Visualization of Debris Disks Morphologies in Scattered Light

Robin Leman,[1,2,*] Supervisors: Eve J. Lee,[1,3,4,†] and Clark Verbrugge[2,‡]

[1] *Department of Physics, McGill University, Montréal, Québec, Canada*
[2] *School of Computer Science, McGill University, Montréal, Québec, Canada*
[3] *McGill Space Institute, McGill University, Montréal, Québec, Canada*
[4] *Institute for Research on Exoplanets, Montréal, Québec, Canada*

(Dated: February 28, 2022)

This paper builds on the work of Lee & Chiang (2016) [1]to visualize and characterize the morphologies of debris disk in solar systems. By using 3-dimensional computer simulation, we were able to accurately render debris disk dust particles in scattered light. The minimal model simulated is a "narrow ring of parent bodies, secularly forced by a single planet on a possibly eccentric orbit, colliding to produce dust grains that are perturbed by stellar radiation pressure" (Lee & Chiang (2016) [1]). Our model allows for a more practical simulation by leveraging 3-dimensional real time rendering. The second part of the research will focus at expanding the visualization to various wavelengths.

I. INTRODUCTION

The formation of planets in a solar system, according to the core accretion model (Maoz (2016) [2]), is the result of parent body collisions in a protoplanetary disk. This collisions launch remnants; dust particles of various sizes orbitting the system. When the gas of the accretion disk has been dispersed, at around 10Myr old, these dust particles form a debris disk surrounding the solar system.

The dust particles of this debris disk are gravitationally perturbed by the planets of the solar system, directly affecting their orbits (Cataldi (2016) [3]). This results in various debris disk shapes, characterized by the planets orbits, the viewing angle of the disk, and the observed wavelength. Analyzing the morphology of a debris disk can thus give us critical information on the presence of exoplanets in the system.

In previous papers, different morphologies have been discovered; "moths", "needles", "ship warps" or "eccentric ring" (Lee & Chiang (2016) [1]). It has also been shown that a single disk could produce different morphologies depending on the viewing angle.

In this paper, we are using 3-dimensional rendering techniques to visualize the morphologies of a debris disk orbiting a solar system composed of a single planet, with a possibly eccentric orbit. Building up on the work of Lee & Chiang (2016) [1], we suggest a more performant and practical tool to analyze the various shapes of a particular debris disk at different viewing angles.

This interim report focuses on the first part of the project; the visualization of debris disk morphologies in scattered light. The second part of the project will explore debris disk visualization at various wavelength. Observing debris disk at different wavelengths can notably probe dust populations at different distances from the star, and at different sizes (Cataldi (2016) [3]).

* robin.leman@mail.mcgill.ca
† evelee@physics.mcgill.ca
‡ clump@cs.mcgill.ca

II. MODEL

A. Resolving dust orbits in Keplerian coordinates

We compute the orbits of the dust particles forming the debris disk using the same methods as the work of Lee & Chiang (2016) [1]. We instantiate $N_p = 1000$ parent body orbits in a uniformly distributed narrow ring located just outside the planet's elliptical orbit and apsidally aligned with it. From the parent body orbits, we uniformly distribute $N_{launch} = 100$ true anomalies. For each true anomaly, we create a dust particle orbit with Keplerian parameters

$$a_d = \frac{a_p(1-e_p^2)(1-\beta)}{1-e^2-2\beta(1+e_p\cos f_p)} \tag{1}$$

$$e_d = \frac{\sqrt{e_p^2+2\beta e_p\cos f_p+\beta^2}}{1-\beta} \tag{2}$$

$$\omega_d = \omega_p + \arctan\left(\frac{\beta\sin f_p}{e_p+\beta\cos f_p}\right) \tag{3}$$

$$i_d = i_p \tag{4}$$

$$\Omega_d = \Omega_p \tag{5}$$

where the p and d subscripts respectively represent the parent and dust orbital parameters, with a the semi-major axis, e the eccentricity, ω the argument of periapse, Ω the longitude of the ascending node, and f_p the launch true anomaly. For a more performant simulation, we pre-compute parent orbits and store them in a text file.

