

Refraction caustics in forward path tracing

Shawn Halayka*

Sunday 3rd September, 2023 18:17

Abstract

In this paper we introduce a method for producing refraction caustics in forward path tracing. These caustics do not rely on a light location, and as such, do not rely on bidirectional or backward path tracing. Thus, these caustics allow for as many light sources as one would care for; the shapes and positions of the lights are completely arbitrary (can be bunny shaped, etc).

1 Introduction

2 Code

A full Vulkan code exists [1], using a very simple (e.g. not physically accurate) light transport algorithm. The results given here can only get better with the improvement of the light transport algorithm. This code is based off of Sacha Willems' work [2, 3].

3 Acknowledgement

The Cornell boxes used in this paper were developed by Rob Rau.

*sjhalayka@gmail.com

```

vec3 hitPos = o + d * rayPayload.distance;

// If partially transparent
if(rayPayload.opacity != 1.0)
{
    // Incoming ray
    if(dot(d, rayPayload.normal) <= 0.0)
    {
        o = hitPos.xyz - rayPayload.normal * 0.01;
        d = refract(d, rayPayload.normal, eta);
    }
    else // Outgoing ray
    {
        vec3 temp_dir = refract(d, -rayPayload.normal, 1.0/eta);

        if(temp_dir != vec3(0.0))
        {
            o = hitPos.xyz + rayPayload.normal * 0.01;
            d = temp_dir;
        }
        else
        {
            // Total internal reflection
            o = hitPos.xyz - rayPayload.normal * 0.01;
            d = reflect(d, -rayPayload.normal);
        }
    }
}
else // Fully opaque
{
    o = hitPos + rayPayload.normal*0.01;
    vec3 d_cos = cos_weighted(rayPayload.normal, prng_state);
    d = mix(d_cos, reflect(d, rayPayload.normal), rayPayload.reflector);
}

```

Figure 1: Taking transparent objects into consideration. In essence, instead of always producing a pseudorandom cos-weighted reflection vector, refraction occurs for transparent objects.

