

# Relativistic Ray Tracing

Shikai Qiu

## Abstract

Relativistic effects arise when the camera is in relative motion with the scene and will become significant when the speed is comparable to the speed of light. As motion close to the speed of light is rarely encountered during everyday life, a photorealistic renderer capable of rendering relativistic effects allow us to observe and appreciate these effects that are otherwise inaccessible. It is the goal of this project to extend the path tracer from project 3-1 to one that correctly incorporates relativistic effects. For simplicity, we will work in flat spacetime and deal with special relativity only, where light travels in straight lines in any inertial frame thanks to the absence of gravity. In the end, our rendered images will reflect well-known effects such as length contraction and relativistic doppler shift.

## Notation

$c$  the speed of light.

$\vec{\beta} = \vec{v}/c$  velocity divided by the speed of light.

$\gamma = 1/\sqrt{1 - \beta^2}$ .

$x^\mu = (ct, \vec{x})$  a 4-vector that specifies a point in spacetime.

$k^\mu = (c\omega, \vec{k})$  a four wave-vector specifying a frequency and direction of propagation.

$\mathcal{E}(x^\mu)$  the irradiance at a point.

$\mathcal{L}(x^\mu, \vec{k}, \lambda)$  the (outgoing) radiance at a point with a given wavelength and direction.

$\text{tr}$  the trace operator in ray tracing.

The superscript  $\mu$  for 4 vectors will sometimes be suppressed without ambiguity as 3 vectors always have arrows on top.

## Geometry

In this part, we will focus on the geometric effects that arises when the camera is in motion with respect to the scene. Previously, in the wavelength-independent path tracer, the irradiance at a point  $x$  is given by:

$$\mathcal{E}(x) = \int d^2\vec{k} \cos(\theta) \mathcal{L}(\text{tr}(x, \vec{k}), -\vec{k}).$$

In the case of a pin-hole camera with  $x$  a point on the sensor and  $k$  the direction from the point to the pinhole, we can make the approximation that was used in the project 3-1 implementation, assuming small  $\theta$  for all  $x$ :