B. Conversion from Keplerian orbits to Cartesian particles

From the dust particles orbits, we draw $N_d = 1000$ dust particules with uniformally distributed mean anomaly M_d. We solve

$$M_d = E_d - e_d \sin E_d \tag{6}$$

for the eccentric anomaly E_d using Newton's method, stopping the iteration when the new iteration diverges from the previous by less than a threshold $\delta = 0.0001$. This allows us to compute the true anomaly of the f_d as

$$\tan\frac{f_d}{2} = \sqrt{\frac{(1+e_d)}{(1-e)}}\tan\frac{E_d}{2} \tag{7}$$

and the orbit radius r_d

$$r_d = a_d\Big(\frac{1-e_d^2}{1+e_d\cos f_d}\Big). \tag{8}$$

Finally, we compute the particles Cartesian coordinates;

$$x_d = r_d * \cos\Omega_d\cos(\omega_d + f_d) - \sin\Omega_d\sin(\omega_d + f_d)\cos i_d$$
$$y_d = r_d * \sin\Omega_d\cos(\omega_d + f_d) - \cos\Omega_d\sin(\omega_d + f_d)\cos i_d$$
$$z_d = \sin(\omega_d + f_d)\cos i_d$$

from Schwartz (2017) [4] and Murray & Dermott (1999) [5]. Since the dust particles are static, their positions are pre-computed on the CPU.

C. Particle rendering in scattered light

We project every dust particle onto a 1080x1080 pixel grid from the point of view of a camera constrained to a spherical plane of radius d, azimuthal angle ϕ and polar angle θ, with the origin defined at the star of the solar system. Those parameters allow us to simulate the position of the dust particles from the point of view of an observer looking at the star of the solar system.

Every dust particle in a same pixel contribute the same amount of 10% alpha blending (Eck (2021) [6]). We compute the intensity I_d of every dust particles as

$$I_d = \tilde{\beta}_{HG}(g, \cos\theta_d)\frac{1}{\beta_d^2\rho_d^2} \tag{9}$$

where $\tilde{\beta}_{HG}(g, \cos\theta_d)$ is the Henyey-Greenstein phase function (Henyey & Greenstein (1941) [7]) with $g = 0.5$, ρ_d is the dust particle's distance from the observer, and θ_d is the angle between the star, the particle position, and the observer.

The intensity of the particle I_d is then mapped to a 1-dimensional texture and drawn as such, see Appendix A.

We render the dust particles in a real-time environment, taking user input for the position of the observer.

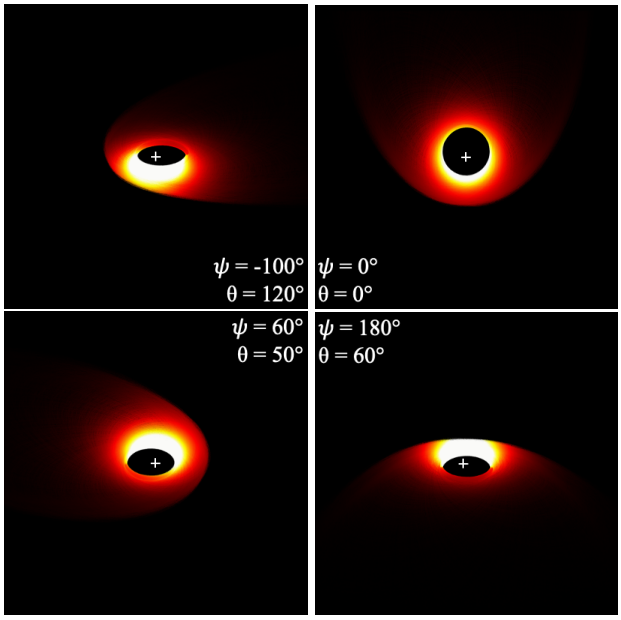

FIG. 1. Rendering of a debris disk with single collision simulated, for a planet of eccentricity $e = 0.25$ and fixed radiation $\beta = 0.35$. The luminosity of the debris disk has been exaggerated to show all the features of the disk. The disk is rendered at different angles ϕ and θ. From this rendering, we can characterize the top right panel as a "ring" and the bottom right panel as a "moth"

We compute the scattering function and draw the particles at every frame, with a simulation running at an adaptive frame rate; a new frame starts as soon as the previous one has been rendered.

