

# HW6 Advanced Rendering

CS 148 Fall 2021-2022

Due Date: Monday, November 1st 2021 by 7 pm PT

Follow the instructions carefully. If you encounter any problems in the setup, please do not hesitate to reach out to CAs on Piazza or attend office hours on Nooks.

Be aware of the **Checkpoints** below. Make sure you complete each one since we will grade based on them.



The classroom scene and its various passes. Scene from [Blender demo files](#).

Continuing from HW4, in this homework we will explore the more advanced features of Blender's Cycles. Specifically, we will play with direct/indirect paths, motion blur, and volumetrics.

Download the .blend file for this homework:

[http://web.stanford.edu/class/cs148/assignments/hw6\\_advanced\\_rendering.blend](http://web.stanford.edu/class/cs148/assignments/hw6_advanced_rendering.blend)

## I. Render Passes



The images we have rendered so far are called the "beauty pass", or "combined pass", which is the final result of combining diffuse pass, specular pass, etc. In practice, saving out individual passes along with object masks (called Cryptomatte in Blender) can be useful when we need little tweaks on certain components in the scene without a full re-render, e.g. making certain objects less glossy, or reducing light through transmissive objects. This not only saves time when iterating your artwork but also gives you finer control over your render. Here we're visualizing each pass for the Cornell box scene, and you should reason about why each pass looks like it does. Read more in this Blender doc.

In Cycles, the passes are separated into light passes and data passes. We are mainly investigating light passes here. The passes are separated into two parts: (Diffuse | Glossy | Transmission) + (Direct | Indirect | Color). Here are the explanations:

- **Diffuse:** Lighting from diffuse and subsurface BSDFs. BSDF color is not included in this pass.
- **Glossy:** same as above, but for glossy (specular) BSDFs.
- **Transmission:** same as above, but for transmission BSDFs.
- **Direct:** Direct illumination. Lighting coming from light sources after a single reflection or transmission off a surface.
- **Indirect:** Indirect (global) illumination. Lighting coming from light sources after more than one reflection or transmission off a surface.

- **Color:** Color weights of the corresponding BSDFs.

By default, Blender uses this formula to obtain the combined pass:

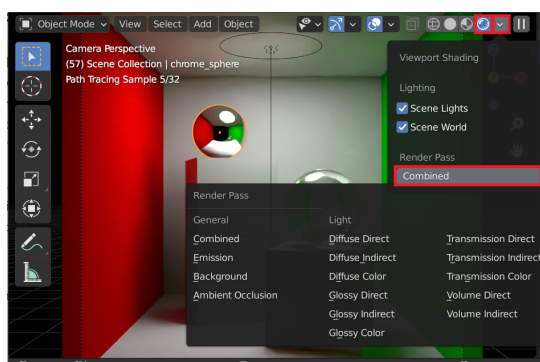
```
(Diffuse Direct + Diffuse Indirect) × Diffuse Color
+ (Glossy Direct + Glossy Indirect) × Glossy Color
+ (Transmission Direct + Transmission Indirect) × Transmission Color
+ Emission
+ Environment
```

The first few lines should look familiar to you at this point: In HW3 we implemented all the direct components and added them to get the final result, and in HW5 we implemented diffuse indirect and multiplied it by the diffuse color to get extra bounces for diffuse. Cycles, as a full-fledged render engine, has all the passes implemented. You'll be able to see what they look like in just a minute.

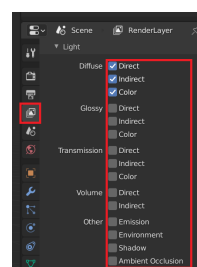
Open the downloaded scene.

To preview the passes, set your viewport shading to **Rendered**, and change the **Render Pass** from **Combined** to other options in the drop-down menu.