$$\mathcal{E}(x) \propto \mathcal{L}(\text{tr}(x, \vec{k}), -\vec{k}).$$

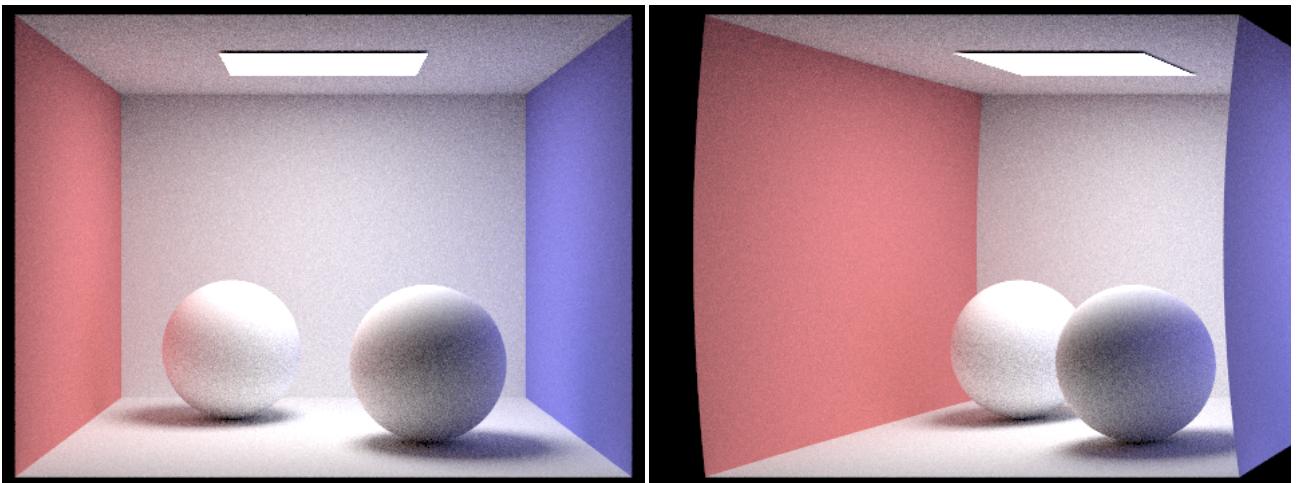
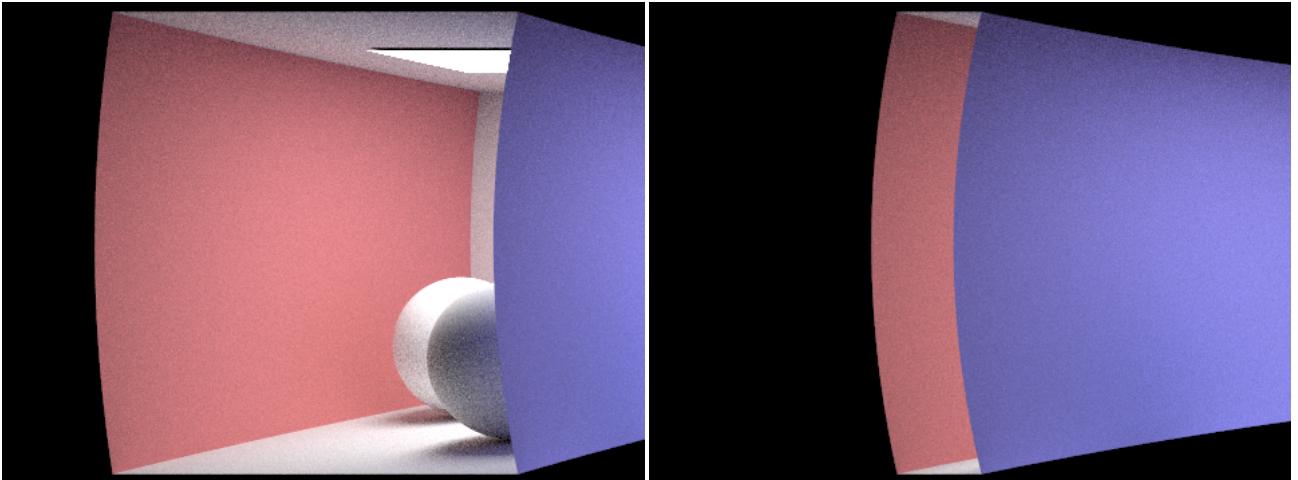
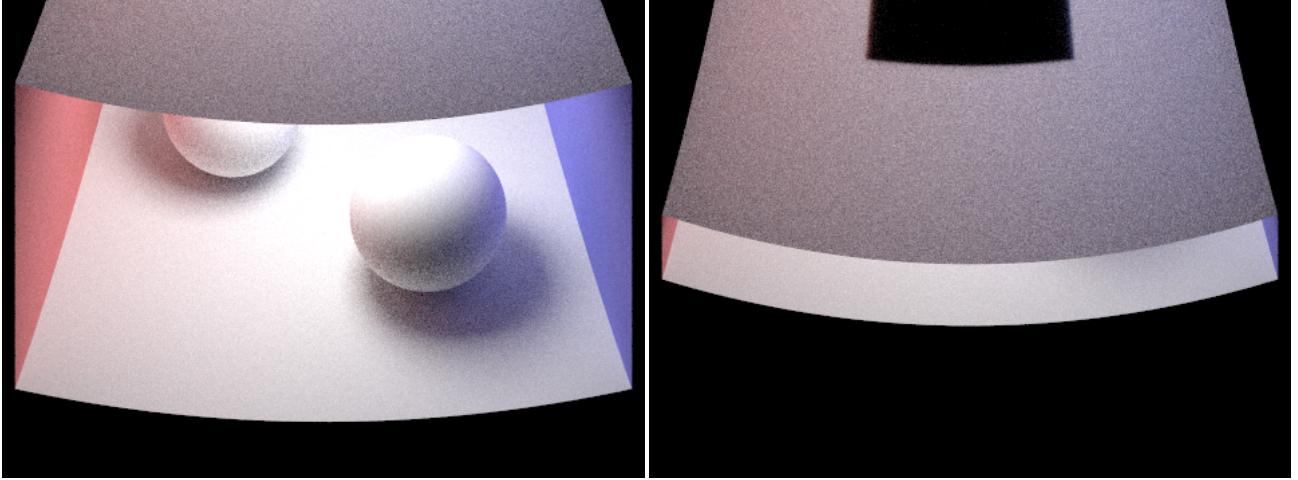
In order to render a photo taken in the camera frame at time  $t'$ , we need to evaluate  $\mathcal{E}'(x')$ , the irradiance at  $x'$  in the the camera frame, for each  $x'$  on the sensor at  $t'$ . We start by transforming  $x'$  into its value in the lab frame(frame in which the scene is stationary) denoted by  $x$ . This transformation is given by the Lorentz transformation:

$$\begin{aligned} x' &= \Lambda x \\ x &= \Lambda^{-1} x' \\ \Lambda &= \begin{pmatrix} \gamma & -\gamma\beta_x & -\gamma\beta_y & -\gamma\beta_z \\ -\gamma\beta_x & 1 + (\gamma - 1)\beta_x\beta_x/\beta^2 & (\gamma - 1)\beta_x\beta_y/\beta^2 & (\gamma - 1)\beta_x\beta_z/\beta^2 \\ -\gamma\beta_y & (\gamma - 1)\beta_x\beta_y/\beta^2 & 1 + (\gamma - 1)\beta_y\beta_y/\beta^2 & (\gamma - 1)\beta_y\beta_z/\beta^2 \\ -\gamma\beta_z & (\gamma - 1)\beta_x\beta_z/\beta^2 & (\gamma - 1)\beta_z\beta_y/\beta^2 & 1 + (\gamma - 1)\beta_z\beta_z/\beta^2 \end{pmatrix} \end{aligned}$$

Using the Lorentz transformation, we can estimate the camera frame irradiance as follows:

$$\mathcal{E}'(x') \approx \mathcal{E}(\Lambda^{-1}x')$$

This is approximation can give us the wrong color and brightness for reasons which we will discuss later. Nevertheless, it will allow us to render images with the correct geometry.

