CS 184 HW3.1 Report

Catherine Gai, Wenhan Sun

2022 Spring

Webpage Link

`https://github.com/cal-cs184-student/sp22-project-webpages-Wenhan0112/blob/master/proj3-1/index.html`

Introduction

In this homework, a global illumination scheme with adaptive sample is implemented. The core content are:

- Implementation of the determination of intersection between a ray and a primitive object. (Part 1)
  - The ray-triangle intersection is computed via Eq.1.
  - The ray-sphere intersection is computed via Eq.2.
- Recursive implementation of the bounding volume hierarchy. (Part 2)
  - Creation of bounding volume hierarchy establish the list of primitives for leaf nodes and the child for non-leaf node. The separation criterion is: by the axis of the longest edge of the bounding box, partition the set of primitives into halves by sorting the primitive according to the centroid coordinate on the axis.
  - **EXTRA CREDIT**: The list of points to primitives are not duplicated. Instead, the order of the primitives are rearranged so that each node corresponds to a continuous segment of primitives in the list.
  - The intersection test of the ray with the nodes is sped up as if the ray does not intersect the bounding box of the entire node, the ray does not intersect the descendants of the node.
- Implementation of the direct and global illumination (Part 3, 4).
  - The rendering equation for direct illumination with hemisphere sampling is included as Eq.3.

  - The rendering equation for direct illumination with lighting sampling is included in Eq.4.
  - The rendering equation for global illumination is included in Eq.5.

- Implementation of the adaptive sampling, which adaptive change the pixel sample rate; stop to sample if the sample value has already converged.

Part 1: Ray Generation and Scene Intersection

Task 1: Ray Generation

A ray is a combination of the following parameters:

- The starting point $\mathbf{o} \in \mathbb{R}^3$.
- The ray direction $\mathbf{d} \in \partial B(\mathbf{0}, 1) \subseteq \mathbb{R}^3$.
- The depth of the ray (used for global illumination and default to 0).
- The section of the ray such that is seen by the origin $t_{\min}$ and $t_{\max}$.

Every pixel point in the normalized framebuffer $(x, y) \in [0, 1]^2$ constitute to a ray starting from the origin $(0, 0, 0)$ directing in a way that passes through the focal plane at

$$\left((2x-1)\tan\left(\frac{F_h}{2}\right), (2y-1)\tan(\frac{F_v}{2}), -1\right)$$

where F_h and F_v are the horizontal and vertical field of viewing angles respectively. Therefore, the part of the focal plane that is saw by the camera is:

$$\left[-\tan\left(\frac{F_h}{2}\right), \tan\left(\frac{F_h}{2}\right)\right] \times \left[-\tan\left(\frac{F_v}{2}\right), \tan\left(\frac{F_v}{2}\right)\right]$$

Therefore, the ray is given by the following parametrization:

- The starting point $\mathbf{o} = (0, 0, 0)$.
- The ray direction $\mathbf{d} = \dfrac{\left((2x-1)\tan\left(\frac{F_h}{2}\right), (2y-1)\tan(\frac{F_v}{2}), -1\right)}{\|\left((2x-1)\tan\left(\frac{F_h}{2}\right), (2y-1)\tan(\frac{F_v}{2}), -1\right)\|}$
- The depth of the ray 0. (Since it goes directly to the camera).
- The section of the ray that is seen by the origin, `nClip` and `fClip`.

Task 2: Generating Pixel Samples

For each point $(x, y) \in [0, w] \times [0, h]$ in the unnormalized framebuffer with width w, height h, a number of random sample N_{aa} is drawn uniformly in the pixel region $(x, y) + [0, 1]^2$. The value of the pixel is given by the average of the sample of the pixels.

Task 3: Ray-Triangle Intersection

For a given ray with origin $\mathbf{o}$ and direction $\mathbf{d}$, and a triangle with vertices at $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, any point in the plane can be expressed in the barycentric coordinate of the triangle:

$$\{\alpha\mathbf{x}_1 + \beta\mathbf{x}_2 + \gamma\mathbf{x}_3 | \alpha, \beta, \gamma \in \mathbb{R}, \alpha + \beta + \gamma = 1\}$$

Therefore, suppose the ray intersection the plane of the triangle at parameter t and position of barycentric coordinate (α, β, γ). Then

$$\begin{cases} \alpha\mathbf{x}_1 + \beta\mathbf{x}_2 + \gamma\mathbf{x}_3 = \mathbf{o} + t\mathbf{d} \\ \alpha + \beta + \gamma = 1 \end{cases}$$

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & -\mathbf{d} \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t \end{bmatrix} = \begin{bmatrix} \mathbf{o} \\ 1 \end{bmatrix}$$

The determinant of the matrix is:

$$\det\left(\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & -\mathbf{d} \\ 1 & 1 & 1 & 0 \end{bmatrix}\right) = \mathbf{d} \cdot ((\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1))$$

Therefore, the matrix is invertible iff its determinant is not zero, that is, $\mathbf{d}$ is not parallel to the plane. In this case

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & -\mathbf{d} \\ 1 & 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{o} \\ 1 \end{bmatrix} \tag{1}$$

The intersection is valid iff $\alpha, \beta, \gamma > 0$ and t lies in its range.

The normal at the intersection point is directly given by the barycentric interpolation of the normal vectors on the vertices.

If an intersection is valid, then the maximum parameter value should be updated as the current intersection t value.

Task 4: Ray-Sphere Intersection

For a given ray with origin $\mathbf{o}$ and direction $\mathbf{d}$, and a sphere centered at $\mathbf{c}$ with radius r, for any point on the ray $\mathbf{o} + t\mathbf{d}$ such that it is on the sphere $\partial B(\mathbf{c}, r)$, it must satisfies:

$$\|\mathbf{o} + t\mathbf{d} - \mathbf{c}\|^2 = r^2$$

$$t^2\|\mathbf{d}\|^2 + 2t\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}) + \|\mathbf{o} - \mathbf{c}\|^2 - r^2 = 0$$

$$t^2 + 2t\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}) + \|\mathbf{o} - \mathbf{c}\|^2 - r^2 = 0 \tag{2}$$

This quadratic has a solution iff

$$0 \leq (2\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}))^2 - 4(\|\mathbf{o} - \mathbf{c}\|^2 - r^2) = 4((\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}))^2 - \|\mathbf{o} - \mathbf{c}\|^2 + r^2)$$

$$(\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}))^2 - \|\mathbf{o} - \mathbf{c}\|^2 + r^2 \geq 0$$

The solution is given by:

$$\{-\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}) - \sqrt{(\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}))^2 - \|\mathbf{o} - \mathbf{c}\|^2 + r^2}, -\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}) + \sqrt{(\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}))^2 - \|\mathbf{o} - \mathbf{c}\|^2 + r^2}\}$$

Suppose the solution exists and

$$\begin{cases} t_1 = -\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}) - \sqrt{(\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}))^2 - \|\mathbf{o} - \mathbf{c}\|^2 + r^2} \\ t_2 = -\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}) + \sqrt{(\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}))^2 - \|\mathbf{o} - \mathbf{c}\|^2 + r^2}\} \end{cases}$$

Then $t_1 \leq t_2$. The ray has a section between $t_{\max}$ and $t_{\min}$. The following situations are considered:

- $t_1 \leq t_2 \leq t_{\min} \leq t_{\max}$ (one comparison needed $t_2 \leq t_{\min}$): No intersection is found.
- $t_{\min} \leq t_{\max} \leq t_1 \leq t_2$ (one comparison needed $t_{\max} \leq t_1$): No intersection is found.
- $t_1 \leq t_{\min} \leq t_{\max} \leq t_2$: No intersection is found.
- $t_1 \leq t_{\min} \leq t_2 \leq t_{\max}$: Intersection is found at t_2.
- $t_{\min} \leq t_1 \leq t_2 \leq t_{\max}$: Intersection is found at t_1.
- $t_{\min} \leq t_1 \leq t_{\max} \leq t_2$: Intersection is found at t_1.

The normal at the intersection point $\mathbf{p}$ given by the method above is $\frac{\mathbf{p}-\mathbf{c}}{r}$.

Normal Shading Images

The rendering of of `sky/CBspheres.dae` with normal shading is included as Fig.1.
Following flags are used: `-r 480 360`

The rendering of of `sky/CBbunny.dae` with normal shading is included as Fig.2.
Following flags are used: `-r 480 360`

The rendering of of `sky/CBdragon.dae` with normal shading is included as Fig.3.
Following flags are used: `-r 480 360`

Part 2: Bounding Volume Hierarchy

Task 1: Constructing the BVH

A bounding volume Hierarchy is a tree structure that represents the partition of primitives in the scene. Each node of a tree represents a collection of primitives, which may have children nodes or may be a leaf.

If the node is not a leaf, then it contains a bounding box of the primitives it contains. Moreover, it contains pointers to the left and right child nodes. If the node is leaf, then it contains a bounding box as well as pointers to primitives.

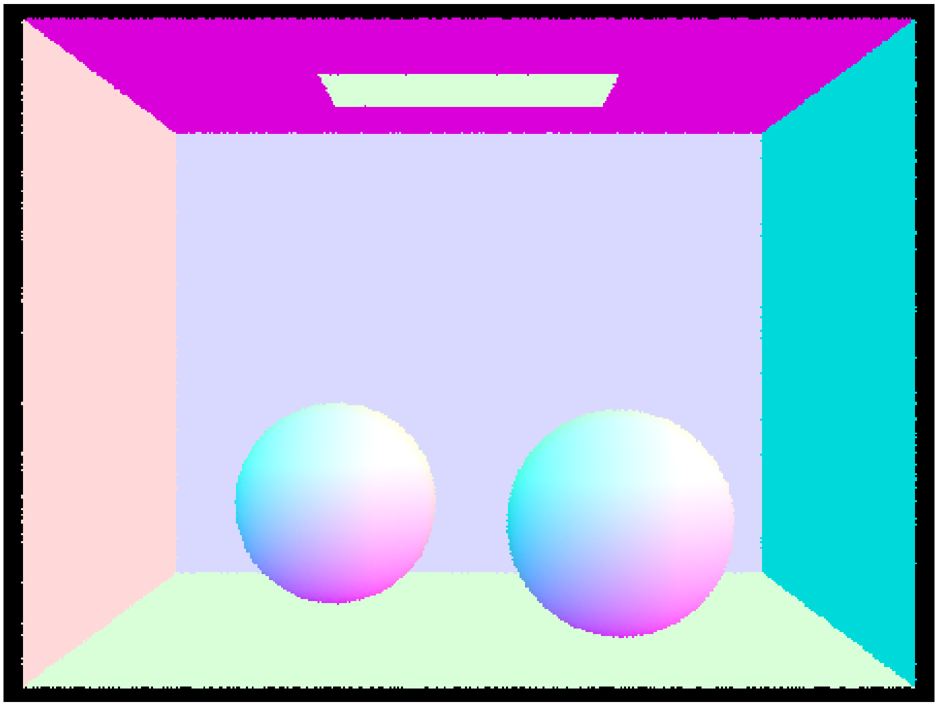


Figure 1: Rendering of `sky/CBspheres.dae` with normal shading and the following flags:
`-r 480 360`

The pseudocode for building a tree is given as below:

global : partition: $A \mapsto$ a partition of A
name : Create_BVH
input : A list of primitives: A
input : Maximum number of primitives in a leaf: $N_{\max}$
output: A bounding volume hierarchy node n_0
if $|A| \leq N_{\max}$ **then**
    $n_0 \leftarrow$ a leaf with elements A;
    Set the bounding box of n_0 according to the set A;
else
    $L, R \leftarrow$ partition(A);
    $n_l \leftarrow$ Create_BVH(L, $N_{\max}$);
    $n_r \leftarrow$ Create_BVH(R, $N_{\max}$);
    $n_0 \leftarrow$ an intermediate node with children n_l and n_r;
    Set the bounding box of n_0 according to the set A;
end

The partition function of the primitives in this implementation aims to produced a balanced tree. Therefore, the partition function is implemented as:

Choose a standard axis of $\mathbb{R}^3$ such that the span of primitives is the largest. Then sort the primitives by the coordinate of the centroid of the primitives on this axis. The primitives are then separated by the median of the coordinate values.

