Volumetric Fog Rendering

Siim Raudsepp, University of Tartu

Supervisor: Jaanus Jaggo, MSc

INTRODUCTION

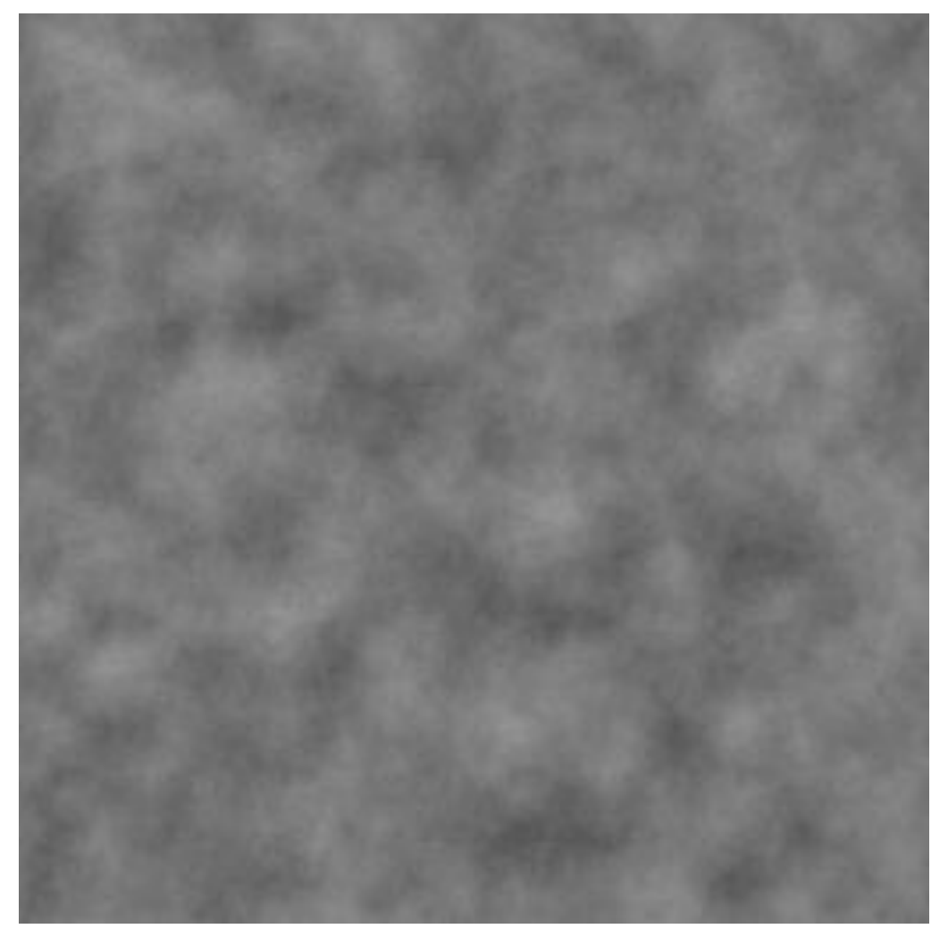
- Today, the computational power of graphics cards has increased so much that computer graphics applications do not have to rely on cheap hacks to simulate real world effects such as fog.
- Historically, fog was rendered according to the distance from the camera (depth fog) or using billboards, which are 2D images that rotate so they are always facing the camera.
- These methods cannot accurately simulate light rays moving through a volume.
- Thus, a new technique called volumetric fog was developed.
- Volumetric fog simulates light rays moving through a volume by using a technique known as raymarching.