Fig. 1-1:  $ct = 5 \vec{\beta} = (0, 0, 0)$ Fig. 1-2:  $ct = 5 \vec{\beta} = (0.5, 0, 0)$ Fig. 1-3:  $ct = 5 \vec{\beta} = (0.75, 0, 0)$ Fig. 1-4:  $ct = 5 \vec{\beta} = (0.99, 0, 0)$ Fig. 1-5:  $ct = 5 \vec{\beta} = (0, 0.75, 0)$ Fig. 1-6:  $ct = 5 \vec{\beta} = (0, 0.99, 0)$ 

At  $t = 0$ , the camera is at its initial position specified by the input files. The caption for each image indicates the camera-frame time at which the photo is taken and the velocity of the camera observed in the lab frame. Lorentz transformation predicts that space will be contracted in the direction of relative motion. In Fig. 1-4, we can clearly see that is indeed the case. We also see that the box appears to be non-linearly deformed, which can not be a result of Lorentz transformation as it is a linear transformation. Rather, this effect arises because light from different points on the object takes different amount of time to reach the sensor, which can introduce non-linearity in the resulted image. This also explains why the right wall of the box appears to be displaced further towards the  $x$  direction as we move towards the back, because as we trace a ray deeper into the scene, we also go further back in time.

## Relativistic Doppler shift

As mentioned earlier, so far our path tracer only guarantees the correct geometry of the rendered image. Special relativity predicts that the wavelength of light will change depending on the relative motion between an observer and the light source. In particular, one can show from the fact that four wave vector  $k^\mu = (c\omega, \vec{k})$  transforms as a four vector that

$$\omega' = \frac{1}{\alpha(\hat{k}')} \omega$$

$$\lambda' = \alpha(\hat{k}')\lambda$$

$$\alpha(\hat{k}') := \gamma(1 + \hat{k}' \cdot \vec{\beta})$$

This shift in frequency is known as the relativistic Doppler shift. To implement this effect, we must first extend our path tracer to perform wavelength dependent ray tracing. We make the following design choices:

- $\mathcal{L}(x, \vec{k}, \lambda) \propto \mathcal{L}(x, \vec{k})$  light source emits with a constant spectrum distribution over all wavelength (one might argue this is unphysical, but we can do so for simplicity).
- Approximate human eye response functions  $r_{\text{color}}(\lambda)$  with normal distributions  $\mathcal{N}(\mu_{\text{color}}, \sigma_{\text{color}})$ .
- $(\mu_{\text{red}}, \mu_{\text{green}}, \mu_{\text{blue}}) = (600, 550, 450)$
- $(\sigma_{\text{red}}, \sigma_{\text{green}}, \sigma_{\text{blue}}) = (25, 25, 15)$
- Wavelength remains unchanged before and after scattering, namely  $f(x, \vec{k}, \vec{k}', \lambda, \lambda') = f(x, \vec{k}, \vec{k}', \lambda)\delta(\lambda - \lambda')$ , so that the rendering equation simplifies and has the same form as the wavelength-independent one:

$$\mathcal{L}(x, \vec{k}, \lambda) = \int d^2 \vec{k}' \cos(\theta') \int d\lambda' f(x, \vec{k}, \vec{k}', \lambda, \lambda') \mathcal{L}(\text{tr}(x, \vec{k}', -\vec{k}', \lambda') = \int d^2 \vec{k}' \cos(\theta') f(x, \vec{k}, \vec{k}', \lambda) \mathcal{L}(\text{tr}(x, \vec{k}'), -\vec{k}', \lambda)$$

- Construct wavelength dependent BSDF  $f$  according to the  $(r, g, b)$ -valued BSDF  $\vec{f}$  as follows:

$$f(x, \vec{k}, \vec{k}', \lambda) \propto \sum_{i \in \{r, g, b\}} f_i(x, \vec{k}, \vec{k}') g_i(\lambda, f_i(x, \vec{k}, \vec{k}')) / |f|^{1/2}$$

$$g_i(\lambda, f) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-(\lambda - \mu_i)^2 / 2\sigma_i^2} \quad \sigma_i = \sigma_i / f$$