The pseudocode is given by:

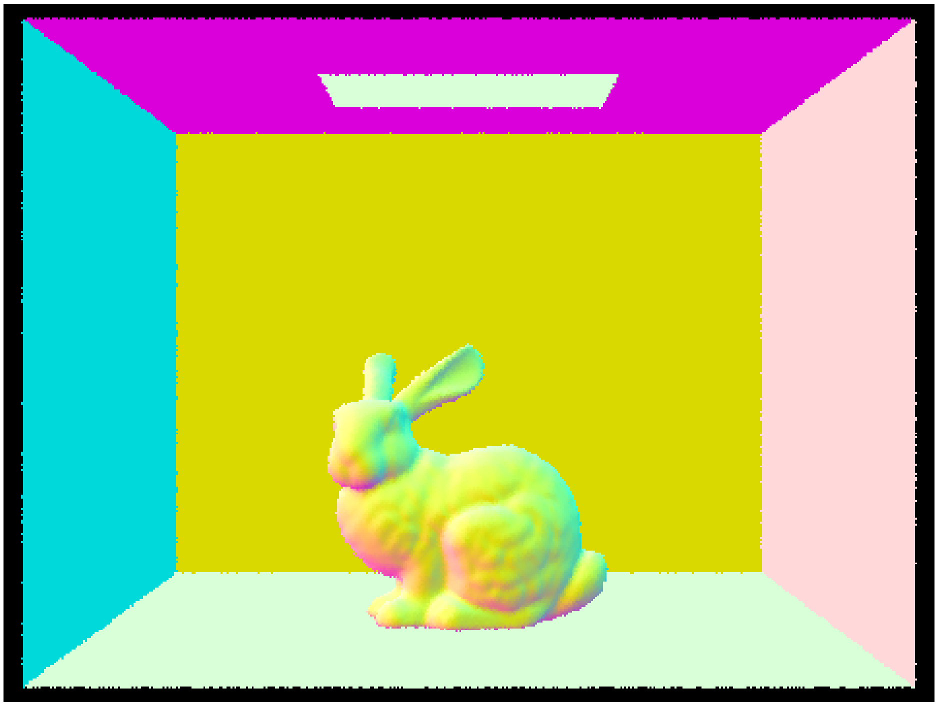


Figure 2: Rendering of `sky/CBbunny.dae` with normal shading and the following flags: `-r 480 360`

global : p: $(a, i) \mapsto$ the i-coordinate of centroid of primitive a
input : A set of primitives: A
output: A 2-partition of A: (B, C)
$[x_0, x_1] \times [y_0, y_1] \times [z_0, z_1] \leftarrow$ the bounding box of A;
$i \leftarrow \underset{i \in \{x,y,z\}}{\operatorname{argmax}}(|i_1 - i_0|)$;
$A = \{a_j | j \in \mathbb{N}_+(N)\}$ such that $\forall j, k \in \mathbb{N}_+(N)$ where $j > k$, $p(a_j, i) \geq p(a_k, i)$;
$B \leftarrow \left\{a_j \middle| j \in \mathbb{N}_+ \left(\text{int}\left(\frac{N}{2}\right)\right)\right\}$;
$C \leftarrow A \setminus B$

EXTRA CREDIT

The origin data structure keeps an array of pointers to primitives. Suppose the array is given by the sequence: $\{a_i | i \in \mathbb{N}_+(N)\}$ where $N \in \mathbb{N}_+$ is the number of primitive. The pointer is given initially by position 1 and N for the entire tree.

Suppose for each node, the primitives that a node contains is specified as $\{a_i | i \in \mathbb{Z}(i_l, i_r)\}$ where $i_l, i_r \in \mathbb{N}_+(N)$ and $i_l < i_r$ (this is the case in the base case where $i_l = 1$, $i_r = N$). Then the partition algorithm is simply the following:

Sort a view of the array $\{a_i | i \in \mathbb{Z}(i_l, i_r)\}$ in place, according to the comparator described above. ;
$i_m \leftarrow \text{int}\left(\frac{i_l + i_r}{2}\right)$;
$B \leftarrow \{a_i | i \in \mathbb{Z}(i_l, i_m)\}$ (This is done directly by passing in pointers to a_{i_l} and a_{i_m});
$C \leftarrow \{a_i | i \in \mathbb{Z}(i_m + 1, i_r)\}$ (This is done directly by passing in pointers to a_{i_m} and a_{i_r});

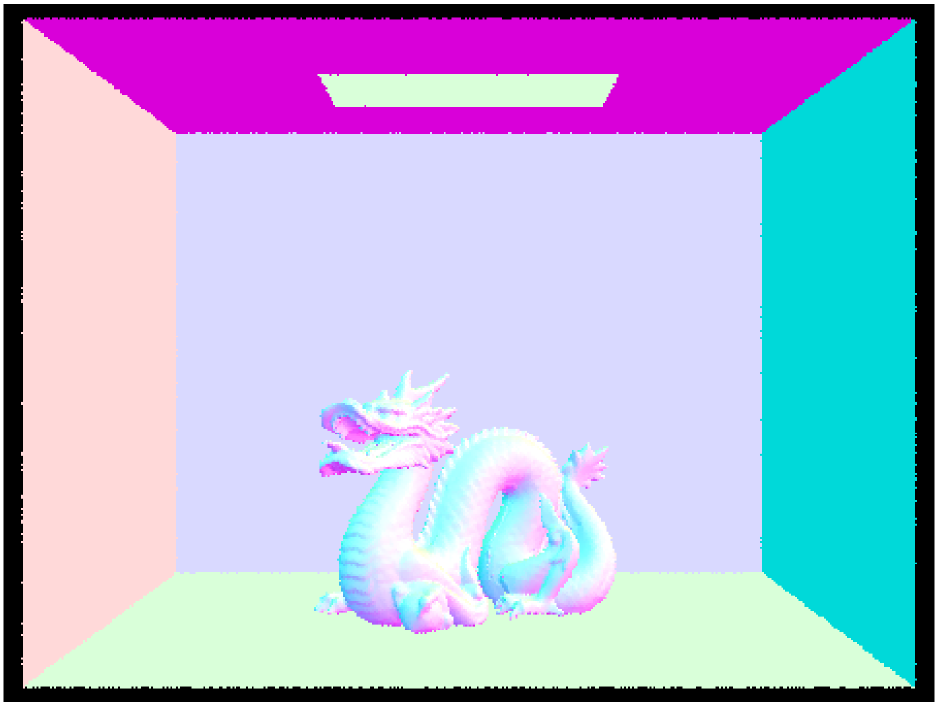


Figure 3: Rendering of `sky/CBdragon.dae` with normal shading and the following flags:
`-r 480 360`

Therefore, no part of the original array of pointers to primitive are replicated. Each node corresponds to a segments of an array. Each time a node is created, the array is sorted in place and separated into two smaller parts corresponding to the two children. Therefore, no additional memory is required to store the pointers to primitives.

Task 2: Intersecting the Bounding Box

Suppose a ray is given with starting point $\mathbf{o} = (o_x, o_y, o_z) \in \mathbb{R}^3$ and direction $\mathbf{d} = (d_x, d_y, d_z) \in \partial B(\mathbf{0}, 1) \subseteq \mathbb{R}^3$. Suppose a bounding box is given as $B = [x_0, x_1] \times [y_0, y_1] \times [z_0, z_1]$.

The set of all points in the ray with time range $R = (t_{\min}, t_{\max}) \subseteq \mathbb{R}_+$ is given by:

$$
\begin{aligned}
&\{\mathbf{p} = (p_x, p_y, p_z) = \mathbf{o} + t\mathbf{d} | t \in R, \mathbf{p} \in B = [x_0, x_1] \times [y_0, y_1] \times [z_0, z_1]\} \\
=&\{\mathbf{o} + t\mathbf{d} | t \in R, o_x + td_x \in [x_0, x_1], o_y + td_y \in [y_0, y_1], o_z + td_z \in [z_0, z_1]\} \\
=&\{\mathbf{o} + t\mathbf{d} | t \in R, td_x \in [x_0 - o_x, x_1 - o_x], td_y \in [y_0 - o_y, y_1 - o_y], td_z \in [z_0 - o_z, z_1 - o_z]\}
\end{aligned}
$$

The condition that the set is nonempty is equivalent to the following:

$\exists t \in R$ such that $\forall i \in \{x, y, z\}$, if $d_i = 0$, $0 \in [i_0 - o_i, i_1 - o_i]$, otherwise, $t \in \left[\frac{i_0 - o_i}{d_i}, \frac{i_1 - o_i}{d_i}\right]$

which is equivalent to the following:

Suppose $I = \{i \in \{x, y, z\} | d_i = 0\}$ and $\bar{I} = \{x, y, z\} \setminus I$. Then $\forall i \in I$, $0 \in [i_0 - o_i, i_1 - o_i]$. Moreover,

$$\bigcap_{i \in \bar{I}} \left[\frac{i_0 - o_i}{d_i}, \frac{i_1 - o_i}{d_i}\right] \neq \emptyset$$

which is equivalent to the following:

Suppose $I = \{i \in \{x, y, z\} | d_i = 0\}$ and $\bar{I} = \{x, y, z\} \setminus I$. Then $\forall i \in I$, $0 \in [i_0 - o_i, i_1 - o_i]$. Moreover,

$$\max_{i \in \bar{I}} \left(\min \left(\frac{i_0 - o_i}{d_i}, \frac{i_1 - o_i}{d_i} \right) \right) \leq \min_{i \in \bar{I}} \left(\max \left(\frac{i_0 - o_i}{d_i}, \frac{i_1 - o_i}{d_i} \right) \right)$$

Task 3: Intersecting the BVH

The intersection test of a ray to a bounding volume hierarchy is semantically a depth-first search where the branches could be pruned. That is, the depth-first search is searching a a node where the ray could not insect the bounding box of the node, the bounding boxes of the descendants of the node must be contained in the parent bounding box, so the ray will not intersect their bounding boxes as well. The search could be terminated on this node.

The algorithm is implemented as the following:

global : Bounding_Box_Test: (Ray r, Node n)$\mapsto$ True iff r intersects the bound box n
global : Intersect_Primitive: (Ray r, Primitive a)$\mapsto$ (Intersection point p, Range-modified ray r')
p is the intersection point between r and a. p is none if they do not intersect.
r' is the range-modified r based on the intersection.
name : Intersect_BVH
input : A ray with origin $\mathbf{o}$, direction $\mathbf{d}$ and time range R:
$r = (\mathbf{o}, \mathbf{d}, R) \in \mathbb{R}^3 \times \partial B(\mathbf{0}, 1) \times \mathcal{P}(\mathbb{R})$
input : A bounding volume hierarchy node n_0
output: The output points of the node or none: p_i
output: The ray object with modified range: $r = (\mathbf{o}, \mathbf{d}, R') \in \mathbb{R}^3 \times \partial B(\mathbf{0}, 1) \times \mathcal{P}(\mathbb{R})$

```
if Bounding_Box_Test(r, n0) then
    if n0 is a leaf node then
        foreach a in the list of primitives of n0 do
            (p, r) ←Intersect_Primitive(r, a);
            pi ← p if p is not none;
        end
    else
        (pi, r) ←Intersect_BVH(r, left child of n0);
        (p, r) ←Intersect_BVH(r, right child of n0);
        pi ← p if p is not none;
    end
end
```

Normal Shading Images

The rendering of `meshedit/cow.dae` with normal shading is included as Fig.4.
Following flags are used: `-t 8 -r 800 600`

The rendering of `meshedit/maxplanck.dae` with normal shading is included as Fig.5.
Following flags are used: `-t 8 -r 800 600`

The rendering of `meshedit/CBlucy.dae` with normal shading is included as Fig.6. Following flags are used: `-t 8 -r 800 600`

Figure 4: Rendering of `meshedit/cow.dae` with normal shading and the following flags: `-t 8 -r 800 600`

Timing Comparison

The table of the timing comparison between with or without bounding volume hierarchy is given as Tab.1. As is shown in the table, the time used to render without bounding volume hierarchy increases drastically with the complexity (number of primitives). However, the time used to render with bounding volume hierarchy remains approximately constant.