The rendering pipeline (OpenGL Wiki [8]) is as follows. Firstly, the vertex stage iterates over every particle and projects them onto a 1080x1080 pixel grid. Then, it computes the intensity I_d of the particle. Secondly, the fragment stage maps the intensity I_d to a colormap texture, and performs alpha blending to add the contribution of every particle to the pixel. The rendering pipeline is performed on the GPU.

III. RESULTS

As a preliminary result, we simulate the debris disk of a system with a singular planet of eccentricity $e = 0.25$, and fixed radiation $\beta = 0.35$. We render the disk at different angles in scattered light.

We present the scattered light render in Figure 1. Already from the scatter light image we can identify two morphology types depending on the viewing angle of the disk; a moth and a ring.

Loading 10,000 parent orbits from a pre-computed text file takes a total time of 0.0535s. Converting the parent orbits to dust particles takes an average of 321.39μs repeated for a total of 10^7 dust particles, so a total time of 3.2139s (Figure 2). When the rendering loop begins, ev-

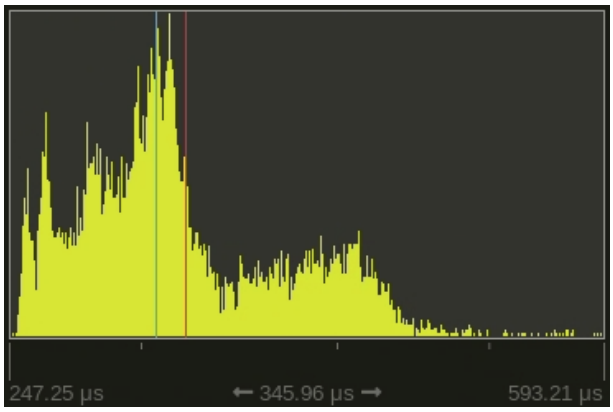

FIG. 2. Time performance histogram for each conversion from parental orbits to dust particles. In red, the mean time 321.39 μs, in blue, the median time 307.04μs.

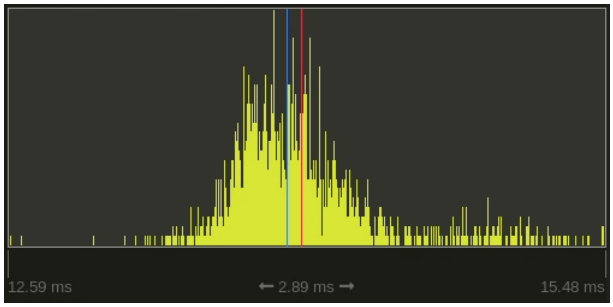

FIG. 3. Time performance histogram for each rendering frame. In red, the mean frame time 13.94ms, in blue the median frame time 13.87ms

ery frame takes an average of 13.94ms, or 72FPS (Frame Per Second) (Figure 3). See Appendix B for hardware specifications.

We also present a comparative analysis of a debris disk, with planet eccentricity $e = 0.1$ and fixed $\beta = 0.4$, at different resolutions (Figure 4). This debris disk is successful at demonstrating the ring morphology for a small planet eccentricity. An increasing resolution allows for a better demonstration of the scattering function, and better probing of dust population density. However, the average frame time for 10^8 render particles is now 484.06ms, or 2FPS, with a median at 132.13ms, or 8FPS.

IV. DISCUSSION

Overall, our model is successful at providing a more preformant and practical debris disk simulation in scattered light. While our results are similar to the work of Lee & Chiang (2016) [1], it allows for a better resolution and more flexibility in the design. This flexibility will greatly help us in adapting the model to different wavelengths in the second part of the research.