The above definitions provide everything we need for wavelength dependent ray tracing. Our final equation for obtaining the RGB value  $\vec{\psi}' = (r, g, b)$  for each pixel  $p$  in the camera frame then follows:

$$\begin{aligned} \vec{\psi}'(t') &= \int_{\text{pixel}} d^2 \vec{x}' \int d\lambda' \vec{r}(\lambda') \mathcal{E}'(x', \lambda') \\ &\approx \int_{\text{pixel}} d^2 \vec{x}' \int d\lambda' \vec{r}(\lambda') \mathcal{E}(\Lambda^{-1}x, \lambda'/\alpha(\hat{k}')) \\ &\propto \int_{\text{pixel}} d^2 \vec{x}' \int d\lambda' \vec{r}(\lambda') \mathcal{L}(\text{tr}(\vec{x}, \vec{k}), -\vec{k}, \lambda) \\ &= \mathbb{E}_{x' \sim p(x')} \left[ \frac{1}{p(x')} \mathbb{E}_{\lambda' \sim p(\lambda')} \left[ \frac{1}{p(\lambda')} \vec{r}(\lambda') \mathcal{L}(\text{tr}(\vec{x}, \vec{k}), -\vec{k}, \lambda) \right] \right] \end{aligned}$$

where  $\vec{k}'$  points from  $\vec{x}'$  to the pinhole in the camera frame

Here in line 2 we used  $\mathcal{E}'(x', \lambda') \approx \mathcal{E}(\Lambda^{-1}x, \lambda'/\alpha(\hat{k}'))$  as motivated by the transformation properties of  $x'$  and  $k'$ , in line 3 we used the pinhole and small angle approximation as before, and in line 4 we used Monte-Carlo integration. The expression in the last line explicitly shows how the new algorithm is implemented. In essence, the only difference from the original path tracer is the addition of a round of sampling over wavelength for each point on the sensor, and an application of Lorentz transformation for each sampled ray to obtain the lab frame sensor coordinates, wave propagation direction, and wavelength.

