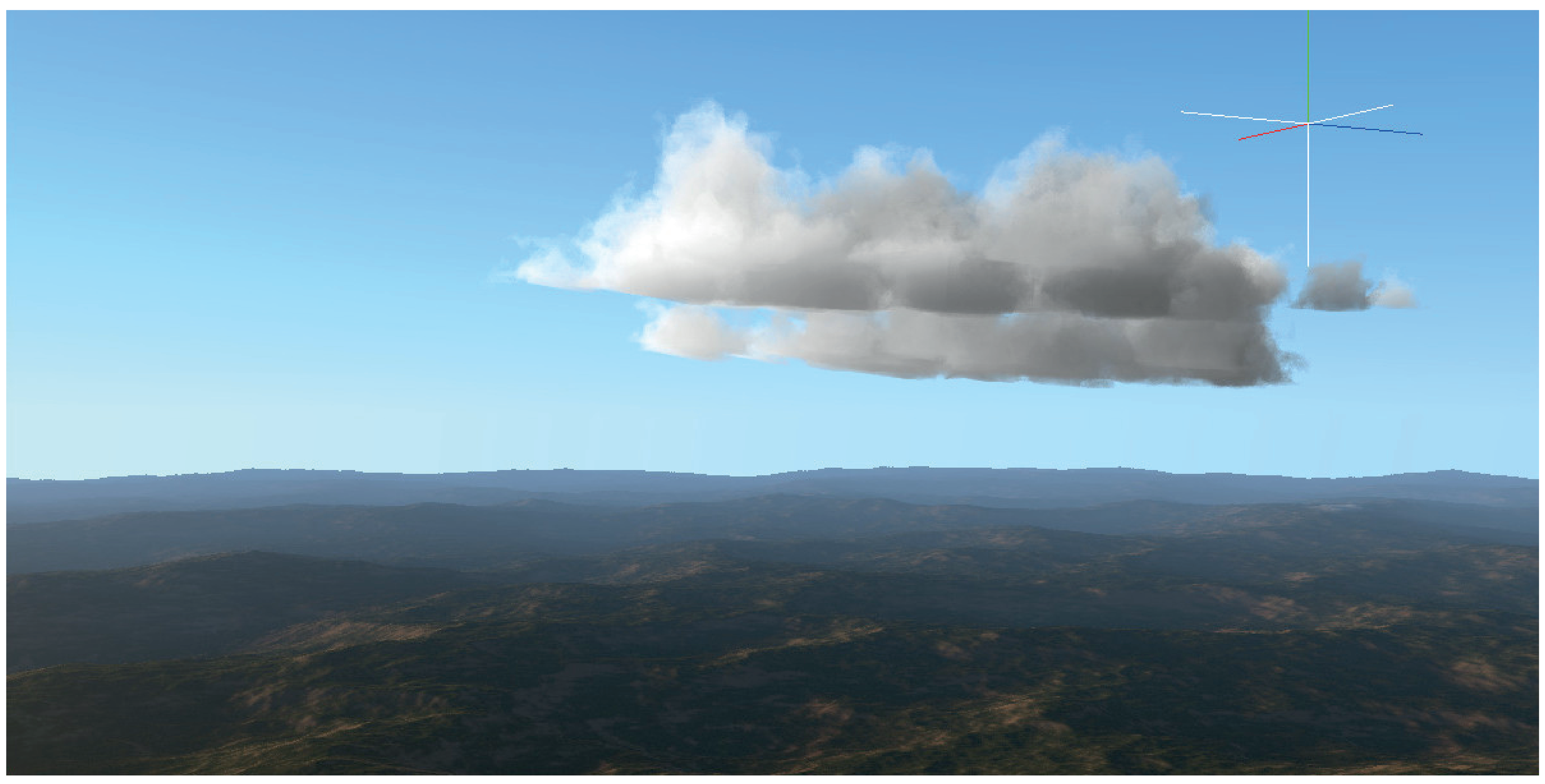
$$x = \frac{dir\vec{E}llip \times dir\vec{T}ria}{\|dir\vec{E}llip \times dir\vec{T}ria\|}$$

$$\theta = cos^{-1}\left(\frac{dir\vec{E}llip \cdot dir\vec{T}ria}{\|dir\vec{E}llip\| \cdot \|dir\vec{T}ria\|}\right)$$