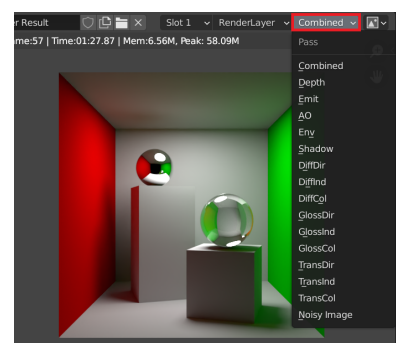
To actually render the passes, tick the boxes to enable corresponding passes in **View Layer Properties** ▶ **Light**. After you render the scene, you can view the passes you've enabled from the drop-down menu in the **Image Editor**.



Visualize passes in render preview.



Set passes for render.



See passes in render result.

- ☐ **Checkpoint 1: Save all the lighting passes for diffuse, glossy, and transmission (9 images in total), or show them with render preview during**

**grading. Explain why each pass looks like that.** Some of the passes could be totally black.

Change the preview **Render Pass** back to **Combined** before proceeding.

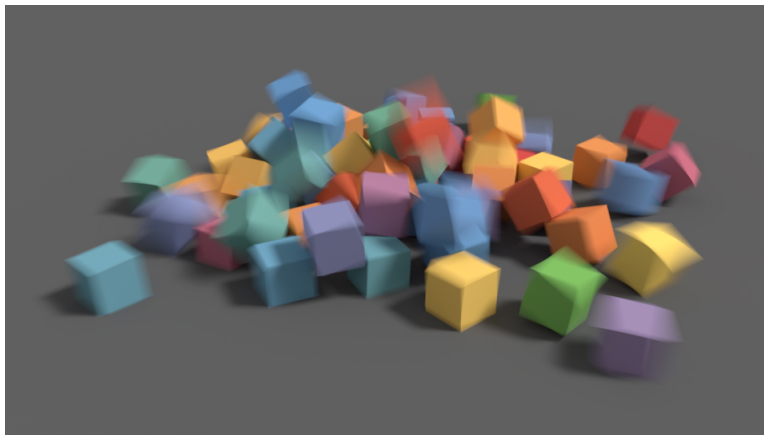


With the knowledge of passes, you can create non-physically-based materials with the Light Path node. In the Shader Editor, you can use the node to combine different materials by the kind of incoming ray the shader is being executed with. See an example here.

## II. Motion Blur



Real-world cameras need some exposure time to gather light information onto the film/sensor. When fast-moving objects appear in the frame, it will appear blurry in the direction of its movement. We can also simulate this with the render engine.

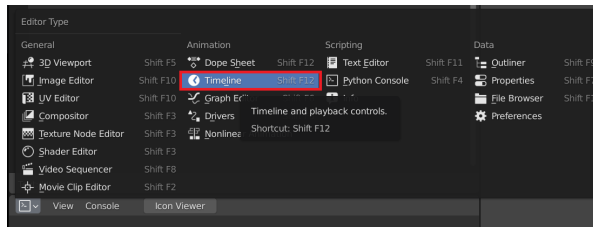


Motion blur. Source: [Blender Doc](#).

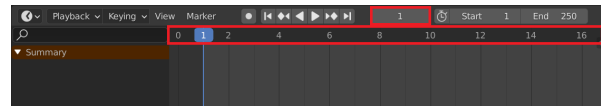
We will add motion to the small cube in the Cornell box, then turn on motion blur in Cycles. The first step is to animate the objects. If you're not familiar with animation or how Blender animation tools work, watch [this official tutorial](#) before proceeding.

Navigate to the **Timeline Editor** at the bottom. This is where we will be adding keyframes to define the motion of the cube.

The *Playhead* is the blue vertical line with the current frame number at the top. It can be set or moved to a new position by pressing or holding the left mouse button in the scrubbing area at the top of the timeline, or by entering a number in the **Current Frame** input box. More about the Timeline Editor in this [Blender Doc](#).



Timeline Editor



Two places to set the current frame.