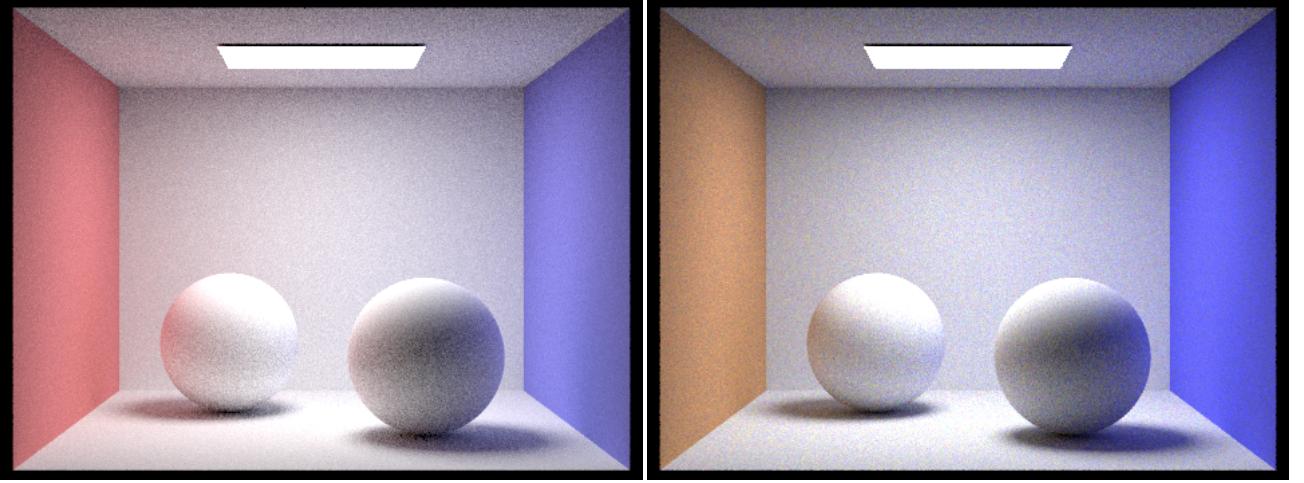
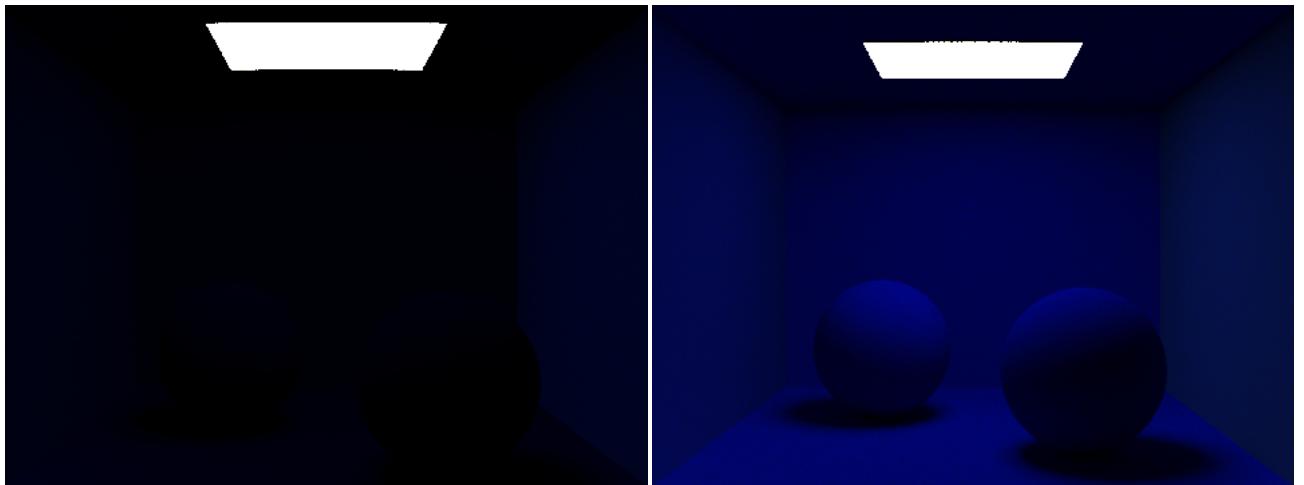
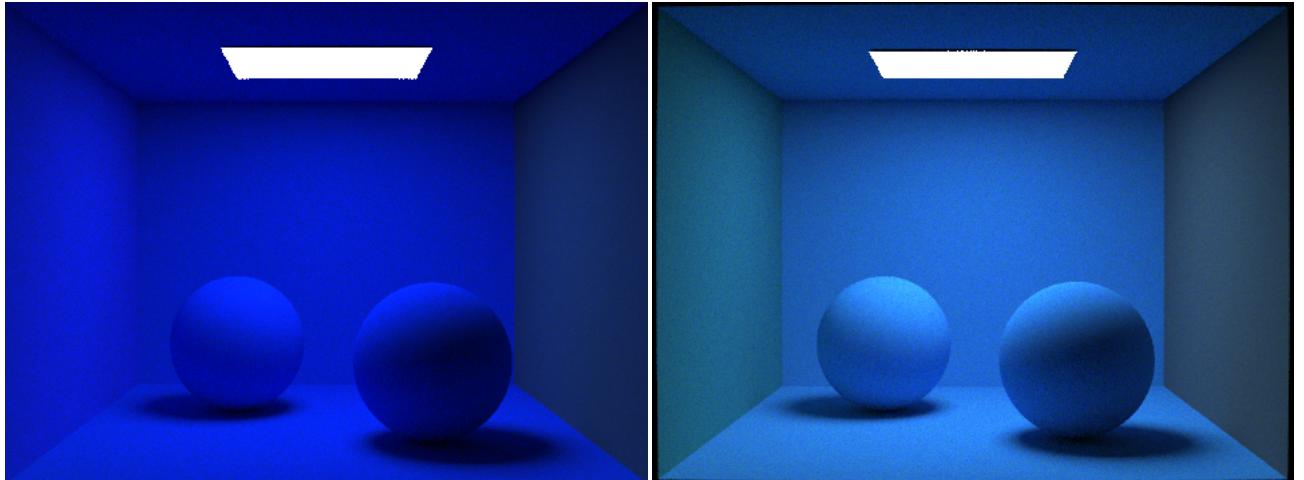
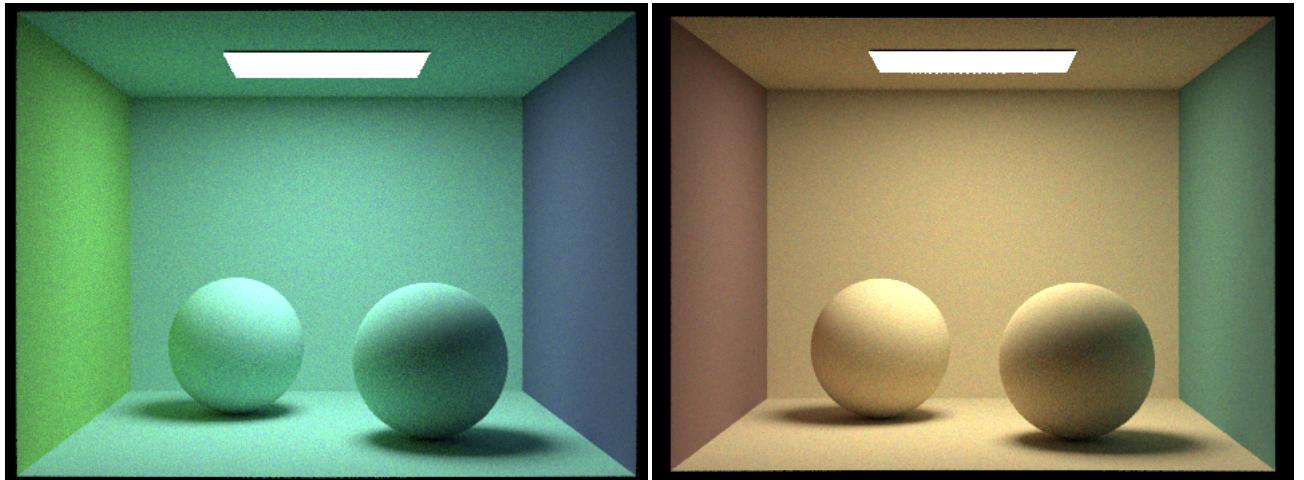
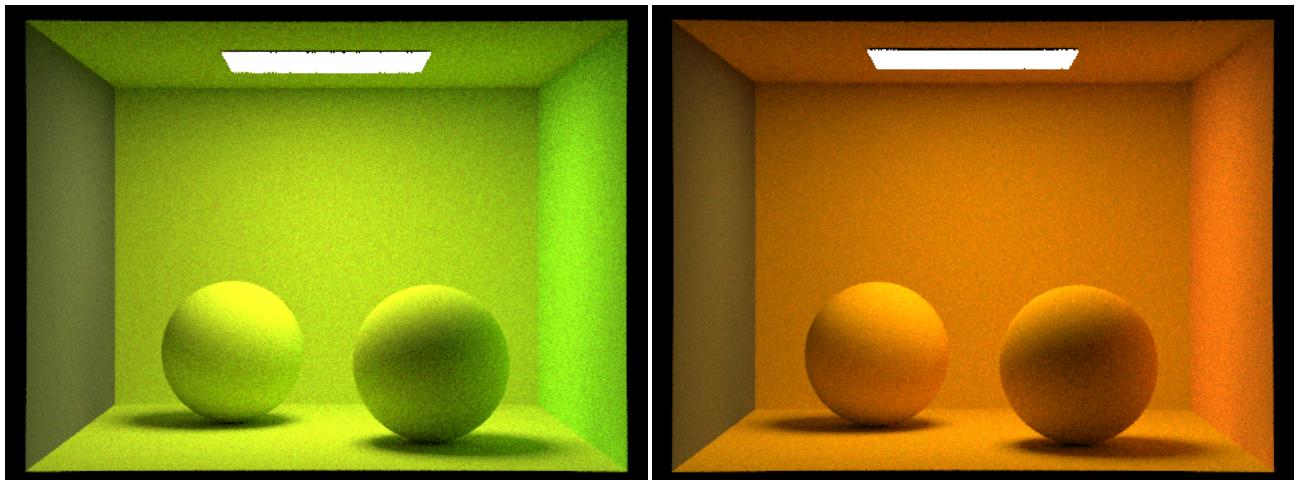


