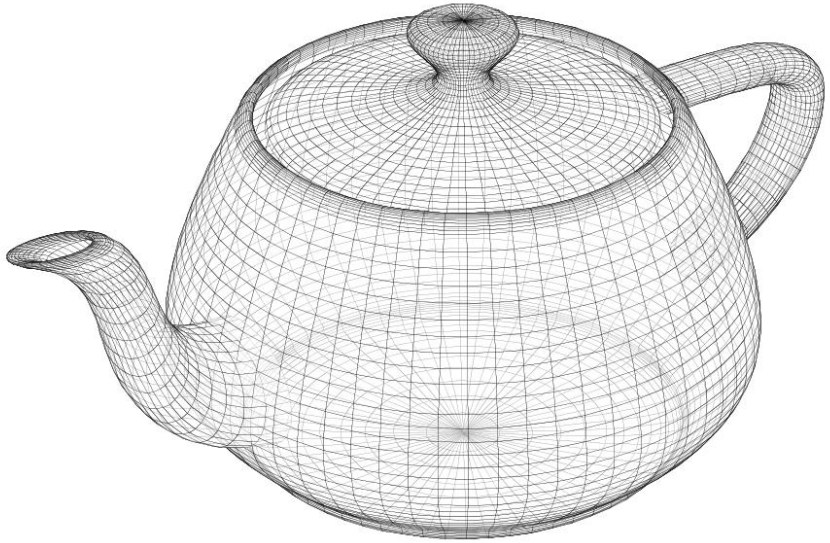


Fog

Interactive Computer Graphics

Eric Shaffer

ILLINOIS

Participating Media

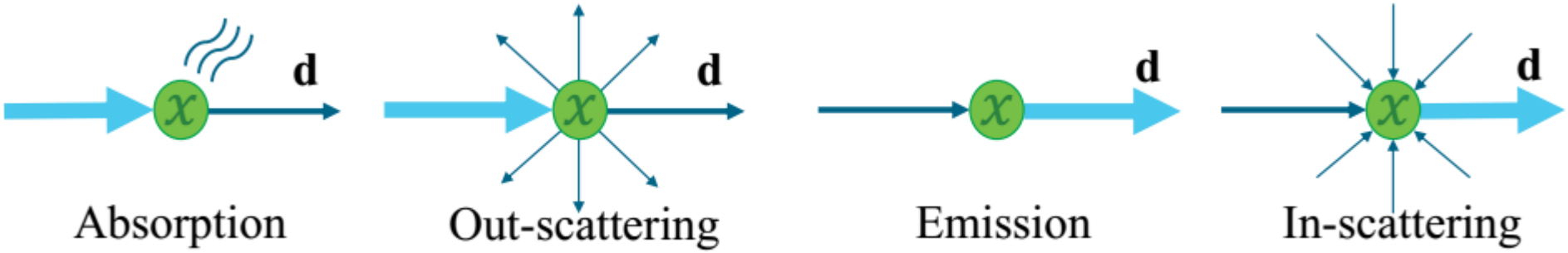
Fog is used in this image of a level from Battlefield 1, a DICE game, to reveal the complexity of the gameplay area. Depth fog is used to reveal the large-scale nature of the scenery.

-Real-Time Rendering, Fourth Edition

Participating media is the term used to describe volumes filled with particles.

They affect light that passes through them via scattering or absorption.

Modeling Light Transport Through Participating Media



Four types of events affect amount of radiance propagating along a ray through a medium

- Absorption — Photons are absorbed by the medium and transformed into heat or other forms of energy.
- Out-scattering — Photons are scattered away by bouncing off particles in the medium matter.
- Emission—Light can be emitted when media reaches a high heat.
- In-scattering—Photons from any direction can scatter into the current light path after bouncing off particles.