We will animate the location of the small cube in two steps:

For the first keyframe, move the cube left and add a keyframe at frame 1:

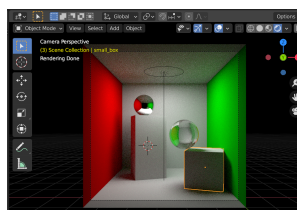
Select the small cube, move it along the X-axis by -1 (shortcut **G, X, -1**). Set the **current frame** to 1 from the Timeline Editor. Add a keyframe using one of two methods:

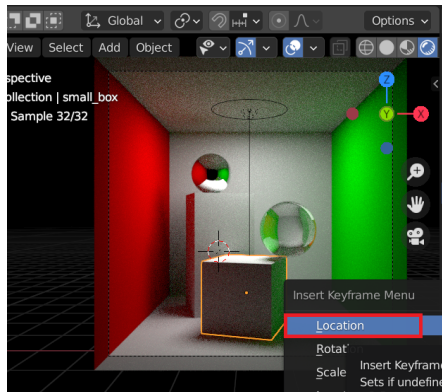
- 1) Press **I** in the 3D Viewport and select **Location**
- 2) **Right-click** on **Location X** and select **Insert Keyframes** in **Object Properties**.

Then move the cube right and add a keyframe at frame 3:

Select the small cube, move it along the X-axis by 2 (shortcut **G, X, 2**). Set the **current frame** to 3 from the Timeline Editor. Add a keyframe using the same methods explained above.

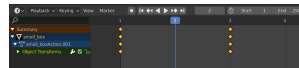
Now scrub through the timeline between frames 1 and 3, and you should see the cube moving.



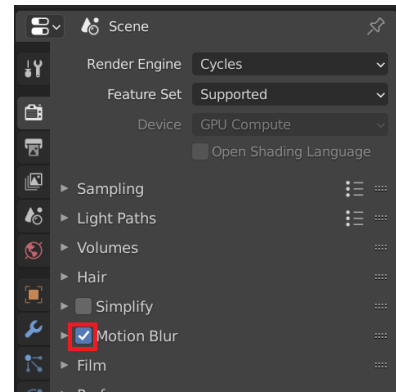


First keyframe.

Second keyframe.



Timeline editor with all the keyframes.



Turn on Cycle's motion blur.

Now turn on motion blur in **Render Properties**. Just tick the box! You can dig in and play with the settings. Read more in this [Blender doc](#).

Set your current frame to 2. Render the image (F12).

☐ **Checkpoint 2: save the image with motion blur. Discuss what you observe.**



You can move/scale the keyframes with the same key you move objects within the 3D viewport: **G** to move, **S** to scale.

## III. Depth of Field



Real-world cameras transmit light through a lens that bends and focuses it onto the sensor. Because of this, objects that are a certain distance away are in focus, but objects in front and behind that are blurred. Read more in this [Blender doc](#). This effect is implemented by Cycles, and we will test it out.

First, remove the animation of the small cube from the last step:

Select the small cube, press **F3**, and type **Clear Keyframes** to remove all the animations. You can also select all keyframes and delete them from the **Timeline Editor**.

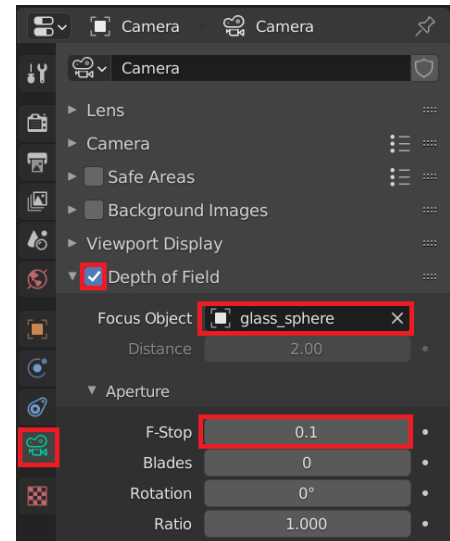
Then turn on depth of field for the camera:



Select the camera, check the box for Depth of Field from Object Data Properties. Set the Focus Object to `glass_sphere`, and F-Stop to 0.1.