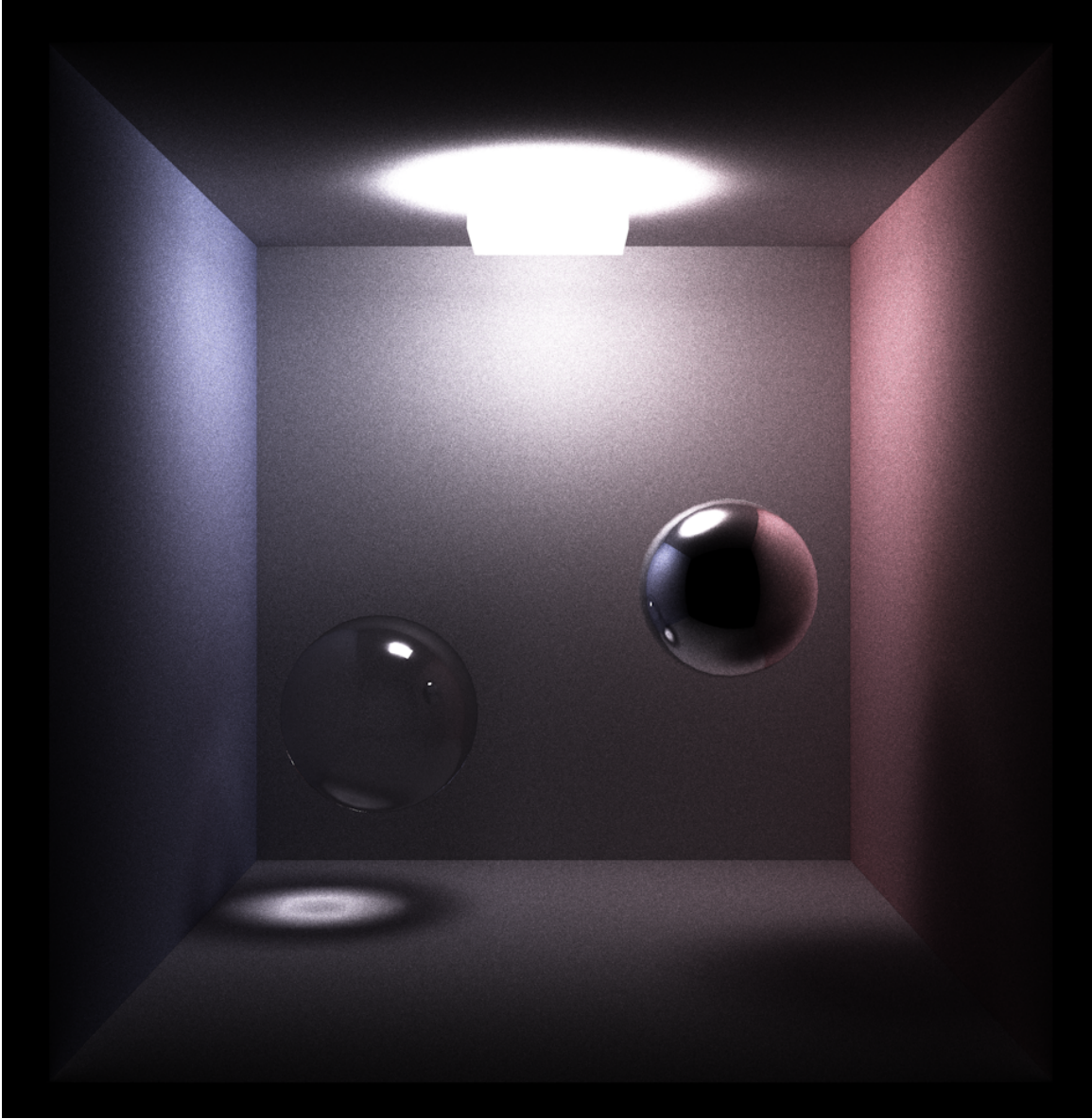


Figure 2: Refraction caustic, without chromatic aberration.

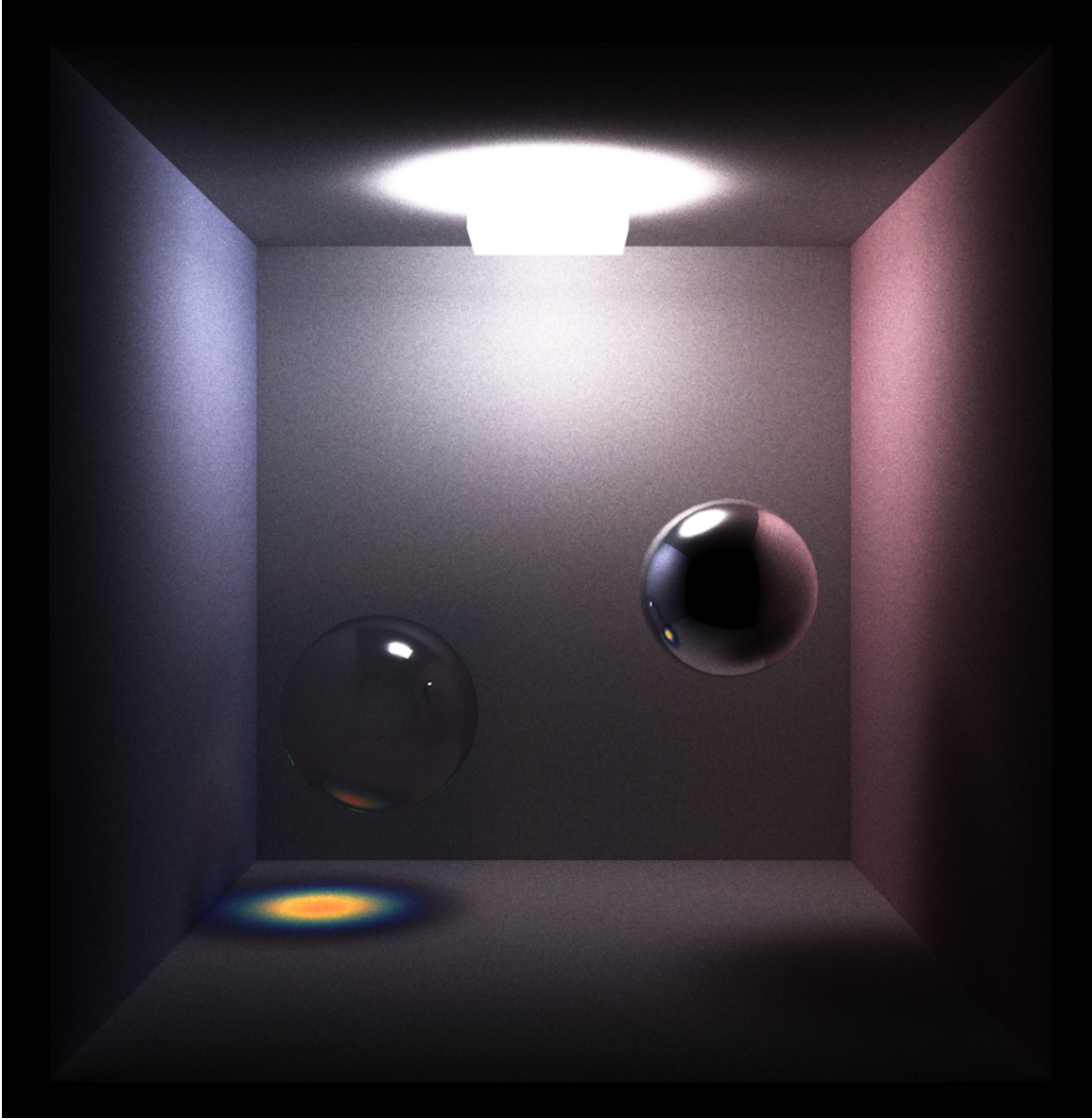


Figure 3: Refraction caustic, with chromatic aberration.