File	Number of primitives	t_0	t_{BVH}	Flags
`meshedit/cow.dae`	5856	43.0572s	0.0891s	`-t 8 -r 800 600`
`meshedit/maxplanck.dae`	50801	445.3300s	0.1094s	`-t 8 -r 800 600`
`sky/CBlucy.dae`	133796	$>$ 1000s	0.0943s	`-t 8 -r 800 600`

Table 1: t_0 is the rendering time without bounding volume hierarchy. t_{BVH} is the rendering time with bounding volume hierarchy. The device is: MacBook Pro (16-inch, 2019) with 2.3 GHz 8-Core Intel Core i9.

Figure 5: Rendering of `meshedit/maxplanck.dae` with normal shading and the following flags: `-t 8 -r 800 600`

Part 3: Direct Illumination

Task 1: Diffuse BSDF

A diffuse bidirectional scattering distribution function is defined as a bidirectional scattering distribution that is a constant on the outer hemisphere and zero in the inner hemisphere of $\partial B(\mathbf{0}, 1)$ with respect to the normal $\mathbf{n}$.

Suppose the bidirectional scattering distribution function is given by a constant f in the outer hemisphere H. The total incoming irradiance is given by:

$$\int_{\partial B(\mathbf{0},1)} L_i(\mathbf{e}_i)\mathbf{e}_i \cdot \mathbf{n} \, \mathrm{d}\mathbf{e}_i = \int_{H} L_i(\mathbf{e}_i)\mathbf{e}_i \cdot \mathbf{n} \, \mathrm{d}\mathbf{e}_i$$

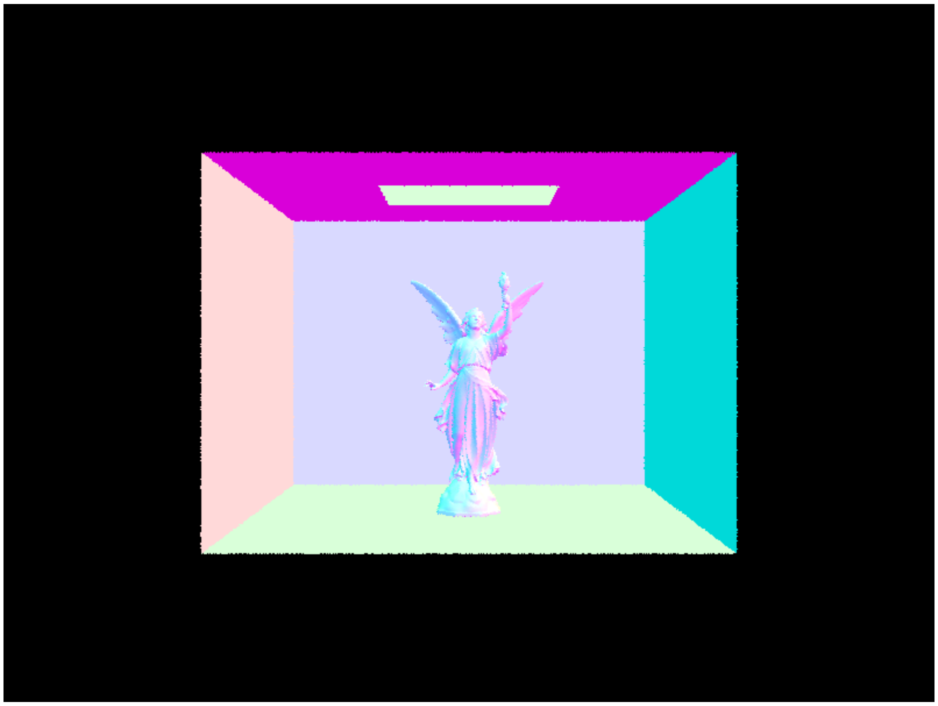


Figure 6: Rendering of `meshedit/CBlucy.dae` with normal shading and the following flags:
`-t 8 -r 800 600`

The total outgoing irradiance is given by:

$$\begin{aligned}
\int\limits_{\partial B(\mathbf{0},1)} L_o(\mathbf{e}_r)\mathbf{e}_r \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_r &= \int\limits_{\partial B(\mathbf{0},1)} \left(\int\limits_{\partial B(\mathbf{0},1)} f(\mathbf{e}_i, \mathbf{e}_r) L_i(\mathbf{e}_i)\mathbf{e}_i \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_i \right) \mathbf{e}_r \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_r \\
&= \int\limits_H \left(\int\limits_H f L_i(\mathbf{e}_i)\mathbf{e}_i \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_i \right) \mathbf{e}_r \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_r \\
&= f \left(\int\limits_H \mathbf{e}_r \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_r \right) \left(\int\limits_H L_i(\mathbf{e}_i)\mathbf{e}_i \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_i \right)
\end{aligned}$$

The total outgoing irradiance must be the total incoming irradiance multiplied by the reflectance α. Therefore,

$$f \left(\int\limits_H \mathbf{e}_r \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_r \right) \left(\int\limits_H L_i(\mathbf{e}_i)\mathbf{e}_i \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_i \right) = \alpha \int\limits_H L_i(\mathbf{e}_i)\mathbf{e}_i \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_i$$

$$\alpha = f \left(\int\limits_H \mathbf{e}_r \cdot \mathbf{n}\,\mathrm{d}\mathbf{e}_r \right) = f \int_{\theta=0}^{\frac{\pi}{2}} \int_{\phi=0}^{2\pi} \cos(\theta)\sin(\theta)\,\mathrm{d}\theta\,\mathrm{d}\phi = f\pi$$

$$f = \frac{\alpha}{\pi}$$

Therefore, the bidirectional scattering distribution function on a diffuse surface with reflectance α is given by:

$$f = \left((\mathbf{e}_i, \mathbf{e}_r) \mapsto \begin{cases} \dfrac{\alpha}{\pi} & , \mathbf{e}_i, \mathbf{e}_r \in H \\ 0 & , \text{otherwise} \end{cases} \right)$$

Note that in terms of the color (wavelength), this is interpreted as for a specific color (wavelength).

Task 2: Zero-bounce Illumination

Zero-bounce illumination is the ray that goes directly from the source to the camera with no change in the ray property. Therefore, given an intersection of the ray, the zero-bounce illumination is simply the emission radiance of the intersection point.

Task 3: Direct Lighting with Uniform Hemisphere Sampling

Direct lighting is the lighting scheme such that only zero-and-one-bounce illumination is considered. Therefore, the ray that is traced with only 0 or 1 change in the ray property.

Given a ray that intersect the scene at position p in direction $\mathbf{e}_0$, the received radiance is given by:

$$L_e(-\mathbf{e}_0) + \int\limits_{\partial B(\mathbf{0},1)} f(\mathbf{e}_i, -\mathbf{e}_0) L_i(\mathbf{e}_i) \mathbf{e}_i \cdot \mathbf{n} \, \mathrm{d}\mathbf{e}_i$$

where L_e is a function that maps a direction to the emitted radiance of the point, L_i is a function that maps a direction to the incoming radiance of the point, $\mathbf{n}$ is the outward surface normal, and f is the bidirectional scattering distribution function at the point. Moreover, L_i is the zero-bounce illumination of the point since this is the one-bounce illumination term in direct illumination.

Suppose f is non-zero only when evaluated in $H \times H$ where H is the outer hemisphere. Then the received radiance is given by:

$$L_e(-\mathbf{e}_0) + \int\limits_{H} f(\mathbf{e}_i, -\mathbf{e}_0) L_i(\mathbf{e}_i) \mathbf{e}_i \cdot \mathbf{n} \, \mathrm{d}\mathbf{e}_i$$

Suppose $\{\mathbf{E}_j | j \in \mathbb{N}_+(N)\}$ is a sequence of independent identically distributed random variable in the outer hemisphere H with probability density function p such that p is nonzero in H. Then by Monte-Carlo integration, the received radiance is given by the expected value of the following random variable:

$$L_e(-\mathbf{e}_0) + \frac{1}{N} \sum_{i=1}^{N} \frac{f(\mathbf{E}_j, -\mathbf{e}_0) L_i(\mathbf{E}_j) \mathbf{E}_j \cdot \mathbf{n}}{p(\mathbf{E}_j)} \tag{3}$$

Therefore, this is the estimator for the direct illumination in the direction $\mathbf{e}_0$.

In the uniform sampling scheme, p is a constant function, which is given by a constant of:

$$p = \frac{1}{\displaystyle\int\limits_H \mathrm{d}\mathbf{e}} = \frac{1}{2\pi}$$

Task 4: Direct Lighting by Importance Sampling Lights

In direct lighting scheme, the received radiance as in the previous task is given by:

$$L_e(\mathbf{p}, -\mathbf{e}_0) + \int_H f(\mathbf{e}_i, -\mathbf{e}_0) L_i(\mathbf{e}_i) \mathbf{e}_i \cdot \mathbf{n} \, \mathrm{d}\mathbf{e}_i$$

where L_i is the zero-bounce illumination of the point.

Suppose there are multiple lights in the scene S_l. Suppose S_{lp} and S_{le} is a partition of S_l such that S_{lp} is the set of point lights and S_{le} is the set of extended light.

For a point light source $s_{lp} \in S_{lp}$, the radiance of point light source at point $\mathbf{p}_{s_{lp}}$ on a point $\mathbf{p} \in \mathbb{R}^3$ can be given by $\frac{E_{s_{lp}}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|^2}\delta\left(\frac{\mathbf{p}-\mathbf{p}_{s_{lp}}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|}\right)$, where $E_{s_{lp}} \in \mathbb{R}$. Therefore, the one-bounce illumination contribution of the point light source is given by;

$$f\left(\frac{\mathbf{p}-\mathbf{p}_{s_{lp}}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|}, -\mathbf{e}_0\right) V(\mathbf{p}, \mathbf{p}_{s_{lp}}) \frac{E_{s_{lp}}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|^2} \frac{\mathbf{p}_{s_{lp}}-\mathbf{p}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|} \cdot \mathbf{n}$$

where V is the binary visibility function of the scene.

For an extended light source $s_{le} \in S_{le}$, suppose the set of all points on the light source is $A_{s_{le}}$. Then the radiance from point $\mathbf{p}_0 \in A_{s_{le}}$ to $\mathbf{p}$ is $L_e\left(\mathbf{p}_0, \frac{\mathbf{p}-\mathbf{p}_0}{\|\mathbf{p}-\mathbf{p}_0\|}\right)$. Therefore, the one-bounce illumination contribution of the extended light source is given by:

$$\int_{A_{s_{lp}}} f\left(\frac{\mathbf{p}-\mathbf{p}_0}{\|\mathbf{p}-\mathbf{p}_0\|}, -\mathbf{e}_0\right) V(\mathbf{p}, \mathbf{p}_0) L_e\left(\mathbf{p}_0, \frac{\mathbf{p}-\mathbf{p}_0}{\|\mathbf{p}-\mathbf{p}_0\|}\right) \frac{(\mathbf{p}_0-\mathbf{p})\cdot\mathbf{n}}{\|\mathbf{p}-\mathbf{p}_0\|^4} (\mathbf{p}-\mathbf{p}_0)\cdot \, \mathrm{d}\mathbf{S}(\mathbf{p}_0)$$

where V is the binary visibility function of the scene.