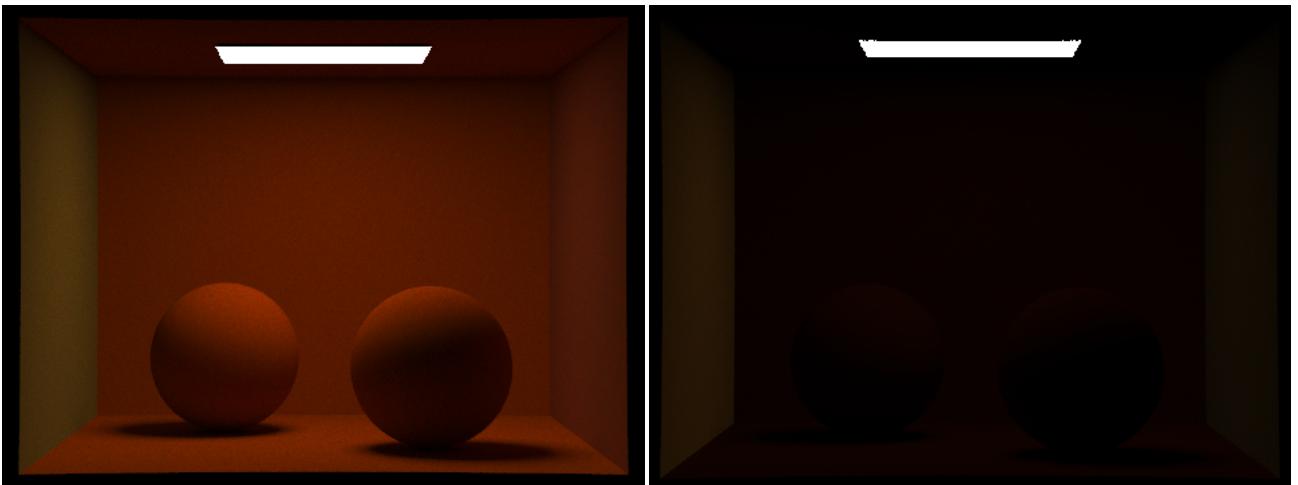
Fig. 2-1: wavelength-independent  $ct = 0 \vec{\beta} = (0, 0, 0)$