ILLINOIS

Modeling Media Properties

Symbol	Description	Unit
σ_a	Absorption coefficient	m^{-1}
σ_s	Scattering coefficient	m^{-1}
σ_t	Extinction coefficient	m^{-1}
ρ	Albedo	unitless
p	Phase function	sr^{-1}

$$extinction\ \sigma_t = \sigma_a + \sigma_s;$$

$$\rho = \frac{\sigma_s}{\sigma_s + \sigma_a} = \frac{\sigma_s}{\sigma_t}$$

- Adding photons to a path is a function of in-scattering and emission.
- Removing photons is a function of extinction
  - representing both absorption and out-scattering.
- Albedo represents the importance of scattering relative to absorption
  - The overall reflectiveness of the medium.
  - The value of ρ is within the range [0, 1].
  - A value close to 0 results in a murky medium, such as dark exhaust smoke.
  - A value close to 1 results in a brighter medium, such as air, clouds

ILLINOIS

Examples

Red wine features almost no scattering

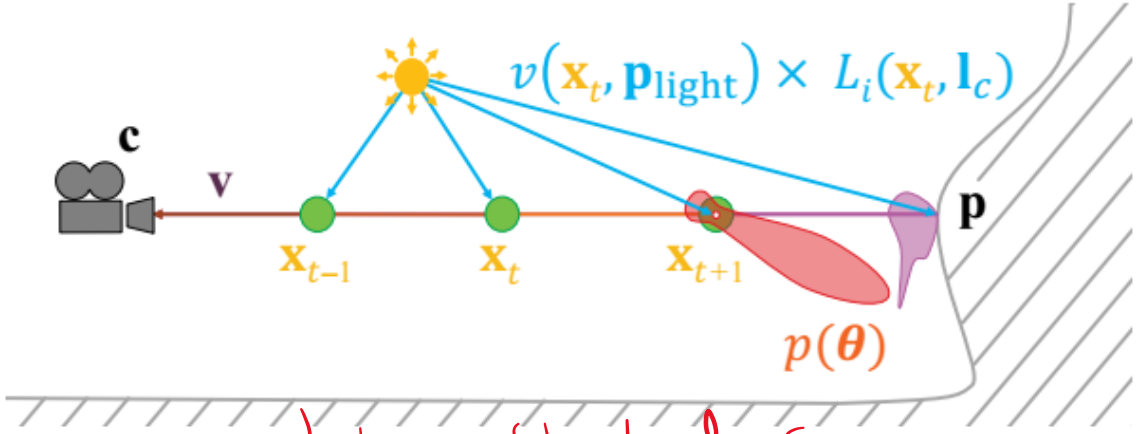
Has high absorption, giving it a translucent and colored appearance.

Milk has high scattering values, producing a cloudy and opaque appearance.

Milk also appears white thanks to a high albedo

ILLINOIS

Light Transport Equation



light reflected from p in direction v

light in-scattered at c-vt in direction v

$$L_i(\mathbf{c}, -\mathbf{v}) = T_r(\mathbf{c}, \mathbf{p}) L_o(\mathbf{p}, \mathbf{v}) + \int_{t=0}^{\|\mathbf{p}-\mathbf{c}\|} T_r(\mathbf{c}, \mathbf{c}-\mathbf{v}t) L_{\text{scat}}(\mathbf{c}-\mathbf{v}t, \mathbf{v}) \sigma_s dt,$$

light arriving @ c from direction $-v$

fraction of light allowed through from p to c

fraction of light allowed through from c-vt to c

scattering coefficient

ILLINOIS

Transmittance

The transmittance T_r represents the ratio of light that is able to get through a medium over a certain distance.

According to the Beer-Lambert law:

$$T_r(\mathbf{x}_a, \mathbf{x}_b) = e^{-\tau}, \quad \text{where} \quad \tau = \int_{\mathbf{x}=\mathbf{x}_a}^{\mathbf{x}_b} \sigma_t(\mathbf{x}) \, \|d\mathbf{x}\|.$$