OBJECTIVES

1) Explain how fog interacts with light in real life
2) Describe the algorithm for rendering volumetric fog
3) Implement a proof of concept application

NOISE

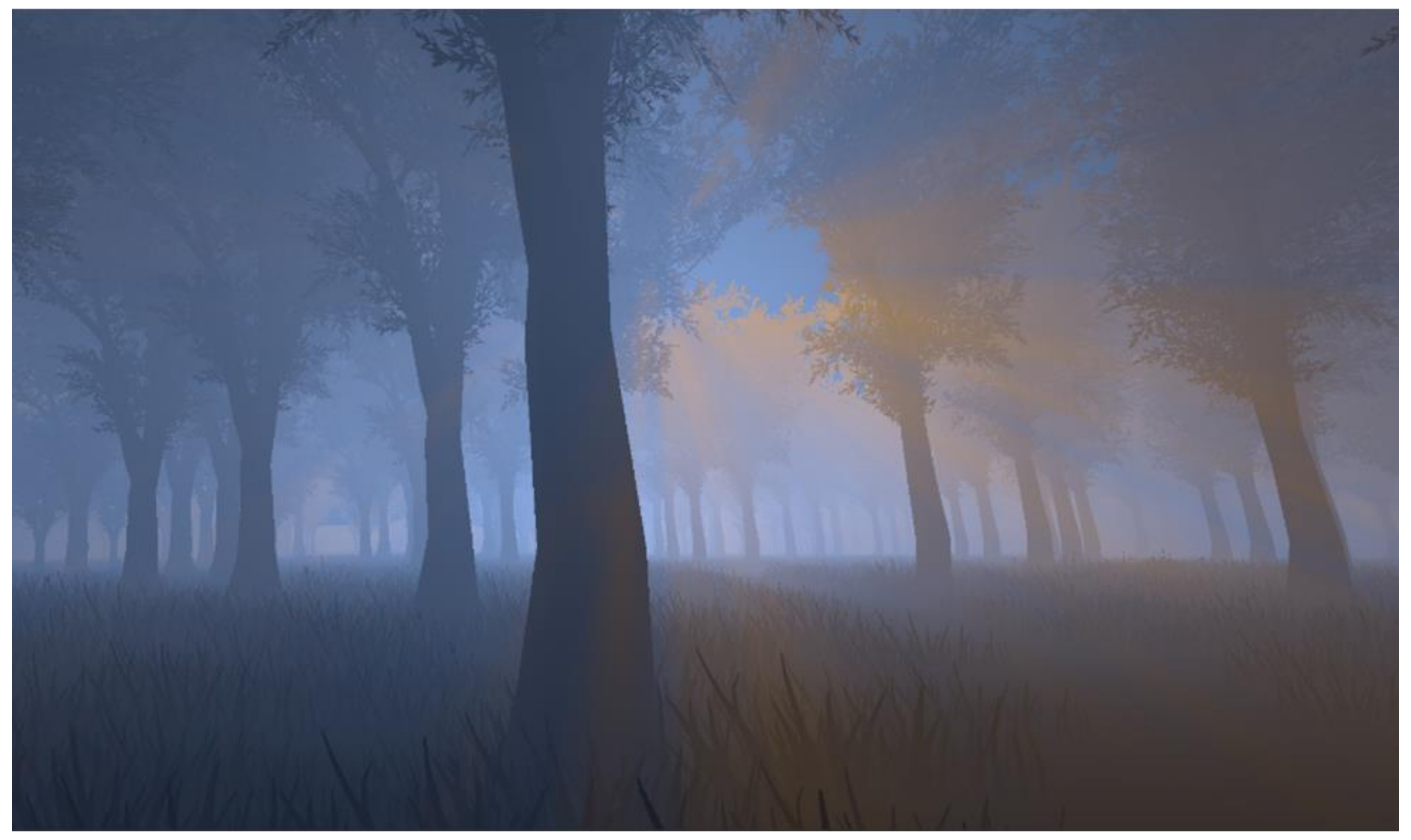
The texture that the fog is sampled from is generated procedurally using noise functions. Noise functions are functions that return a deterministic value corresponding to a given input. Multiple 2D noise textures can then be stitched together to create a 3D noise texutre.