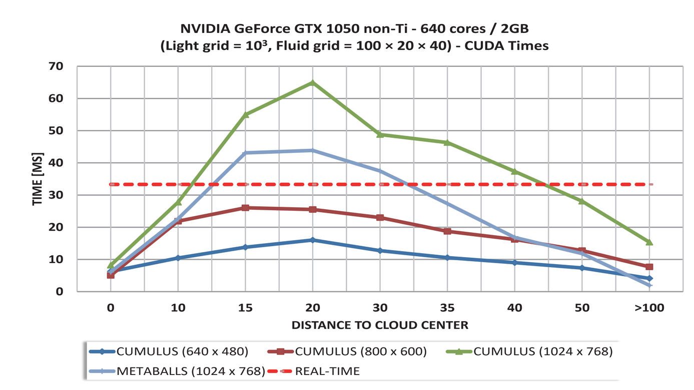
$$R = e^{A\theta} = I + sin(\theta)\cdot A + (1 - cos(\theta))\cdot A^2$$

$$A = [x]_x = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

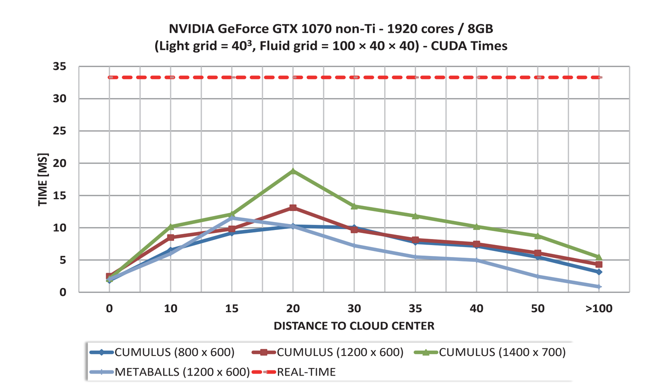
QUALITY OF RENDERING



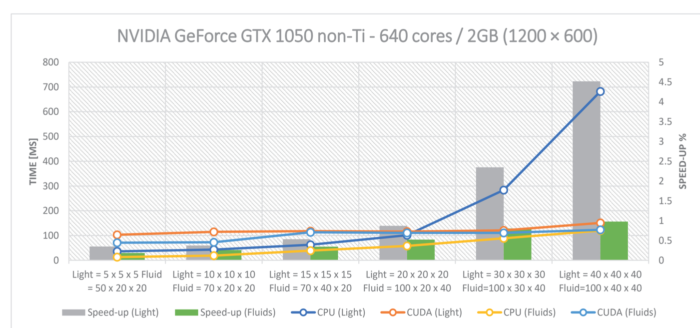
RESULTS

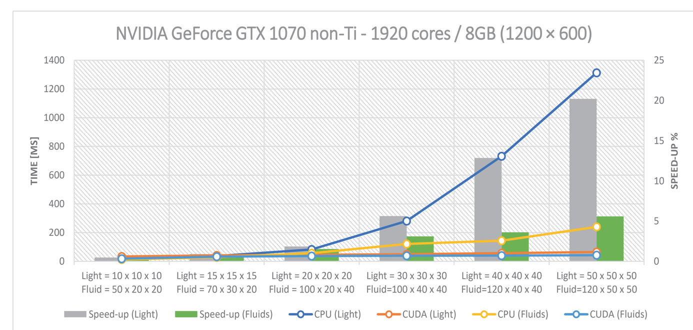


77.7 % of samples are above 30 FPS



100 % of samples are above 30 FPS





CONCLUSIONS

As a conclusion, the overall performance tested in different low-mid-level graphics cards and the image quality results demonstrate real-time rendering with likely real clouds of a quality similar to other particle and slow hyperrealistic implementations. Hence, our algorithms are suitable for application in the standard graphics industry [1][2].

REFERENCES

[1] C. Jiménez de Parga and S. Gómez Palomo. "Efficient Algorithms for Real-Time GPU Volumetric Cloud Rendering with Enhanced Geometry". In: Symmetry, Volume 10, Issue 4 (2018), p. 125.

[2] C. Jiménez de Parga and S. Gómez Palomo. "Parallel Algorithms for Real-Time GPGPU Volumetric Cloud Dynamics and Morphing". In: Journal of Applied Computer Science and Mathematics, Volume 13, Issue 1 (2019), pp. 25-30.