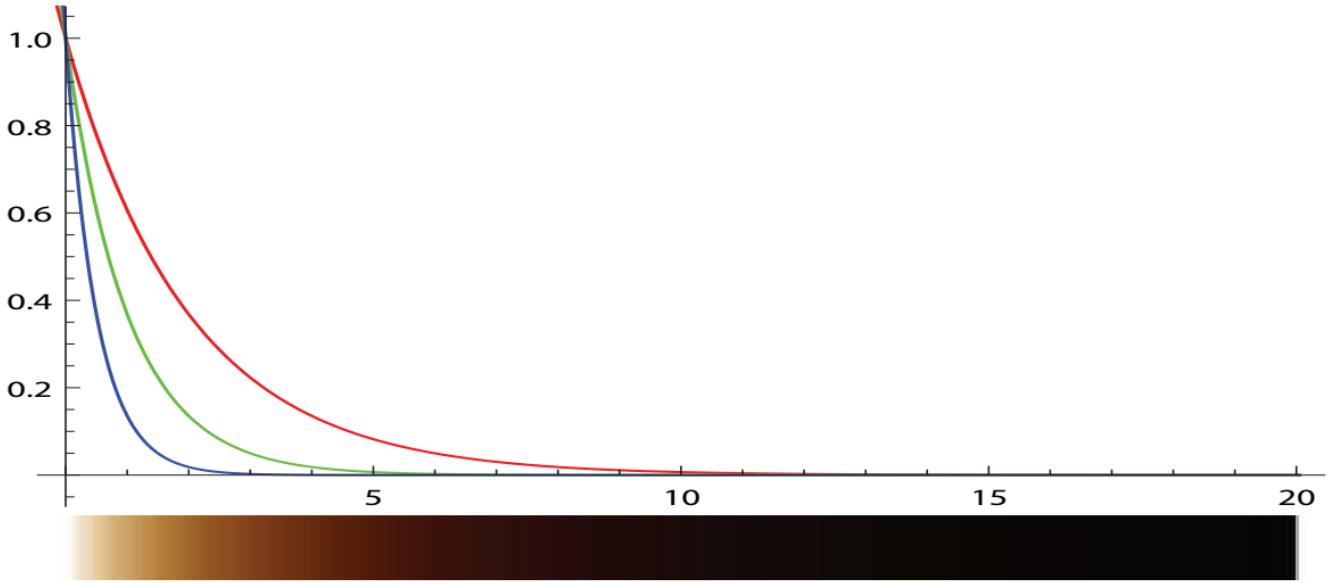
The ***optical depth*** τ is unitless and represents the amount of light attenuation.

The higher the extinction σ_t or distance traversed, the larger the optical depth, the less light will travel through the medium.

An optical depth τ = 1 will remove approximately 60% of the light.

ILLINOIS

Transmittance



Transmittance as a function of depth, with σ_t = (0.5, 1.0, 2.0).

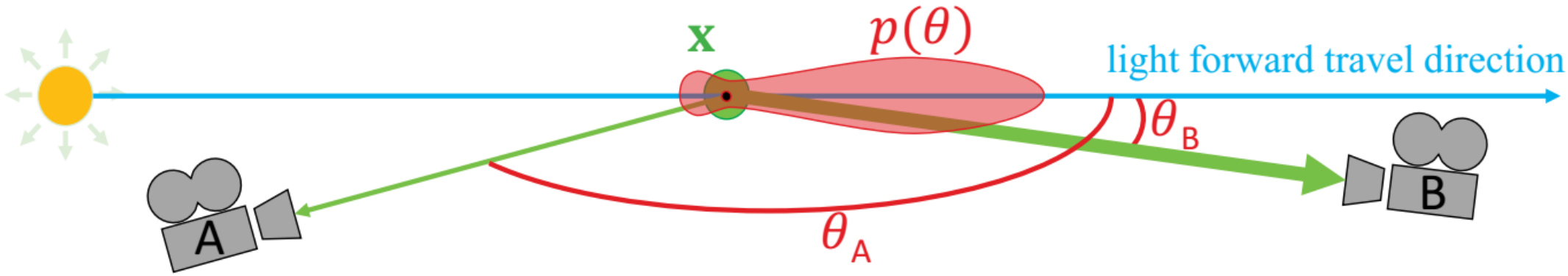
Lower extinction coefficient σ_t for the red component leads to more red color being transmitted.

Note that transmittance is wavelength dependent…why might that be?

$$T_r(\mathbf{x}_a, \mathbf{x}_b) = e^{-\tau}, \quad \text{where} \quad \tau = \int_{\mathbf{x}=\mathbf{x}_a}^{\mathbf{x}_b} \sigma_t(\mathbf{x}) \, \|d\mathbf{x}\|.$$

ILLINOIS

Phase Functions



A participating medium is composed of particles with varying radii.

These particles influence probability light will scatter in a given direction, relative to light's forward travel direction.