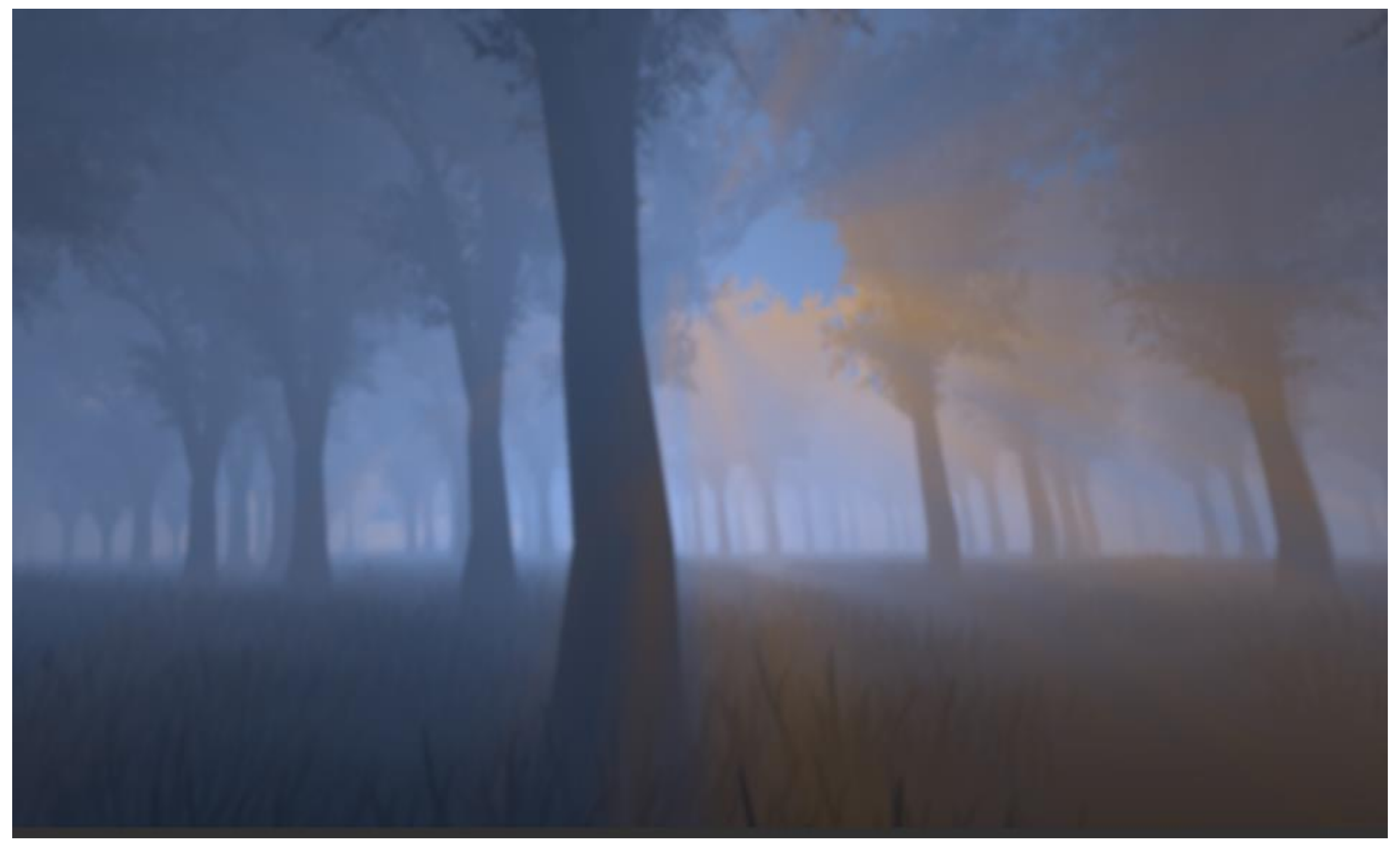
Perlin noise [1]