Suppose $\mathbf{n}_{s_{le}}$ maps a point in $A_{s_{le}}$ to its unit outward normal. Then the integral is given as:

$$\int_{A_{s_{lp}}} f\left(\frac{\mathbf{p}-\mathbf{p}_0}{\|\mathbf{p}-\mathbf{p}_0\|}, -\mathbf{e}_0\right) V(\mathbf{p}, \mathbf{p}_0) L_e\left(\mathbf{p}_0, \frac{\mathbf{p}-\mathbf{p}_0}{\|\mathbf{p}-\mathbf{p}_0\|}\right) \frac{((\mathbf{p}_0-\mathbf{p})\cdot\mathbf{n})((\mathbf{p}-\mathbf{p}_0)\cdot\mathbf{n}_{s_{le}}(\mathbf{p}_0))}{\|\mathbf{p}-\mathbf{p}_0\|^4} \, \mathrm{d}^2\mathbf{p}_0$$

Suppose $\{\mathbf{P}_{j,s_{le}} | j \in \mathbb{N}_+(N_{s_{le}})\}$ is a sequence of independent identically distributed random variable in $A_{s_{le}}$ with probability density function p such that p is nonzero in $\{\mathbf{p_0} \in A_{s_{le}} | V(\mathbf{p}, \mathbf{p}_0)\}$. Then by Monte-Carlo integration, the one-bounce illumination contribution of the extended light source is given by the expected value of the following random variable:

$$\frac{1}{N_{s_{le}}} \sum_{i=1}^{N_{s_{le}}} f\left(\mathbf{E}_{j,s_{le}}, -\mathbf{e}_0\right) V(\mathbf{p}, \mathbf{P}_{j,s_{le}}) L_e\left(\mathbf{P}_{j,s_{le}}, \mathbf{E}_{j,s_{le}}\right) \frac{(-\mathbf{E}_{j,s_{le}}\cdot\mathbf{n})(\mathbf{E}_{j,s_{le}}\cdot\mathbf{n}_{s_{le}}(\mathbf{P}_{j,s_{le}}))}{\|\mathbf{p}-\mathbf{P}_{j,s_{le}}\|^2}$$

where $\forall j \in \mathbb{N}_+(N_{s_{le}})$, $\mathbf{E}_{j,s_{le}} = \dfrac{\mathbf{p}-\mathbf{P}_{j,s_{le}}}{\|\mathbf{p}-\mathbf{P}_{j,s_{le}}\|}$

Therefore, the total direct illumination radiance is given by:

$$\begin{aligned}
&L_e(\mathbf{p}, -\mathbf{e}_0) \\
&+ \sum_{s_{lp}\in S_{lp}} f\left(\frac{\mathbf{p}-\mathbf{p}_{s_{lp}}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|}, -\mathbf{e}_0\right) V(\mathbf{p}, \mathbf{p}_{s_{lp}}) \frac{E_{s_{lp}}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|^2} \frac{\mathbf{p}_{s_{lp}}-\mathbf{p}}{\|\mathbf{p}-\mathbf{p}_{s_{lp}}\|} \cdot \mathbf{n} \\
&+ \sum_{s_{le}\in S_{le}} \frac{1}{N_{s_{le}}} \sum_{i=1}^{N_{s_{le}}} f\left(\mathbf{E}_{j,s_{le}}, -\mathbf{e}_0\right) V(\mathbf{p}, \mathbf{P}_{j,s_{le}}) L_e\left(\mathbf{P}_{j,s_{le}}, \mathbf{E}_{j,s_{le}}\right) \frac{(-\mathbf{E}_{j,s_{le}}\cdot\mathbf{n})(\mathbf{E}_{j,s_{le}}\cdot\mathbf{n}_{s_{le}}(\mathbf{P}_{j,s_{le}}))}{\|\mathbf{p}-\mathbf{P}_{j,s_{le}}\|^2}
\end{aligned} \tag{4}$$

where $\forall j \in \mathbb{N}_+(N_{s_{le}})$, $\mathbf{E}_{j,s_{le}} = \dfrac{\mathbf{p} - \mathbf{P}_{j,s_{le}}}{\|\mathbf{p} - \mathbf{P}_{j,s_{le}}\|}$

Example Images

The renderings of `sky/CBbunny.dae` with direct illumination and low sampling rate using hemisphere and lighting sampling are included as Fig.7.

Following flags are used:

Hemisphere sampling: `-t 8 -s 1 -l 1 -m 1 -r 480 360 -H`
Lighting sampling: `-t 8 -s 1 -l 1 -m 1 -r 480 360`

The renderings of `sky/CBbunny.dae` with direct illumination and high sampling rate using hemisphere and lighting sampling are included as Fig.8.

Following flags are used:

Hemisphere sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 360 -H`
Lighting sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 360`

The renderings of `sky/dragon.dae` with direct illumination and high sampling rate using hemisphere and lighting sampling are included as Fig.9. The hemisphere sampling return all black because there are only point light sources in the scene. In hemisphere sampling, the probability is almost zero in sampling a point in the hemisphere.

Following flags are used:

Hemisphere sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 480 -H`
Lighting sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 360`

Quality VS Number of Samples

The renderings of `sky/CBbunny.dae` with direct illumination using lighting sampling of various lighting sample rates are included as Fig.10.

Following flags are used:

Lighting sample rate 1: `-t 8 -s 1 -l 1 -m 6 -r 480 360`
Lighting sample rate 4: `-t 8 -s 1 -l 4 -m 6 -r 480 360`
Lighting sample rate 16: `-t 8 -s 1 -l 16 -m 6 -r 480 360`
Lighting sample rate 64: `-t 8 -s 1 -l 64 -m 6 -r 480 360`

Comparison

Hemisphere sampling samples all directions in the hemisphere but lighting sampling samples only the direction from the lights.

When randomly sampling in the hemisphere, some of the samples will not point to any light source. Therefore, by concentrating the probability density function of the Monte-Carlo estimator into the portion of the sample space where the lighting contributes the most, the variance will decrease, leading to a smoother rendering of image. Moreover, lighting sampling makes point light source possible, since the probability of sampling a point light source in hemisphere sampling is almost zero. This smoothness is clearly demonstrated in Fig.7, Fig.8, and Fig.9.

Part 4: Global Illumination

Task 1: Sampling with Diffuse BSDF

Suppose X is a random variable on the sample space S with probability density function p. Then the function implemented returns a sample of X, x, in S as the outer hemisphere $\partial B(\mathbf{0},1)\cap(\mathbb{R}^2\times\mathbb{R}_+)$, the probability density function evaluated at x, $p(x)$, and the bidirectional scattering distribution function evaluated at the input incoming direction and outgoing direction x.

Task 2: Global Illumination

In this task the global illumination is implemented. Suppose L_e is the global emission radiance function, L_i is the global incoming radiance function, L_o is the global outgoing radiance function, f is the global bidirectional scattering distribution function, t is the transport function, and $\mathbf{n}$ is the unit outward normal field of the scene. Then the rendering equation is given as: $\forall p$ in the scene, $\forall \mathbf{e}\in\partial B(\mathbf{0},1)$,

$$L_o(\mathbf{p},\mathbf{e}) = L_e(\mathbf{p},\mathbf{e}) + \int\limits_{\partial B(\mathbf{0},1)} f(\mathbf{p},\mathbf{e}',\mathbf{e})L_o(t(\mathbf{p},\mathbf{e}'),-\mathbf{e}')\mathbf{e}'\cdot\mathbf{n}(\mathbf{p})\,\mathrm{d}^2\mathbf{e}'$$

Suppose $\{\mathbf{E}'_i|i\in\mathbb{N}_+(N)\}$ where $N\in\mathbb{N}_+$ is a sequence of independent, identically distributed random variable in $\partial B(\mathbf{0},1)$ with probability density function p_0. Then by Monte-Carlo integration:

$$L_o(\mathbf{p},\mathbf{e}) = L_e(\mathbf{p},\mathbf{e}) + \mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{p_0(\mathbf{E}'_i)}f(\mathbf{p},\mathbf{E}'_i,\mathbf{e})L_o(t(\mathbf{p},\mathbf{e}'),-\mathbf{E}'_i)\mathbf{E}'_i\cdot\mathbf{n}(\mathbf{p})\right)$$

However, this is still a recursive equation of L_o. To resolve this, suppose $\mathbb{1}$ is a random variable of Bernoulli distribution $p_t\in(0,1]$, independent of $\{\mathbf{E}'_i|i\in\mathbb{N}_+(N)\}$. Then the following equation holds:

$$\begin{aligned}L_o(\mathbf{p},\mathbf{e}) &= L_e(\mathbf{p},\mathbf{e}) + \frac{\mathbb{E}(\mathbb{1})}{p_t}\mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{p_0(\mathbf{E}'_i)}f(\mathbf{p},\mathbf{E}'_i,\mathbf{e})L_o(t(\mathbf{p},\mathbf{e}'),-\mathbf{E}'_i)\mathbf{E}'_i\cdot\mathbf{n}(\mathbf{p})\right)\\ &= L_e(\mathbf{p},\mathbf{e}) + \mathbb{E}\left(\frac{\mathbb{1}}{p_tN}\sum_{i=1}^{N}\frac{}{p_0(\mathbf{E}'_i)}f(\mathbf{p},\mathbf{E}'_i,\mathbf{e})L_o(t(\mathbf{p},\mathbf{e}'),-\mathbf{E}'_i)\mathbf{E}'_i\cdot\mathbf{n}(\mathbf{p})\right)\end{aligned} \tag{5}$$

Therefore, whenever $\mathbb{1}$ is sampled as 0, the sum does not needs to be evaluated. Therefore, the recursive is terminated. p_t is dynamically set in different recursion depth. In the implementation:

$$p_t = \begin{cases}0.4 & ,\text{depth}\geq 2\\ 1 & ,\text{depth}=1\end{cases}$$

Direct, Indirect, Global Illumination Comparison

The rendering of `sky/CBspheres_lambertian.dae` with global illumination using lighting sampling of pixel sample rate 1024 is included as Fig.11.

Following flags are used: `-t 8 -s 1024 -l 16 -m 5 -r 480 360`

The renderings of `sky/CBspheres_lambertian.dae` with direct and indirect illumination using lighting sampling of pixel sample rate 1024 are included as Fig.12.

Following flags are used: `-t 8 -s 1024 -l 16 -m 5 -r 480 360`

Quality VS Maximum Ray Depth

The renderings of `sky/CBbunny.dae` with global illumination using lighting sampling of various maximum ray depth are included as Fig.13. The pixel sample rate is 1024.

Following flags are used:

Maximum ray depth 1: `-t 8 -s 1024 -l 16 -m 1 -r 480 360`
Maximum ray depth 2: `-t 8 -s 1024 -l 16 -m 2 -r 480 360`
Maximum ray depth 3: `-t 8 -s 1024 -l 16 -m 3 -r 480 360`
Maximum ray depth 100: `-t 8 -s 1024 -l 16 -m 100 -r 480 360`

Quality VS Pixel Sample Rate

The renderings of `sky/CBspheres_lambertian.dae` with global illumination using lighting sampling of various pixel sample rates are included as Fig.14. The lighting sample rate is 4.

Following flags are used:

Pixel sample rate 1: `-t 8 -s 1 -l 4 -m 5 -r 480 360`
Pixel sample rate 2: `-t 8 -s 2 -l 4 -m 5 -r 480 360`
Pixel sample rate 4: `-t 8 -s 4 -l 4 -m 5 -r 480 360`
Pixel sample rate 8: `-t 8 -s 8 -l 4 -m 5 -r 480 360`
Pixel sample rate 16: `-t 8 -s 16 -l 4 -m 5 -r 480 360`
Pixel sample rate 64: `-t 8 -s 64 -l 4 -m 5 -r 480 360`
Pixel sample rate 1024: `-t 8 -s 1024 -l 4 -m 5 -r 480 360`

Part 5: Adaptive Sampling

Task 1: Adaptive Sampling

In this part, a method of finding out whether the pixel sampling converges is implemented.