Phase function: gives the probability and distribution of scattering directions

Phase function in red in diagram
The parameter θ is the angle between the light's forward travel path in blue and toward direction v in green.

Different media are modeled using different phase functions….

ILLINOIS

Scattering

$$L_{\text{scat}}(\mathbf{x}, \mathbf{v}) = \pi \sum_{i=1}^{n} p(\mathbf{v}, \mathbf{l}_{c_i}) v(\mathbf{x}, \mathbf{p}_{\text{light}_i}) c_{\text{light}_i}(\|\mathbf{x} - \mathbf{p}_{\text{light}_i}\|),$$

number of lights

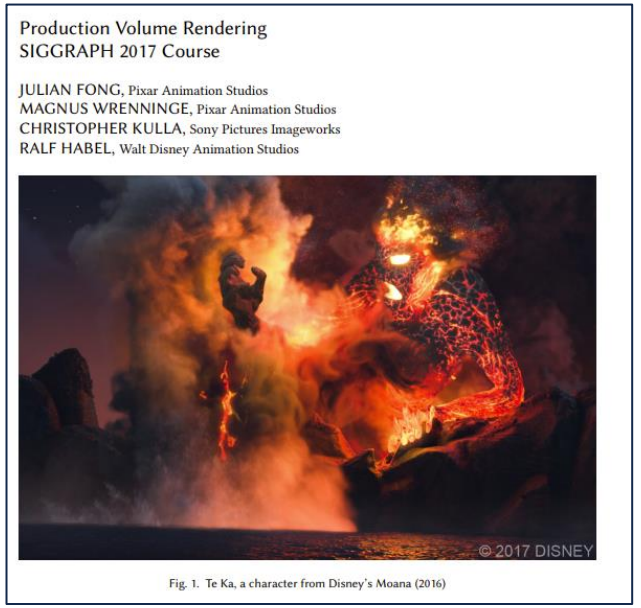
radiance of light i as function of distance

light in-scattering at point x in direction v

phase function

visibility function ... 0 or 1 ?

For more details….like why π….consult
https://graphics.pixar.com/library/ProductionVolumeRendering/paper.pdf



ILLINOIS

Large Scale Fog

Depth fog….simplify by just considering transmittance….