ALGORITHM

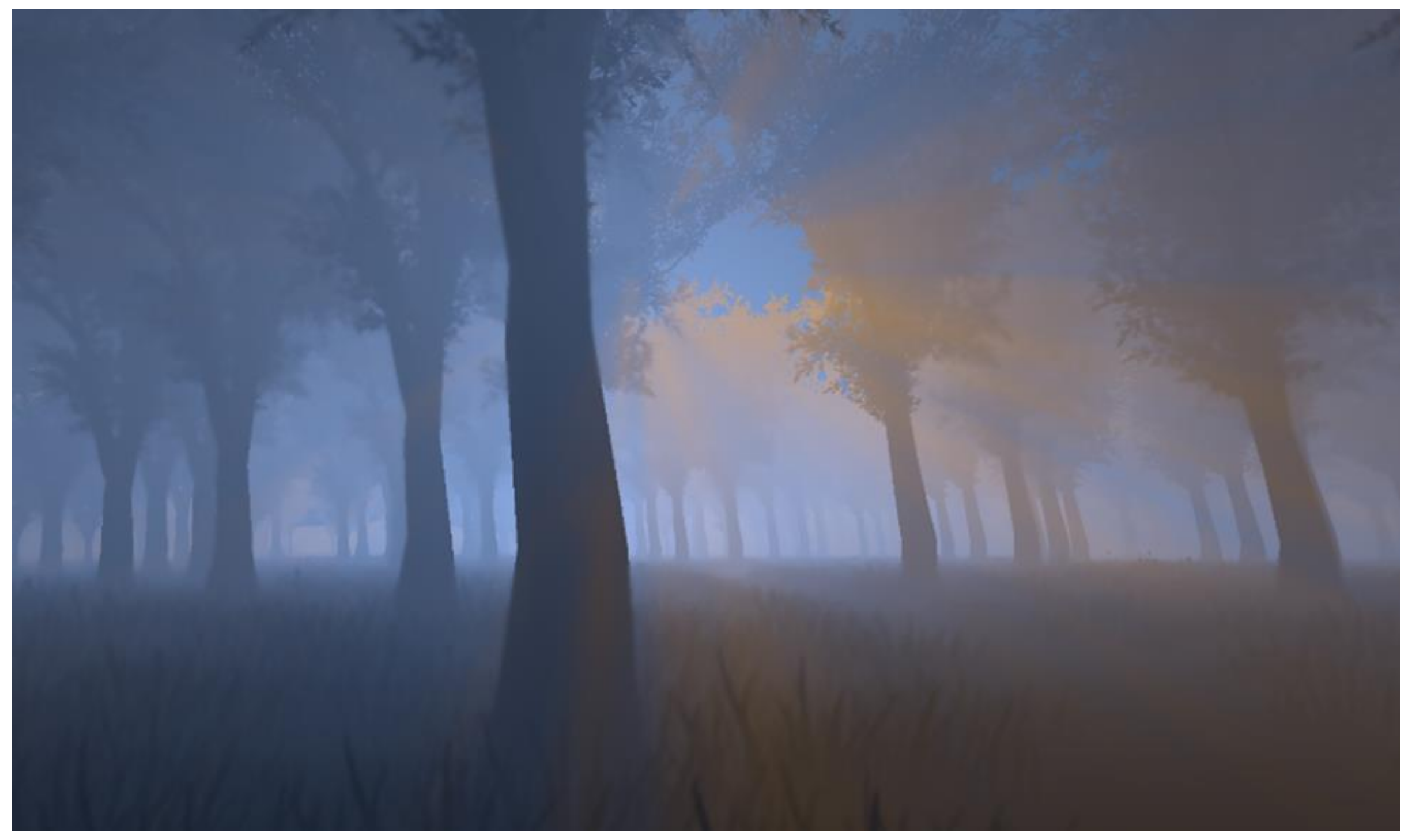
The workflow for rendering volumetric fog:



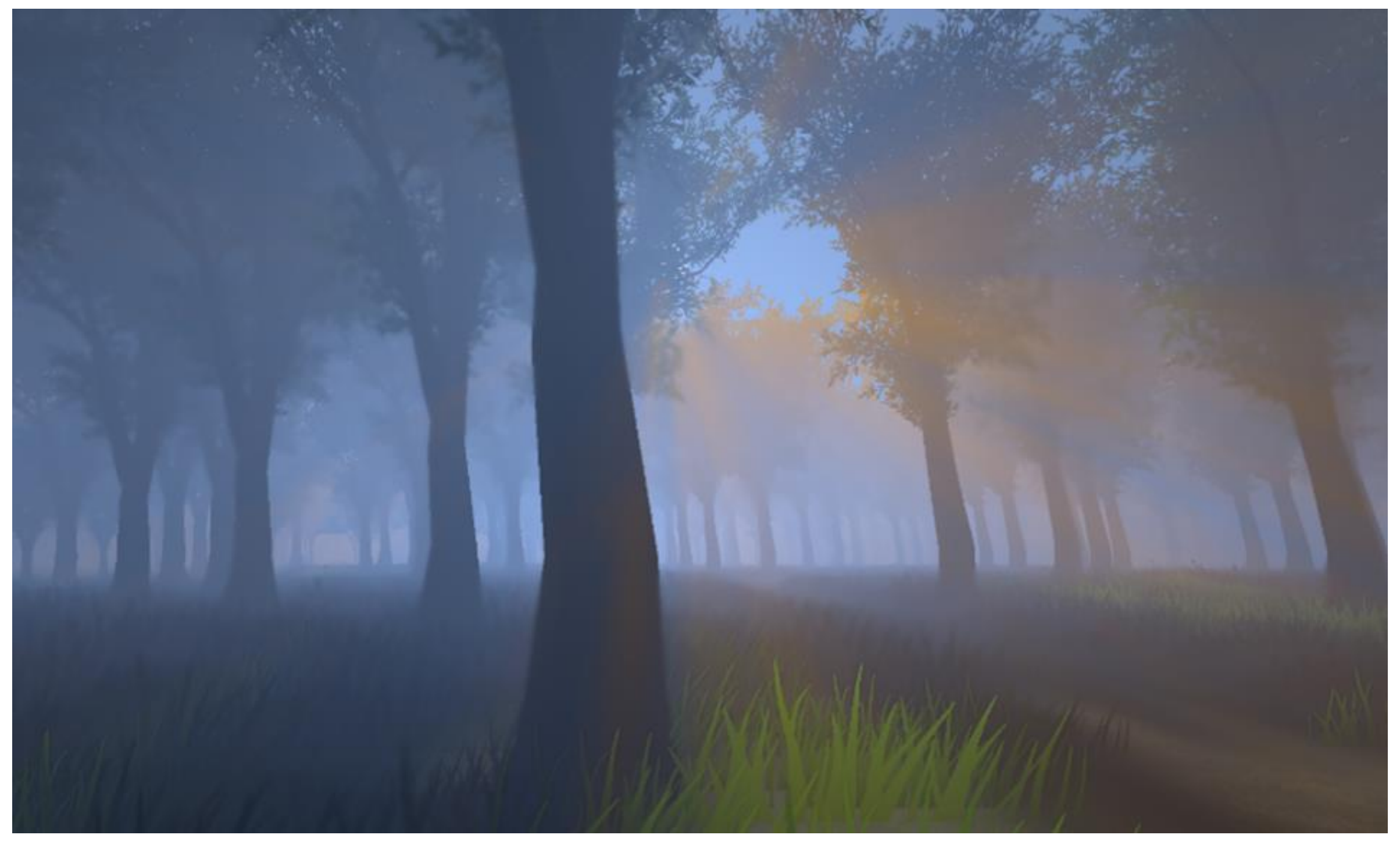
1. At each point of the volume the shadow map and noise texture are sampled. Color is added depending on the point being in shadow or not and the noise value represents the fog density.