Fig. 2-2: wavelength-dependent  $ct = 0 \vec{\beta} = (0, 0, 0)$

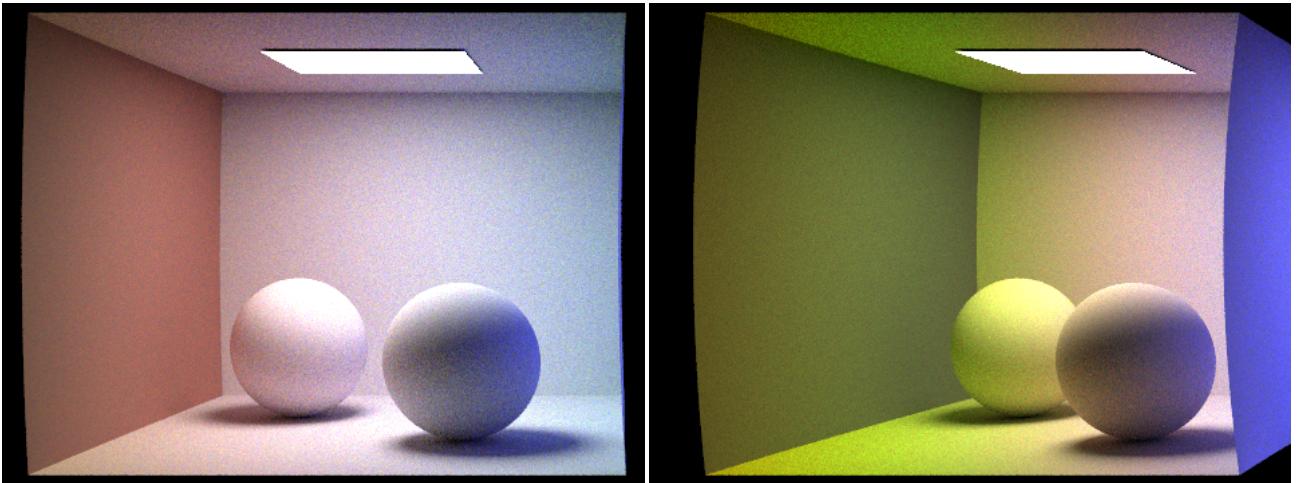
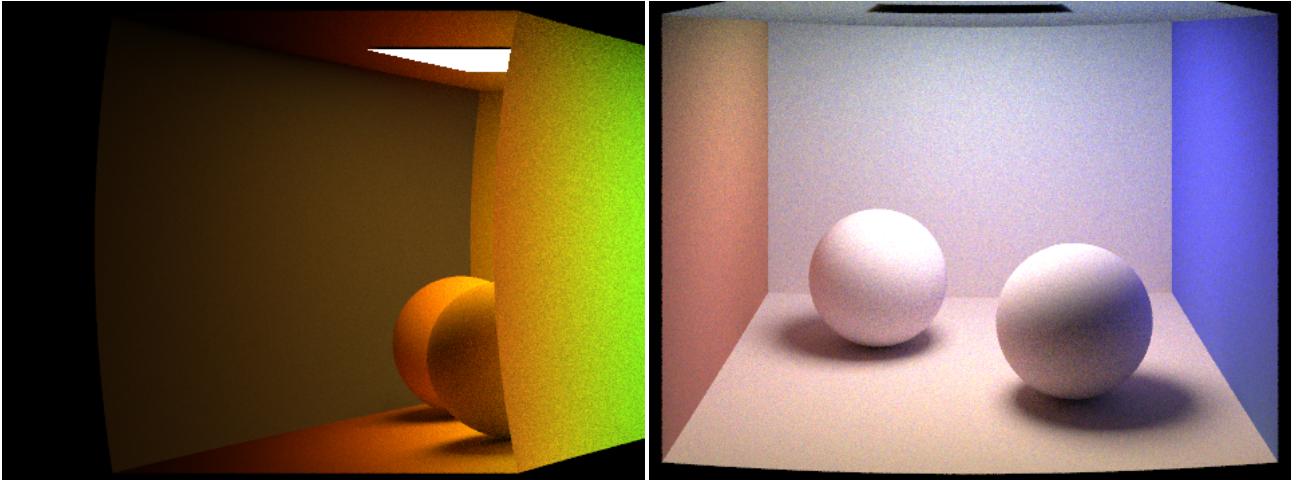
Fig. 2-1 and 2-2 shows a comparison between the image rendered by our wavelength-dependent path tracer and the image rendered by the original implementation from project 3-1 that uses  $(r, g, b)$ -only representation through out the pipeline. Our current parameters do not exactly reproduce the color but is qualitatively correct.