However, our model produces inconclusive results when using a β distribution instead of a fixed β. This

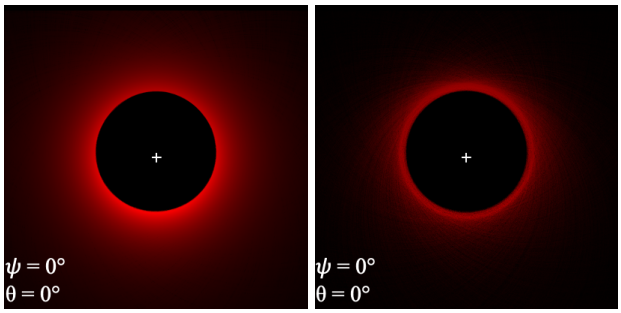

FIG. 4. Scattered light render of a debris disk, with planet eccentricity $e = 0.1$ and fixed $\beta = 0.4$. The left panel has 10^8 particle rendered, the right panel has 10^7.

could be due to a miscalculation or a to the lack of smoothed image output. This will be worked on in the second part of the research.

A deeper analysis of our model will have to be performed in the next part of the research. First, we will need to analyze more disks to accurately relate solar system properties to disk morphologies. Then, we will need to compare our model to real data to assess the accuracy of the simulation.

Our rendering time per frame for 10^7 is excellent, with an average of 72FPS well above the 30FPS standard. This allows for a smooth simulation and quick iteration time. However, we hit a bottleneck at 10^8 particle simulated, with an average of 2FPS. While the simulation is still maniable, it is the limit that it can handle. To simulate more particles, a better hardware will have to be used, or an offline render to be performed.

Another bottleneck is the conversion from keplerian orbits to particles in cartesian coordinates. While the performances of a single orbital conversion is well optimized, at an average of 321.39μs, the need to repeat the conversion for all 10,000 orbits hurts the overall performances. To improve that, we can use the fact that every orbits are independent to spread the work on multiple CPU threads. This optimization will be explored in the next part of the research.

V. CONCLUSION

To conclude, our model is successful at providing accurate visualization of debris disks in scattered light. The performances allow for real time rendering at up to 10^7 particle simulated. In the next part of the research, we will expand the simulation to multiple wavelengths. We will perform deeper analysis to classify the debris disks in terms of morphologies, relating them to the properties of the observed solar system. Finally, we will improve the performances of the simulation with multithreading.

REFERENCES

[1] E. J. Lee and E. Chiang, A primer on unifying debris disk morphologies, The Astrophysical Journal **827**, 11 pp (August 2016).
[2] D. Maoz, *Astrophysics in a Nutshell*, 2nd ed. (Princeton University Press, ISBN 978-0-691-16479-3, 2016).
[3] G. Cataldi, *Debris disks and the search for life in the universe*, 1st ed. (Department of Astronomy, Stockholm University, ISBN 978-91-7649-366-3, 2016).
[4] M. R. Schwarz, Keplerian orbit elements → cartesian state vectors.
[5] C. D. Murray and S. F. Dermott, *Solar System Dynamics*, 1st ed. (Cambridge University Press, ISBN 0-521-57597-4, 1999).
[6] D. J. Eck, Introduction to computer graphics (2021).
[7] L. G. Henyey and J. L. Greenstein, Diffuse radiation in the galaxy, The Astrophysical Journal **93**, 70 (1941).
[8] O. Wiki, Rendering pipeline overview (2021).

Appendix A: Colormap

1-dimensional texture used to map the intensity of a dust particle to a color. For the scattered light, the colormap is taken from Matplotlib's hot texture, see Figure 5

FIG. 5. Matlab "hot" colormap as 1-dimensional texture. From kbinani.

Appendix B: Hardware Specifications

All measurements were made on the same hardware, with specifications detailed in Table I

Processor	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz
RAM	12.0GB
System Type	64-bit Operating System
GPU	Nvidia Quadro P5000

TABLE I. Hardware specifications.