Given a sequence of N samples $\{s_i | i \in \mathbb{N}_+(N)\}$, the empirical mean (that is, the expectation of the empirical mean estimator) is given by:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} s_i$$

The empirical standard error (that is, the standard deviation of the empirical mean estimator) is given by:

$$\sigma = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}(s_i - \mu)^2}$$

Assuming normal distribution for the samples, suppose a confidence interval of $i\%$ is desired within a maximum relative tolerance of δ. Then

$$p\sigma \leq \delta|\mu|$$

where p is the p-score of the confidence interval. If this is the case, then it is believed that the samples have already converged, so μ is adopted as the sample value.

In the implementation $i = 95$ and $p \approx 1.96$.

Therefore, an algorithm could be implemented as follows:

input : Sample function f;
input : Maximum number of samples $N_{\max}$;
input : Number of samples between test N_0;
input : p-score $p = 1.96$;
input : Maximum relative tolerance δ;
output: Sample value s_1;
$n \leftarrow 1$;
$s_1 \leftarrow f(n)$;
$s_2 \leftarrow s_1^2$;
while $n < N_{\max}$ **do**
    if $i \mod N_0 = 0$ **then**
        $\mu = \frac{s_1}{n}$;
        $\sigma = \sqrt{\frac{1}{n(n-1)}\left(s_2 - \frac{s_1^2}{n}\right)}$;
        if $p\sigma \leq \delta|\mu|$ **then**
            break;
        end
    end
    $n \leftarrow n + 1$;
    $s_0 \leftarrow f(n)$;
    $s_1 \leftarrow s_1 + s_0$;
    $s_2 \leftarrow s_2 + s_0^2$;
end
$s_1 = \frac{s_1}{n}$

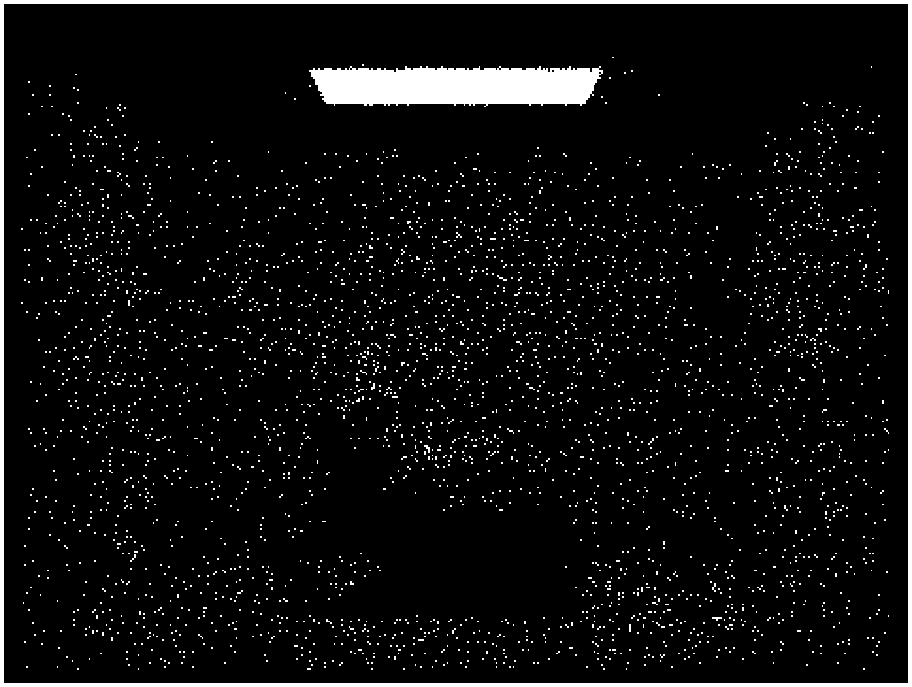
Adaptive Sampling Image

The rendering of `sky/CBbunny.dae` with global illumination using adaptive lighting sampling is included as Fig.15. The sample rate buffer is also included in Fig.15. The maximum relative tolerance is 0.05.

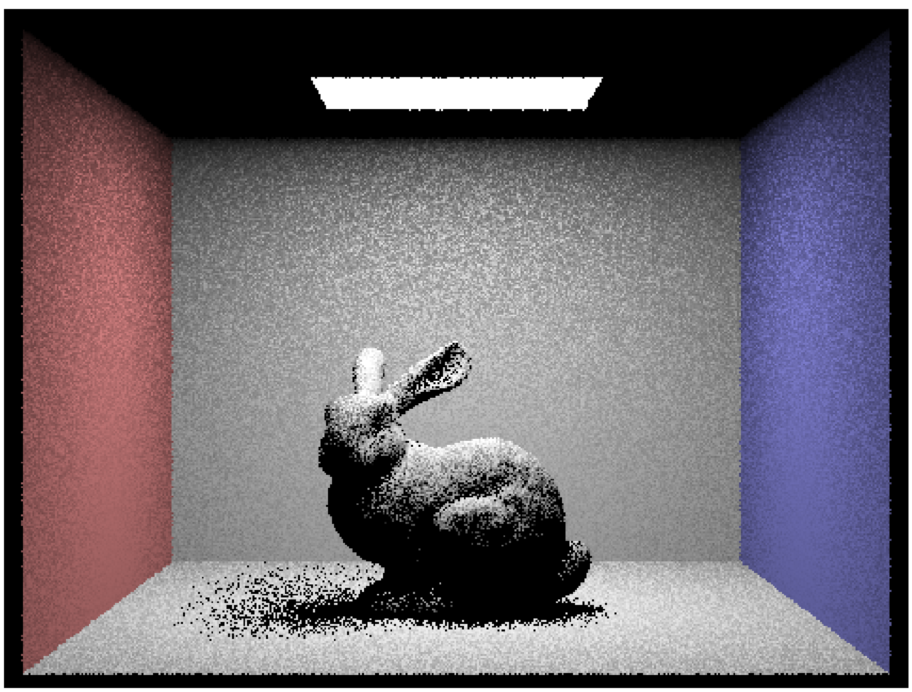
Following flags are used: `-t 8 -s 2048 -a 64 0.05 -l 1 -m 5 -r 480 360`

Collaboration

Wenhan Sun and Catherine Gai worked through all the project detail together. However, due to the large amount of pictures that needed to be rendered, picture rendering is done on multiple computers.

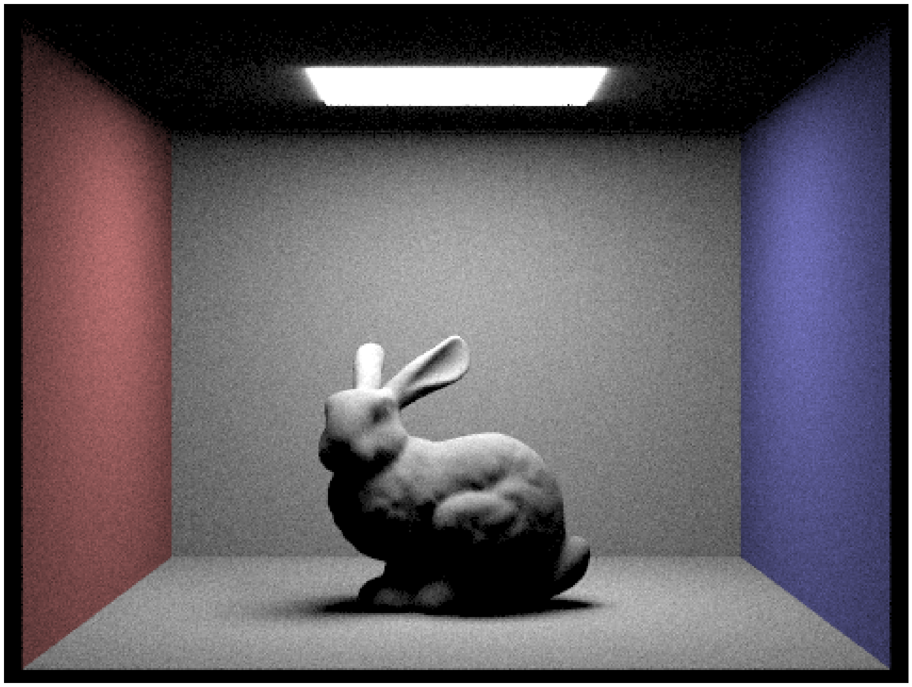


(a) Hemisphere sampling

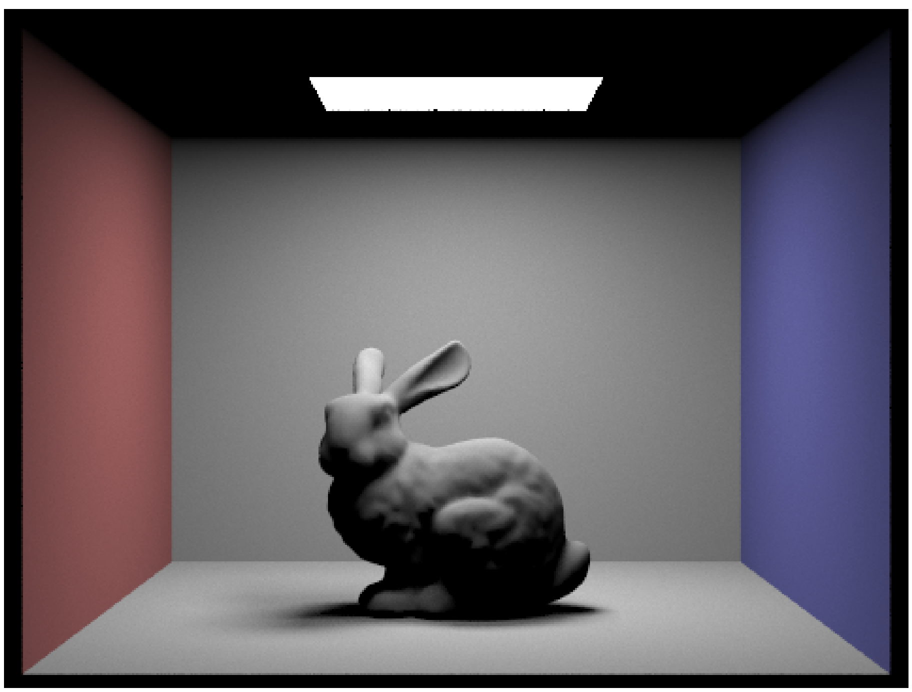


(b) Lighting sampling

Figure 7: Renderings of `sky/CBbunny.dae` with direct illumination and low sampling rate. The following flags are used:
Hemisphere sampling: `-t 8 -s 1 -l 1 -m 1 -r 480 360 -H`
Lighting sampling: `-t 8 -s 1 -l 1 -m 1 -r 480 360`