Clear animation.



Set up depth of field.

☐ **Checkpoint 3: save the image with depth of field. Discuss what you observe.**



We are exaggerating the size of the aperture to better demonstrate the effect of depth of field. Common F-Stop values should be 1.4, 2.0, 2.8, 4.0, 5.6, 8.0 and so on.

## IV. Volumetric Absorption

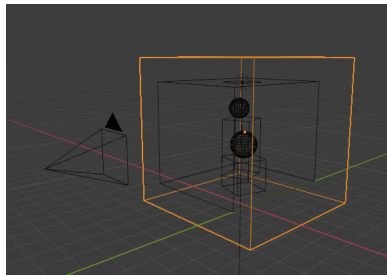


Volume rendering can achieve effects that cannot be represented by mesh surfaces alone. It is specifically designed to render volumetric objects like smoke, fire, fog, and clouds. Here we will fill the Cornell box with participating volume and see what the render looks like. Read more about volume rendering in this [Blender doc](#).

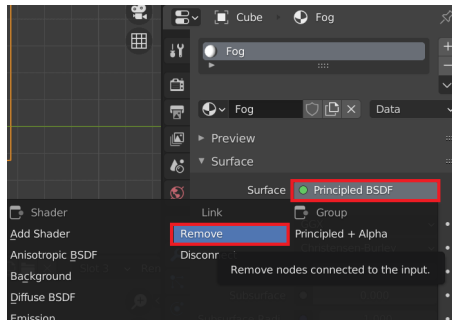
Add a cube to contain our scene, then create a volumetric material for the cube:

Add a cube, set its location to  $(-2.75, 2.75, 2.75)$  and scale to  $(4, 4, 4)$ .

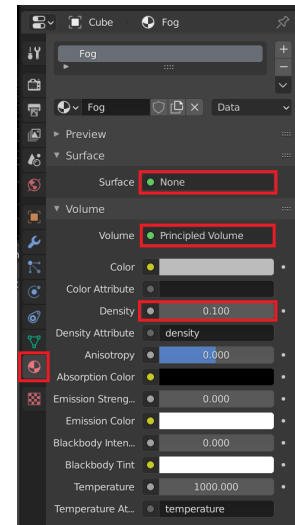
Add a material for the cube in **Material Properties**: remove the **Surface Shader** and add a **Principled Volume** shader for the **Volume** shader. Change the **Density** to 0.1.



Add a cube to surround the scene.



Remove surface shader.



Add volume shader.

- ☐ **Checkpoint 4: save the image with volumetric absorption. Discuss what you observe.** You can either keep or disable DoF from checkpoint 3.

## V. Grading (5 pts total)

This assignment will be graded on the following requirements

### Complete all the checkpoints (4 pts)

1. (1 pt) **Save all the lighting passes for diffuse, glossy, and transmission, or show them with render preview during grading. Explain why each passes look like that.**
2. (1 pt) **Save the image with motion blur. Discuss what you observe.**
3. (1 pt) **Save the image with depth of field. Discuss what you observe.**
4. (1 pt) **Save the image with volumetric absorption. Discuss what you observe.**



## Quiz Question (1 pt)

The TAs will choose one of the following questions during the grading session. Please be prepared to give a one-minute answer with your partner.

1. Describe the relationship between the camera frame rate and the rotation speed of a four-spoked wagon wheel which would cause a temporal aliasing effect of the wheel appearing to rotate slowly backwards.
2. Why should one pre-filter out high frequencies before sampling during ray tracing?
3. How does the box filter remove high frequency details of an image?
4. What is supersampling?
5. Describe the circle of confusion and depth of field. Why does a pinhole camera have infinite depth of field?
6. Explain why sunsets are red.
7. What is in-scattering and why is it used?
8. How can we implement volumetric emission, and what is an example of a real-life volumetric emissive light source?