2. The result from the previous step is blurred by applying Gaussian blur. This leaves the image blurry.



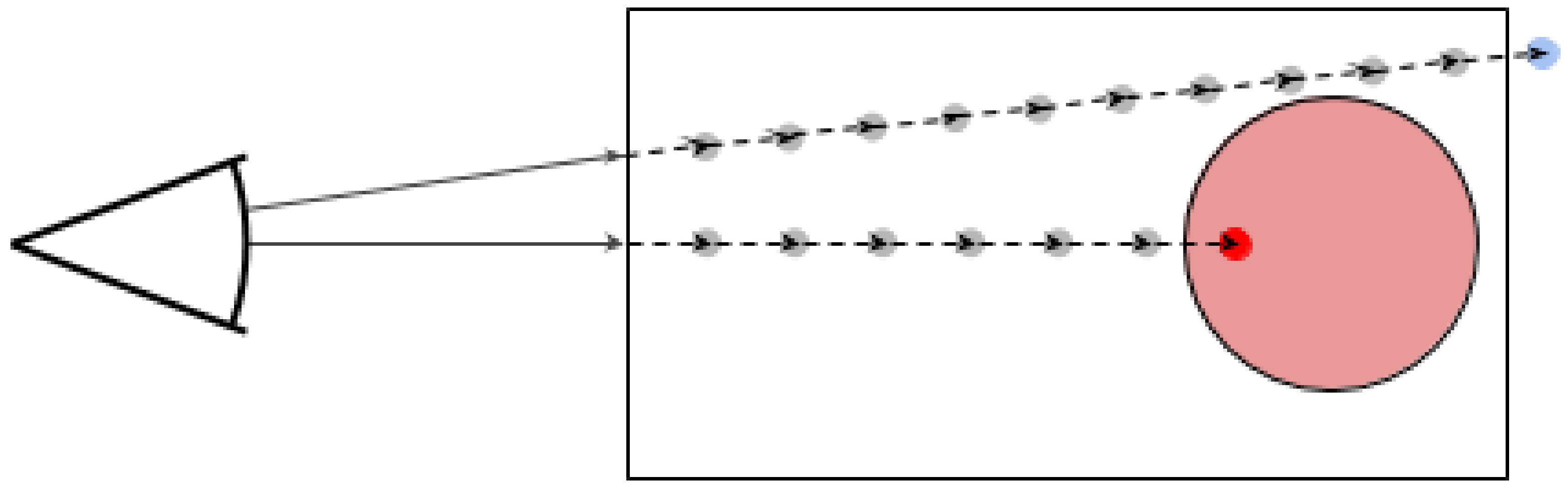
3. Bilateral filtering is added to keep the edges of the geometry sharp.



4. Finally, the fog is blended together with existing scene geometry and the skybox.

RAYMARCHING

Raymarching is a technique used in computer graphics to step through a volume. Any geometry, transparent or not, can be rendered using raymarching. The technique works by casting a ray into each pixel of the screen and then iteratively extending the ray, sampling the volume at each point and calculating the color based on the density value. For rendering fog, the iteration is terminated early if the maximum depth for a pixel is reached or if the alpha value of the pixel reaches 1.



Raymarching [2]

RESULTS

- A proof of concept application was made in the Unity game engine.
- The application is highly customizable allowing to change various parameters such as fog density, blur amount, the noise texture used, the scattering and extinction coefficients and many more.
- The thesis also contains benchmarks of the application.
- Many optimizations could be done in the future to improve the performance of the implementation, which were explained in the thesis.

REFERENCES

1. https://www.filterforge.com/filters/231.jpg
2. https://www.alanzucconi.com/2016/07/01/raymarching/
3. https://github.com/SiiMeR/unity-volumetric-fog project repository

CONTACT

Siim Raudsepp siimraudsepp@msn.com