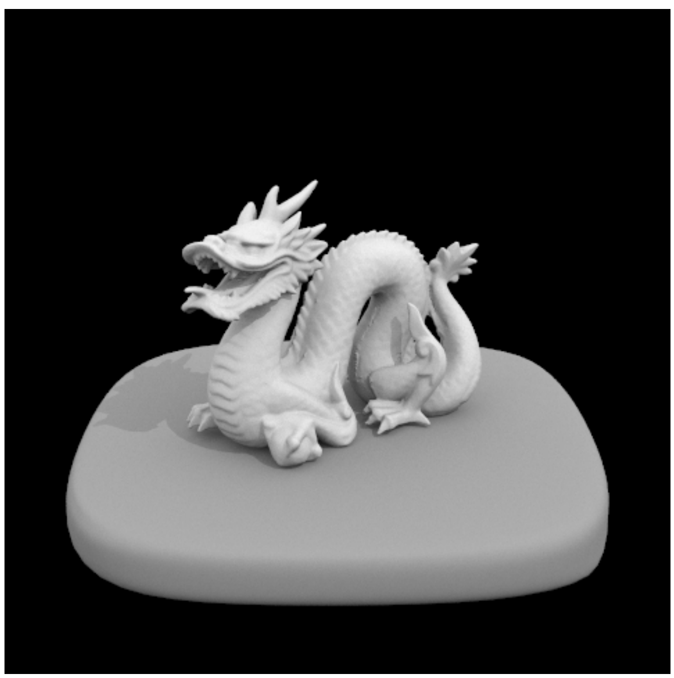
(a) Hemisphere sampling



(b) Lighting sampling

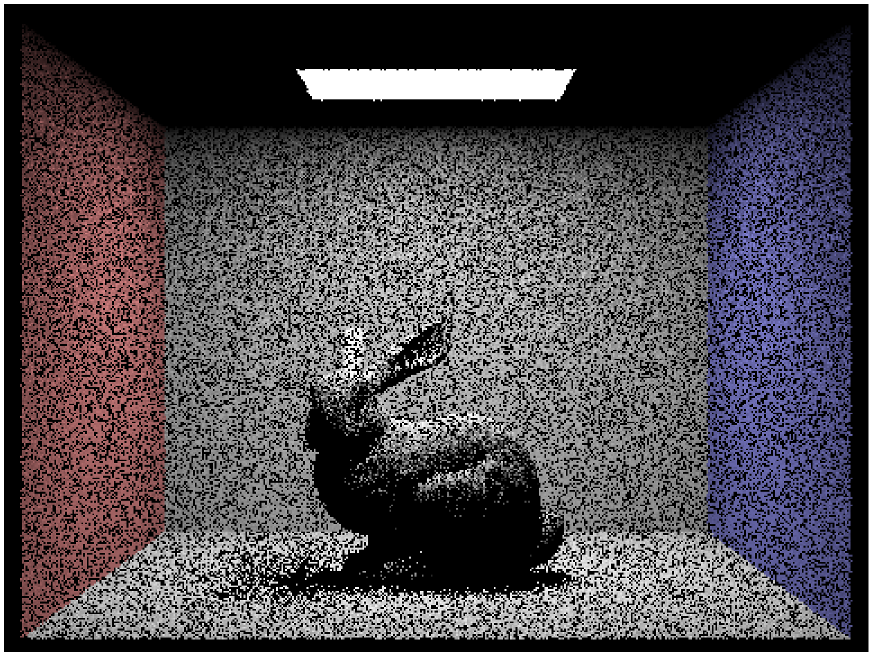
Figure 8: Renderings of `sky/CBbunny.dae` with direct illumination and high sampling rate. The following flags are used:
Hemisphere sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 360 -H`
Lighting sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 360`

(a) Hemisphere sampling

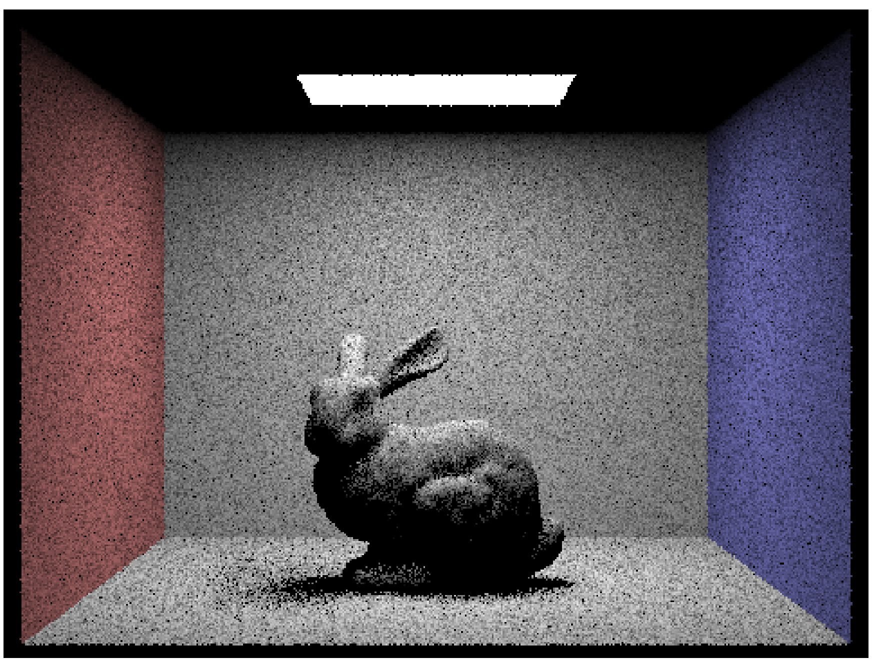


(b) Lighting sampling

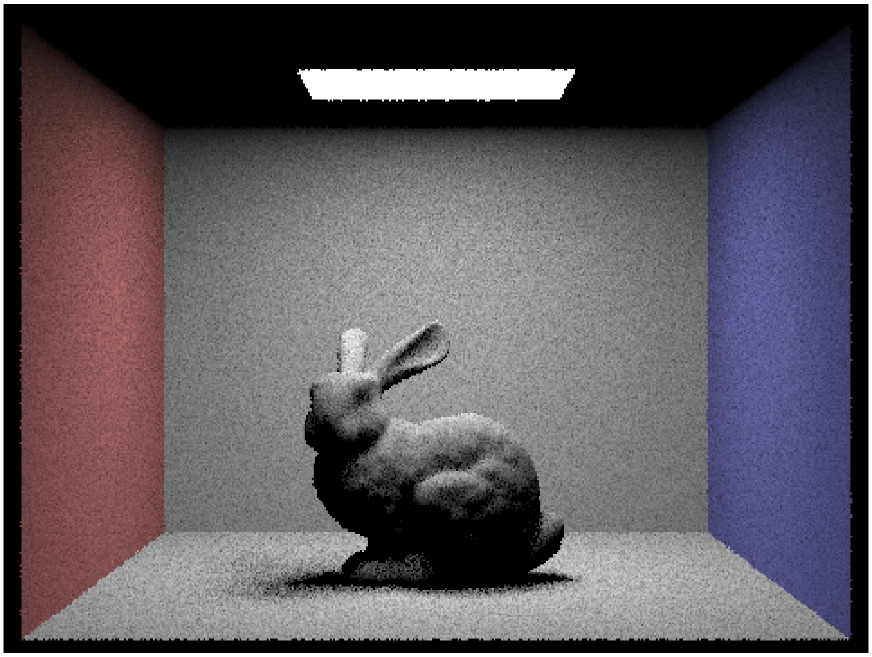
Figure 9: Renderings of `sky/dragon.dae` with direct illumination and high sampling rate. The following flags are used:
Hemisphere sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 360 -H`
Lighting sampling: `-t 8 -s 64 -l 32 -m 6 -r 480 360`



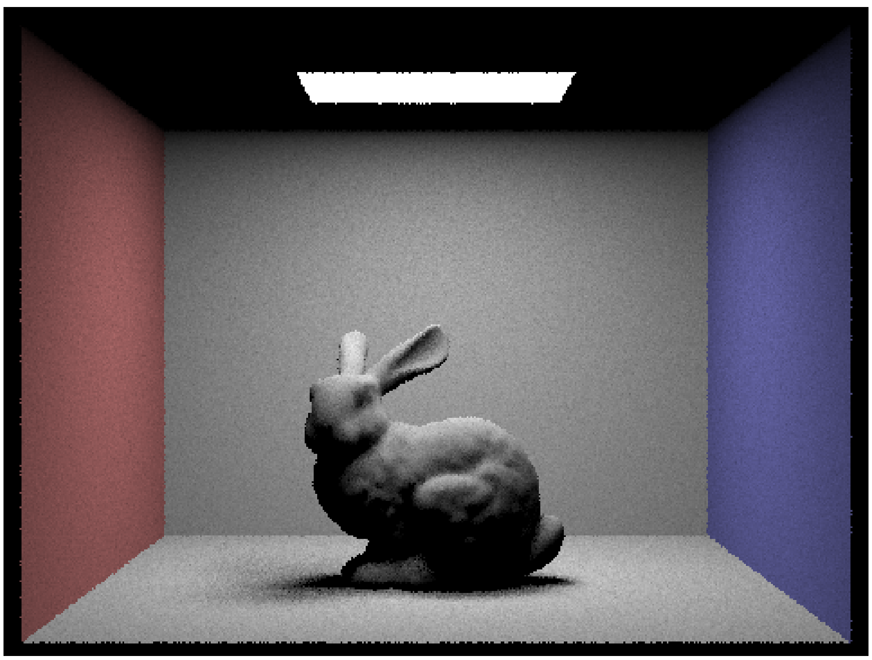
(a) Lighting sample rate 1



(b) Lighting sample rate 4



(c) Lighting sample rate 16



(d) Lighting sample rate 64

Figure 10: Renderings of `sky/CBbunny.dae` with direct illumination using lighting sampling of various lighting sample rates. The following flags are used:
Sampling rate 1: `-t 8 -s 1 -l 1 -m 6 -r 480 360`
Sampling rate 4: `-t 8 -s 1 -l 4 -m 6 -r 480 360`
Sampling rate 16: `-t 8 -s 1 -l 16 -m 6 -r 480 360`
Sampling rate 64: `-t 8 -s 1 -l 64 -m 6 -r 480 360`