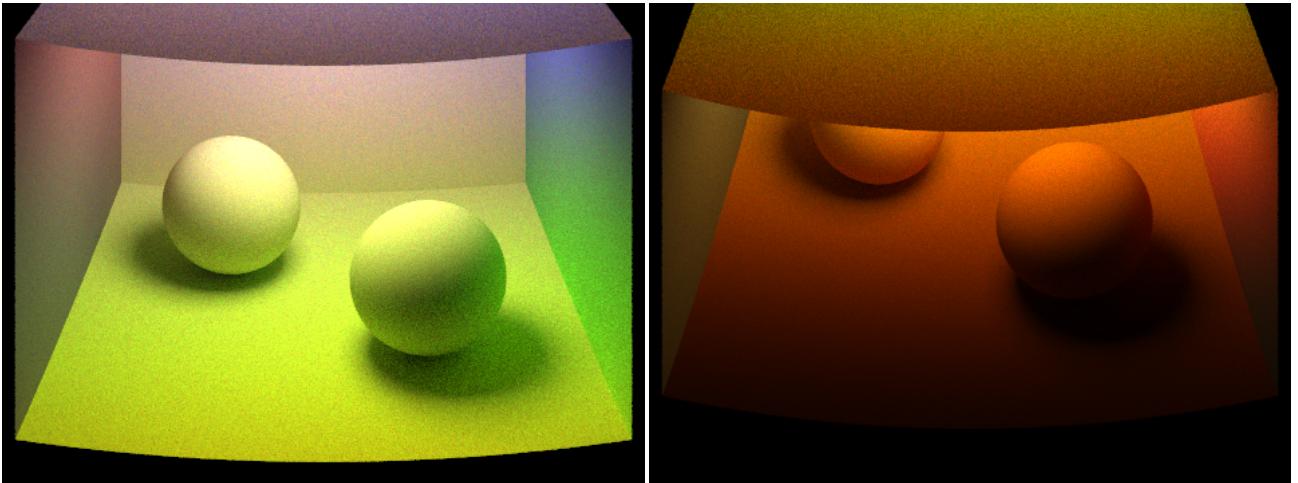
We now move on to demonstrate the relativistic Doppler shift.

Fig. 2-3:  $ct = 1 \vec{\beta} = (0, 0, -0.5)$ Fig. 2-4:  $ct = 1 \vec{\beta} = (0, 0, -0.4)$ Fig. 2-5:  $ct = 5 \vec{\beta} = (0, 0, -0.3)$ Fig. 2-6:  $ct = 5 \vec{\beta} = (0, 0, -0.2)$ Fig. 2-7:  $ct = 5 \vec{\beta} = (0, 0, -0.1)$ Fig. 2-8:  $ct = 6 \vec{\beta} = (0, 0, 0.1)$ Fig. 2-9:  $ct = 6 \vec{\beta} = (0, 0, 0.2)$ Fig. 2-10:  $ct = 6 \vec{\beta} = (0, 0, 0.3)$

Fig. 2-11:  $ct = 6 \vec{\beta} = (0, 0, 0.4)$ Fig. 2-12:  $ct = 6 \vec{\beta} = (0, 0, 0.5)$ 

As predicted by Special Relativity, the wavelength spectrum in the camera frame is red shifted if the source is moving away from the camera and blue shifted if the source is moving towards the camera. Since the emission spectrum of the ceiling light is a constant over all wavelength by design, it does not look any different in the camera frame regardless of the relative motion. As the relative speed becomes too large, the scene will look darker and darker because the BSDF we defined only scatters light close to the visible spectrum. Therefore, as the speed gets larger, the spectrum coming from the scene gets shifted farther away from the visible spectrum and the camera sensor will no longer be sensitive to those wavelengths. We further show some more images where both Lorentz contraction and Doppler shift are visible.

Fig. 2-13:  $ct = 5 \vec{\beta} = (0.25, 0, 0)$ Fig. 2-14:  $ct = 5 \vec{\beta} = (0.5, 0, 0)$ Fig. 2-15:  $ct = 5 \vec{\beta} = (0.75, 0, 0)$ Fig. 2-16:  $ct = 5 \vec{\beta} = (0, 0.25, 0)$

Fig. 2-17:  $ct = 5 \vec{\beta} = (0, 0.5, 0)$ Fig. 2-18:  $ct = 5 \vec{\beta} = (0, 0.75, 0)$ 

## Other corrections

One can incorporate additional corrections for added realism. We have not implemented these corrections as they will not be as visually striking, but we mention them here for completeness. Recall we used the following approximation in our derivation above:

$$\mathcal{E}'(x', \lambda') \approx \mathcal{E}(\Lambda^{-1}x, \lambda'/\alpha(\hat{k}')).$$

One can show from a simple argument based on photon counting, that the right hand side needs to be multiplied by a correction factor  $\eta$  to be exact, where

$$\eta = \delta/\alpha^2(\hat{k}')$$

$$\delta = \frac{d^2\vec{x}}{d^2\vec{x}'} \quad \text{ratio between differential sensor area as observed in lab and camera frame.}$$

The  $\alpha^2$  factor comes in due to an overall scaling of the transformed spectrum distribution which we ignored, along with an application of the Plank-Einstein relation  $E = \hbar\nu$ . The  $\delta$  factor comes in due to the fact that area is not invariant under Lorentz transformation so that the sensor can appear to have a different size in the lab frame.

## Contributions

Shikai Qiu

## References

- [1] [https://en.wikipedia.org/wiki/Lorentz\\_transformation](https://en.wikipedia.org/wiki/Lorentz_transformation)
- [2] [https://en.wikipedia.org/wiki/Spectral\\_sensitivity](https://en.wikipedia.org/wiki/Spectral_sensitivity)
- [3] <https://people.eecs.berkeley.edu/~cecilia77/graphics/a6/#>