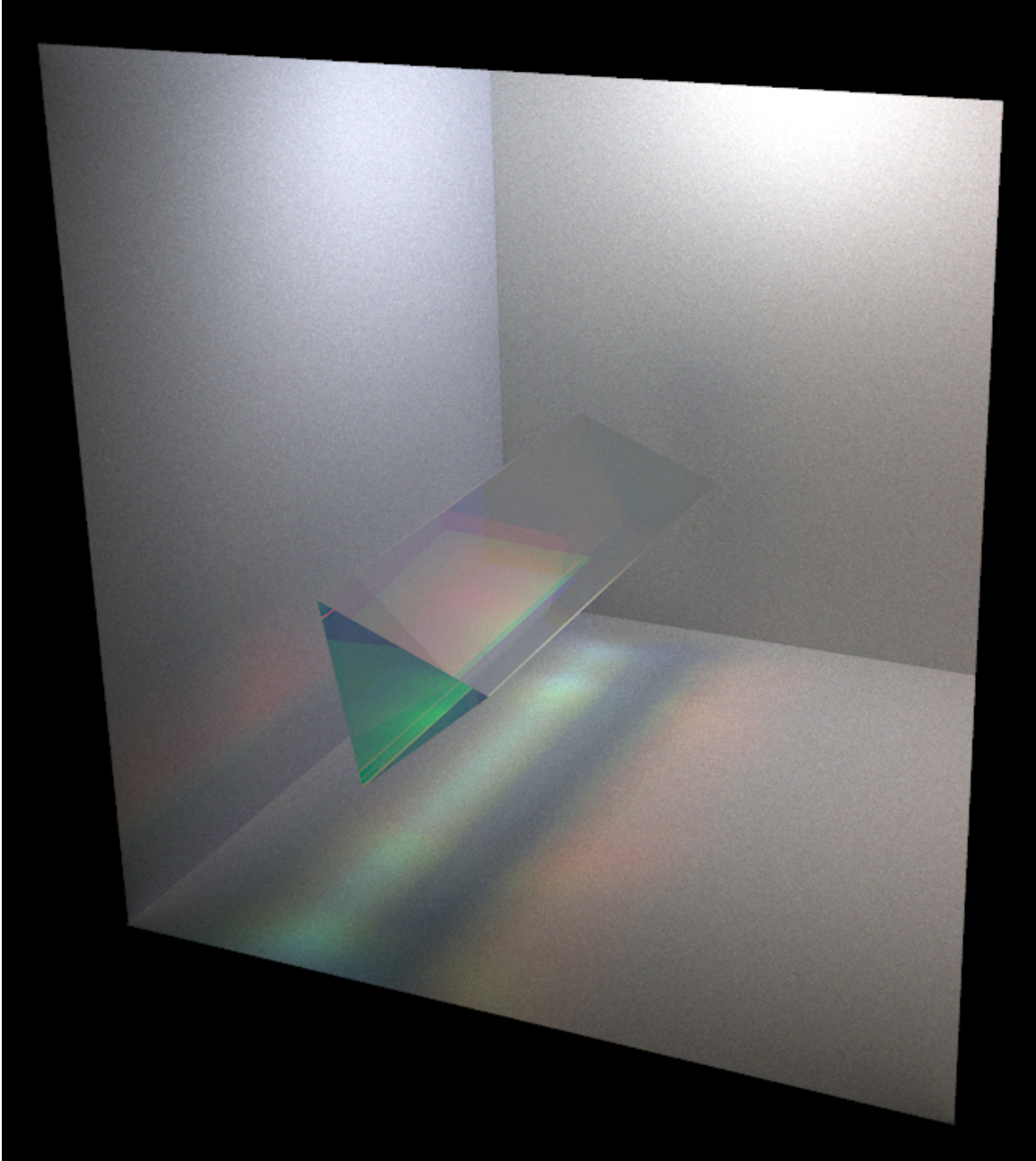


Figure 4: Refraction caustic, with chromatic aberration.

References

- [1] Halayka. Vulkan code. https://github.com/sjhalayka/cornell_box_textured
- [2] Willems. Path tracer code. <https://github.com/SaschaWillems/VulkanPathTracer>
- [3] Willems. Vulkan demo codes. <https://github.com/SaschaWillems/Vulkan>