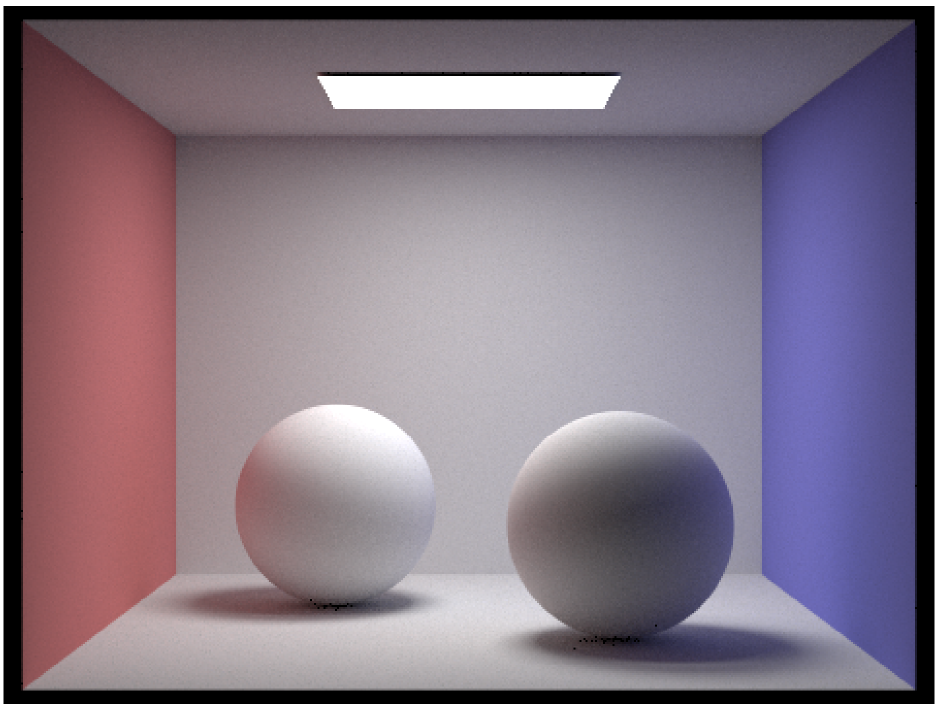
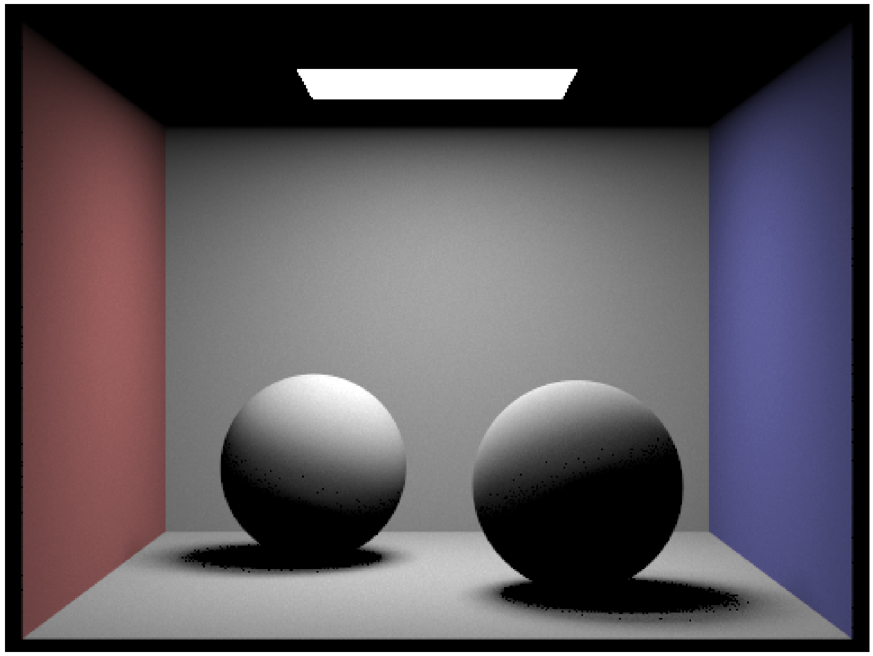
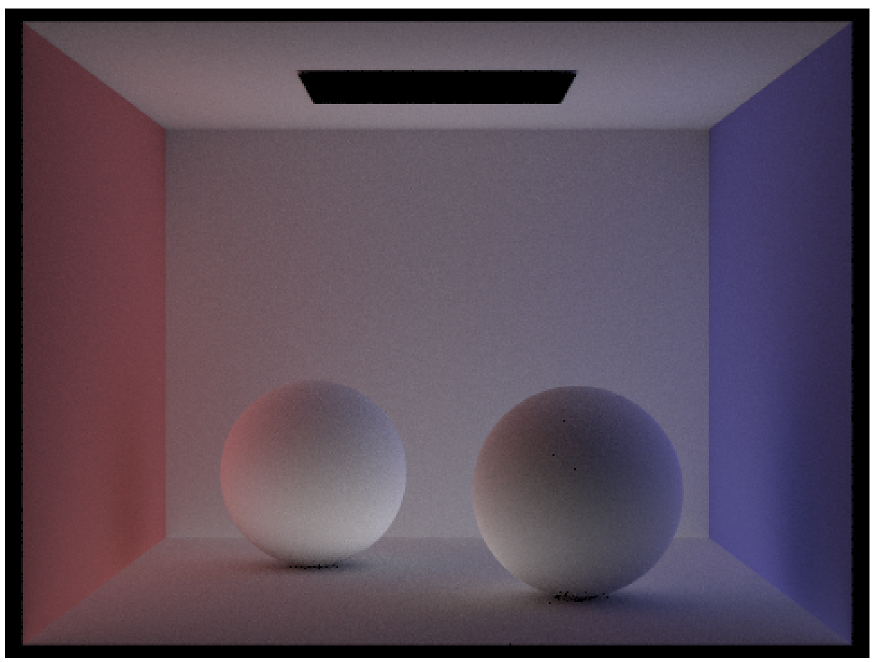


Figure 11: Rendering of `sky/CBspheres_lambertian.dae` with global illumination using lighting sampling of pixel sample rate 1024. The following flags are used:
`-t 8 -s 1024 -l 16 -m 5 -r 480 360`

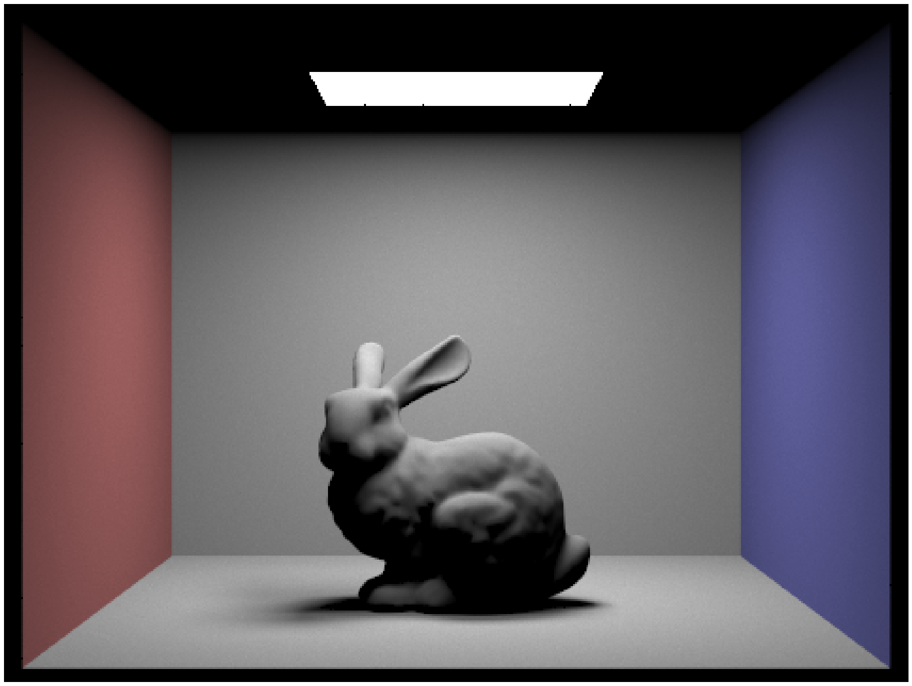


(a) Direct illumination

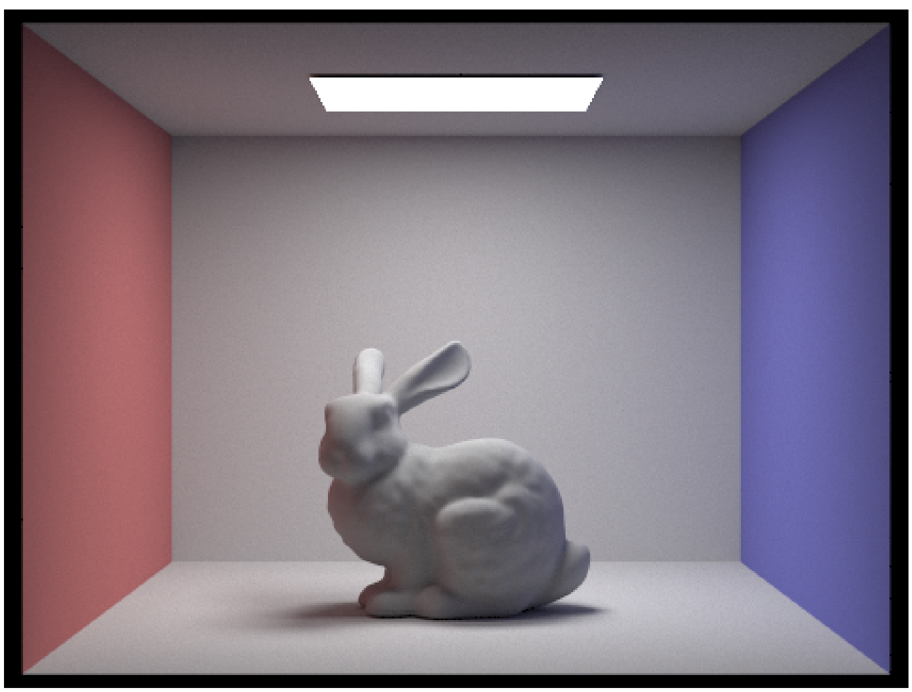


(b) Indirect illumination

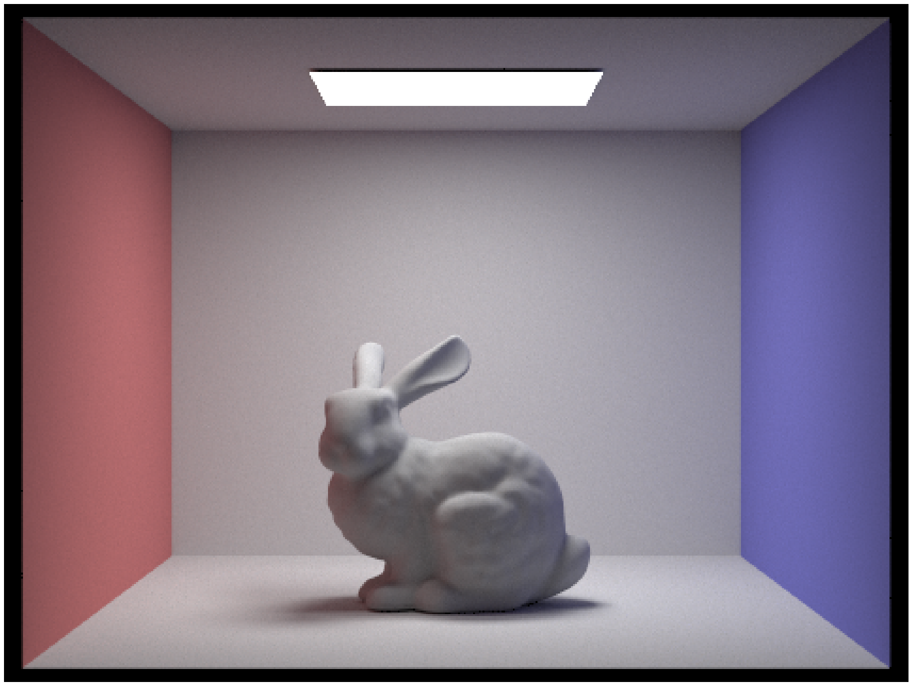
Figure 12: Renderings of `sky/CBspheres_lambertian.dae` with direct and indirect illumination using lighting sampling of pixel sample rate 1024. The following flags are used:
`-t 8 -s 1024 -l 16 -m 5 -r 480 360`