Blend a fog color with shading color as a function of depth

$$\mathbf{c} = f\mathbf{c}_i + (1 - f)\mathbf{c}_f$$

Linear drop off of reflected color

$$f = \frac{z_{\text{end}} - z_s}{z_{\text{end}} - z_{\text{start}}}$$

Or…more physically correct…exponential drop off of reflected color

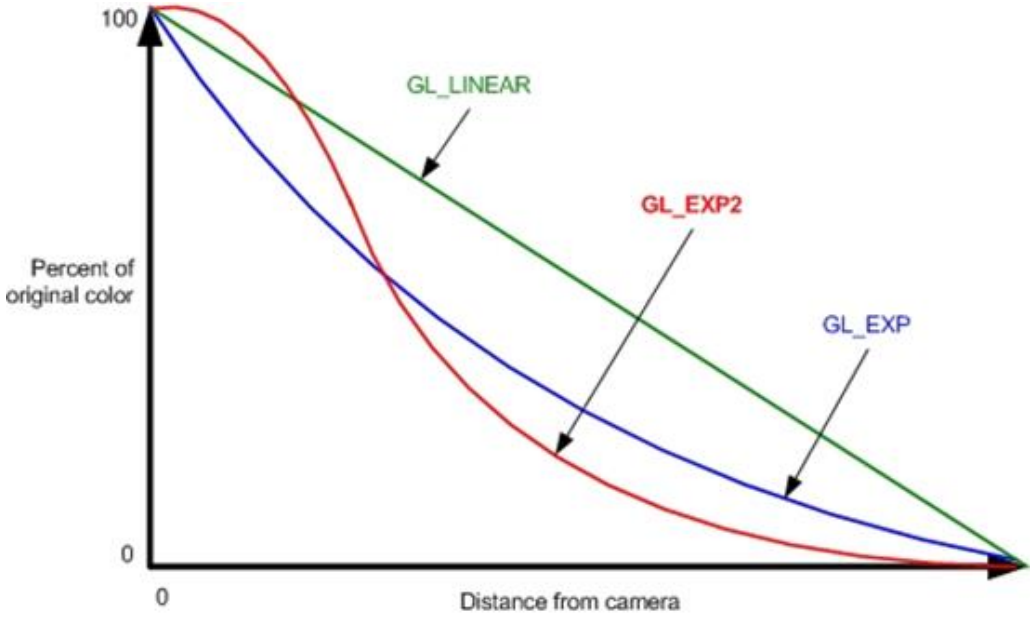
$$f = e^{-d_f z_s}$$

scalar d_f is a user parameter that controls the density of the fog

ILLINOIS

Implementing Fog Factor

```
const float LOG2 = 1.442695;
float fogDensity = 0.0005
float fogFactor = exp2( -fogDensity * fogDensity * fogDist * fogDist * LOG2 );
fogFactor = clamp(fogFactor, 0.0, 1.0);
```



On the next foggy day, you can try to verify by eyesight if fog works this way…

ILLINOIS

Implementing Depth Fog

```
// Passed in from the vertex shader.
in vec3 v_position;
// Passed from JS
uniform vec4 u_fogColor;
uniform float u_fogDensity;
out vec4 outColor;

void main() {
 #define LOG2 1.442695
  vec4 color = …  //shade the fragment
  float fogDistance = length(v_position);
  float fogAmount =
        1. - exp2(-u_fogDensity * u_fogDensity * fogDistance * fogDistance * LOG2);
  fogAmount = clamp(fogAmount, 0., 1.);
  outColor = mix(color, u_fogColor, fogAmount);
}
```

mix performs a linear interpolation between x and y using a to weight between them. The return value is computed as $x \times (1 - a) + y \times a$.

ILLINOIS

Even More of a Hack….

Use gl_FragCoord.z for depth

gl_FragCoord is a global variable that WebGL sets

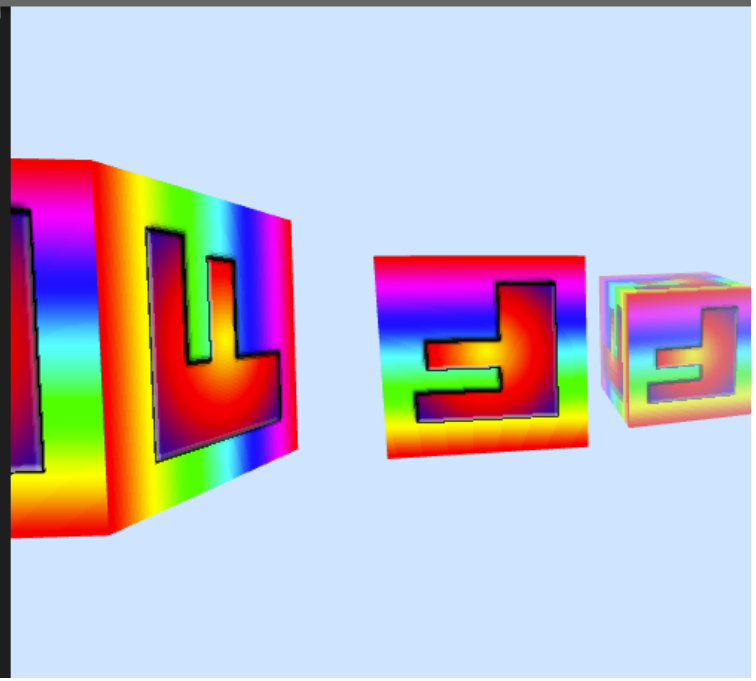
The x and y components are the coordinate of the pixel being drawn

