Gravispy: Gravitational Simulations in Open Source Software*

Spence Norwood, Kellen O'Keefe, and Calvin Ross

University of Texas at Dallas, Physics Department

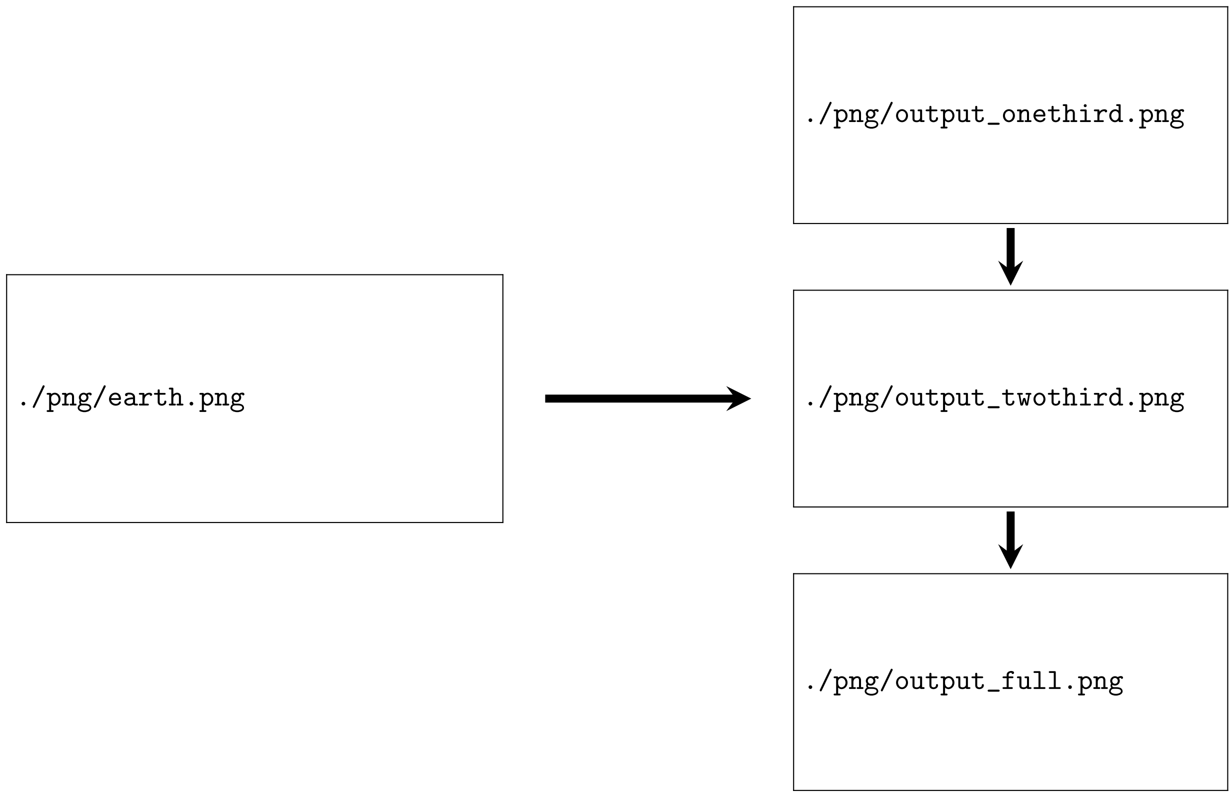
Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur venenatis ultricies purus, vel pellentesque tortor pellentesque vulputate. Sed a tincidunt felis. Integer ultrices, dui in maximus ultrices, leo erat aliquet diam, ut interdum sapien turpis eget lorem. Phasellus lectus augue, vehicula vel faucibus sed, dapibus in dolor. Vivamus interdum diam vitae fermentum mollis. Donec tortor sem, mollis ut tincidunt ut, iaculis id quam. In nunc lorem, mattis id hendrerit eget, accumsan vitae odio. Morbi ipsum massa, volutpat eget magna hendrerit, pellentesque pellentesque augue. Fusce a convallis massa, non mattis nisl. Mauris ac luctus sem. Aenean dui eros, blandit in nisl et, porta malesuada diam. Sed eget turpis eget erat vehicula condimentum. Donec eu tellus ultricies, luctus massa non, convallis lacus. Nunc fringilla porttitor sem nec luctus. Quisque tincidunt dui id finibus facilisis. Proin in consectetur risus, ut ornare ipsum.

Phasellus gravida leo nec quam sagittis, non semper quam molestie. Praesent sed commodo tortor, et ultrices tellus. Suspendisse malesuada sed mi quis ultrices. Suspendisse quis purus non ligula commodo tempor sit amet eget massa. Vivamus nunc diam, elementum sit amet est a, aliquam ultrices purus. Phasellus vestibulum enim nisl, at luctus leo mattis eu. Fusce non commodo est.

* https://github.com/cjayross/gravispy

FIG. 1. Illustration marking intermediary steps during the image forming process. The new image is first created that defaults to black. During execution, the *apply_lens* algorithm iterates through each pixel of the newly generated image and uses the passed lensing map to determine which pixel from the input should be placed at that location.



I. INTRODUCTION

II. MOTIVATION

III. METHODS

IV. RESULTS

V. DISCUSSION AND CONCLUSION

FIG. 2. Sample image used to test the application of a generated lensing map.

./png/earth.png

FIG. 3. Final result of applying the explicit Schwarzschild lensing map on the sample image shown in Fig. (2), with $M = 1\,\mathrm{km}$ and $R_O = 10\,\mathrm{km}$.

./png/example_lens1.png

FIG. 4. Second example of a generated Schwarzschild lensing map using the same space-time as in Fig. (3) but at a radius of $R_O = 2.5\,\mathrm{km}$, which is significantly closer to the event horizon.

./png/example_lens2.png

FIG. 5. Comparison between the three levels of approximation in the Schwarzschild lensing equation. The first graph indicates the differences near the singularity, while the latter shows that the amount of error decreases significantly as the radius of the observer increases.

./png/sc_lensing_near.png

./png/sc_lensing_far.png