The z coordinate is the non-linearly depth of that pixel from 0 to 1

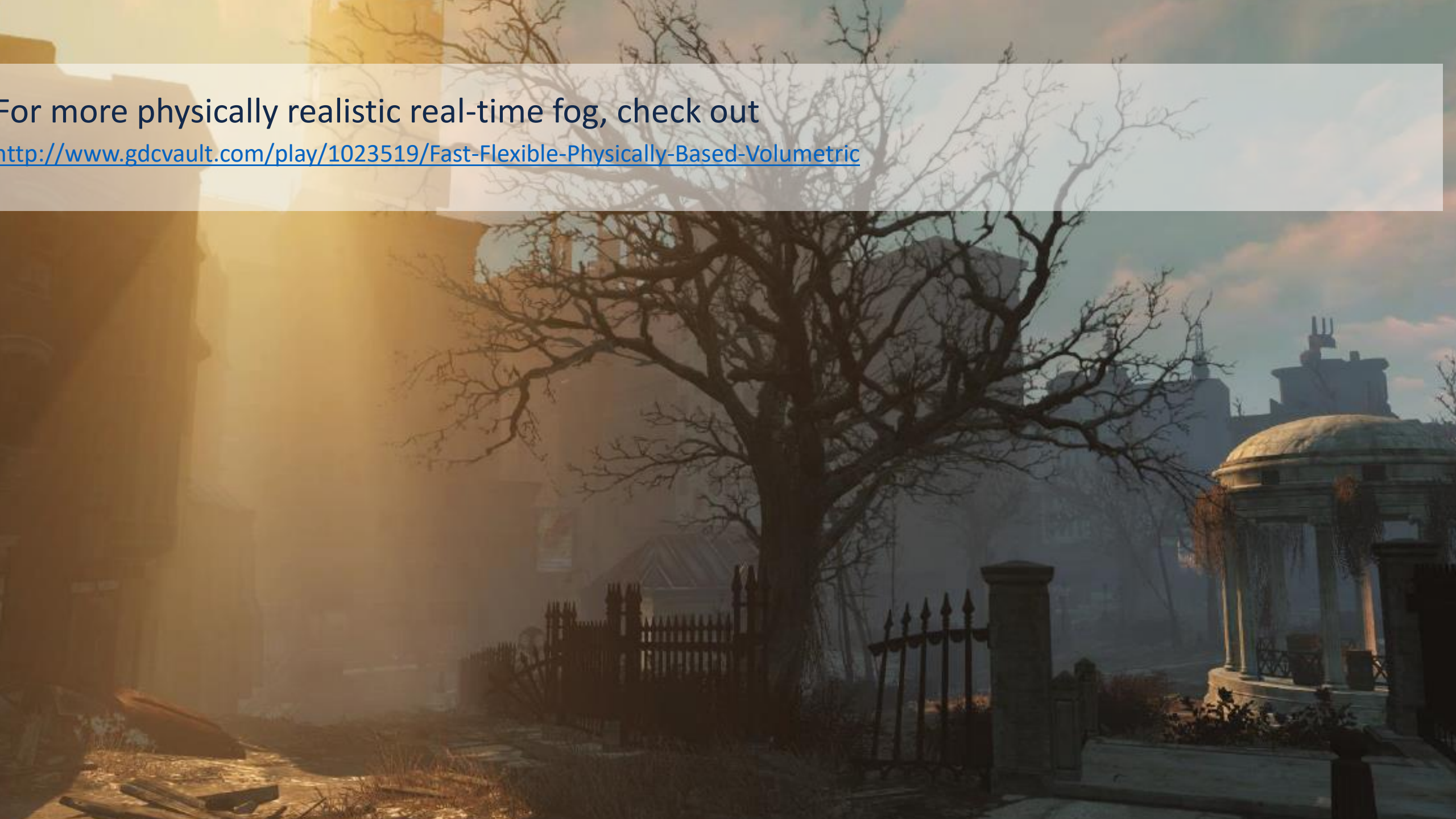
From https://webglfundamentals.org/

JS HTML CSS Export Result Run

```
</script>
<!-- fragment shader -->
<script  id="fragment-shader-3d" type="x-shader/x-fragment">
precision mediump float;

// Passed in from the vertex shader.
varying vec2 v_texcoord;

// The texture.
uniform sampler2D u_texture;
uniform vec4 u_fogColor;
uniform float u_fogNear;
uniform float u_fogFar;

void main() {
  vec4 color = texture2D(u_texture, v_texcoord);

  float fogAmount = smoothstep(u_fogNear, u_fogFar, gl_FragCoord.z);

  gl_FragColor = mix(color, u_fogColor, fogAmount);
}
</script><!--
for most samples webgl-utils only provides shader compiling/linking and
```



ILLINOIS



For more physically realistic real-time fog, check out

http://www.gdcvault.com/play/1023519/Fast-Flexible-Physically-Based-Volumetric