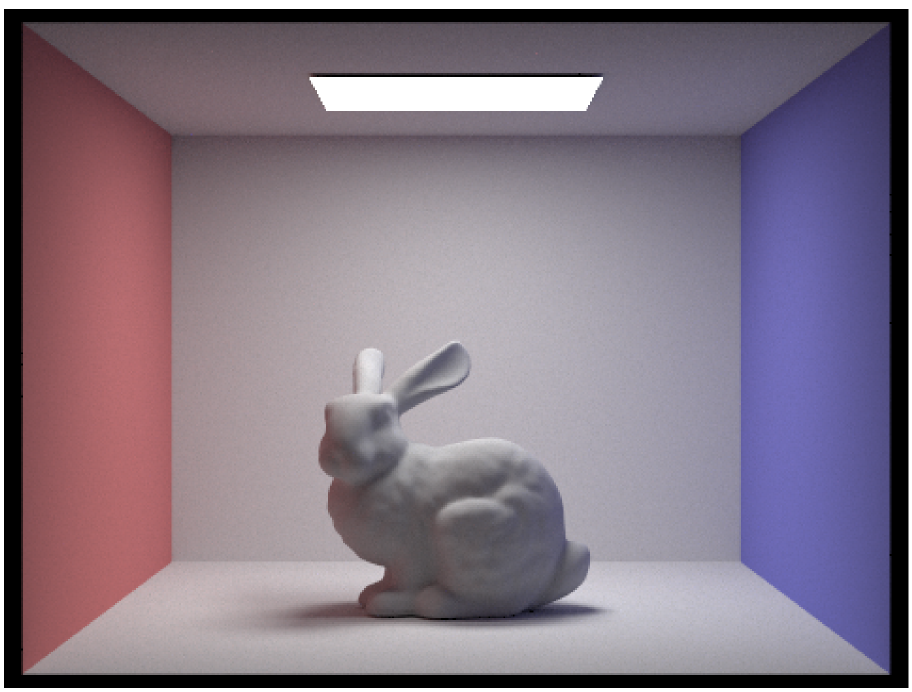
(a) Maximum ray depth 1



(b) Maximum ray depth 2

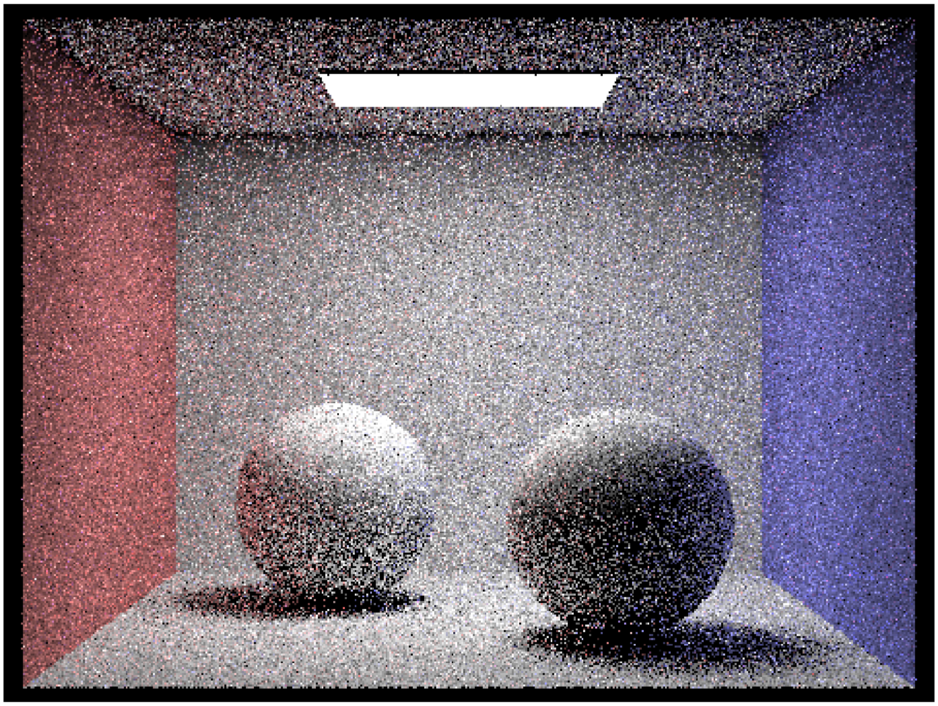


(c) Maximum ray depth 3

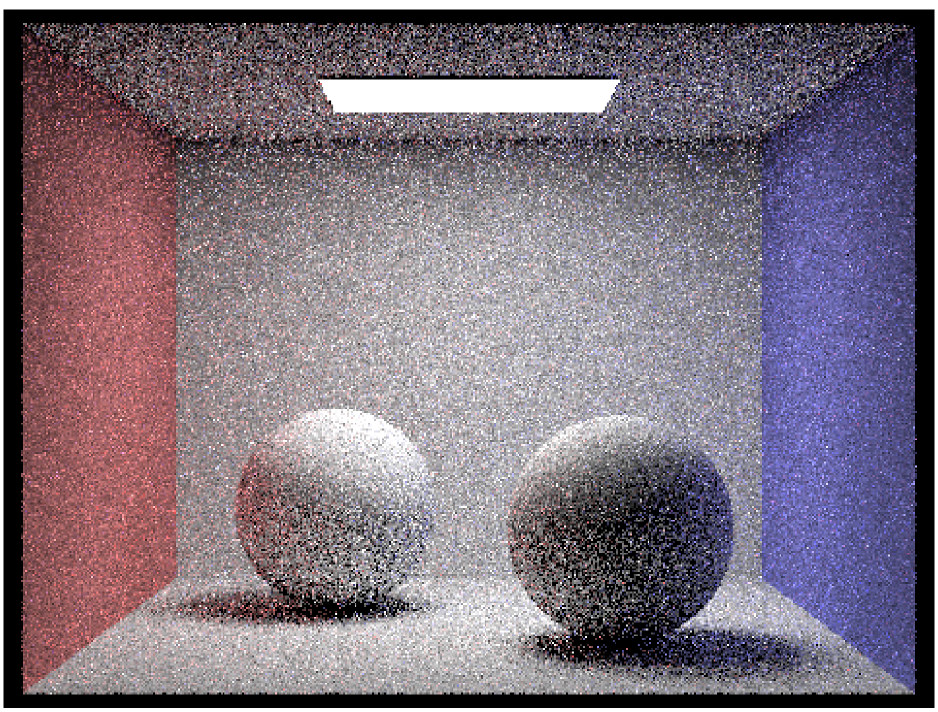


(d) Maximum ray depth 100

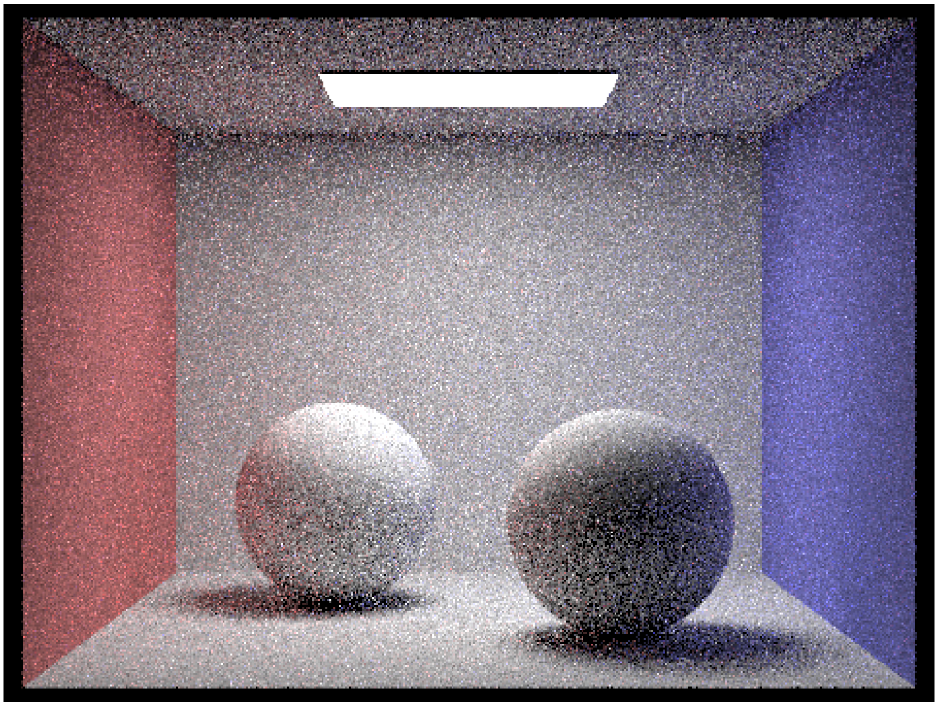
Figure 13: Renderings of `sky/CBbunny.dae` with global illumination using lighting sampling of various maximum ray depths. The pixel sample rate is 1024. The following flags are used: Maximum ray depth 1: `-t 8 -s 1024 -l 16 -m 1 -r 480 360` Maximum ray depth 2: `-t 8 -s 1024 -l 16 -m 2 -r 480 360` Maximum ray depth 3: `-t 8 -s 1024 -l 16 -m 3 -r 480 360` Maximum ray depth 100: `-t 8 -s 1024 -l 16 -m 100 -r 480 360`



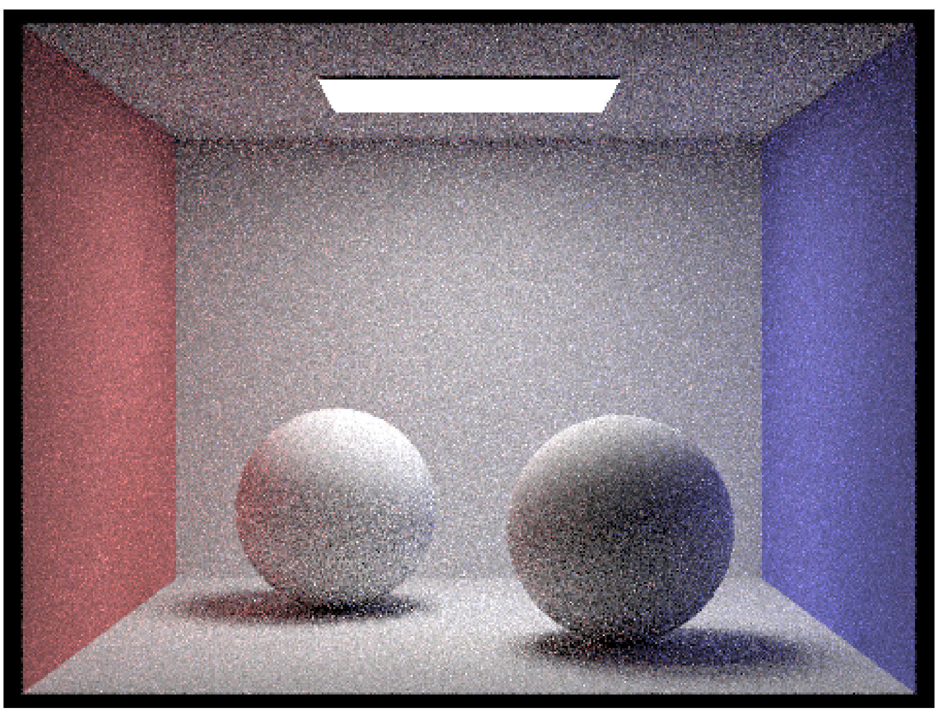
(a) Pixel sample rate 1



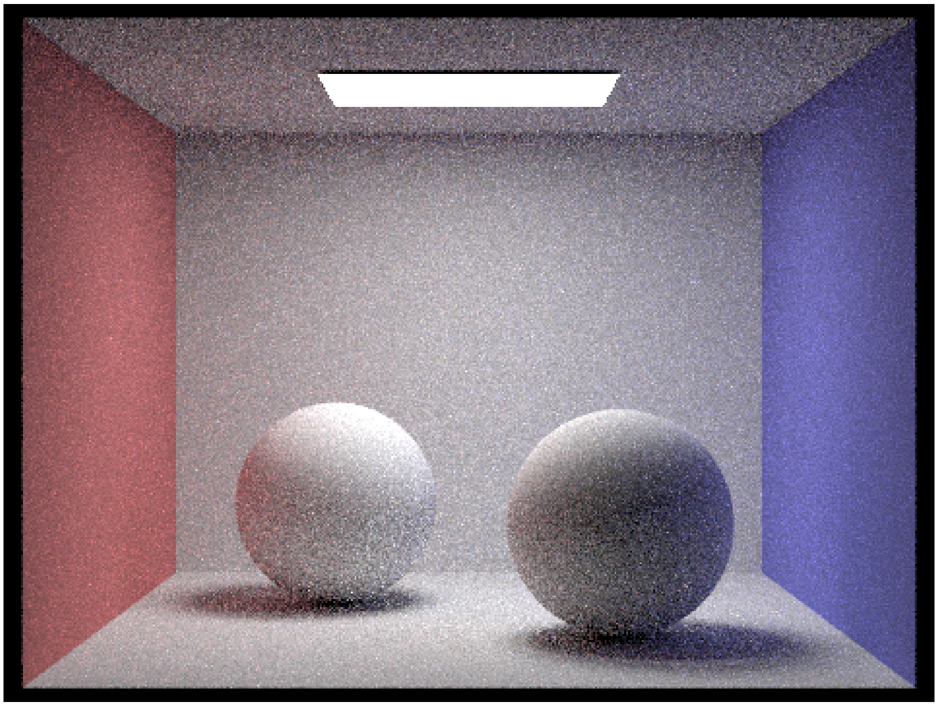
(b) Pixel sample rate 2



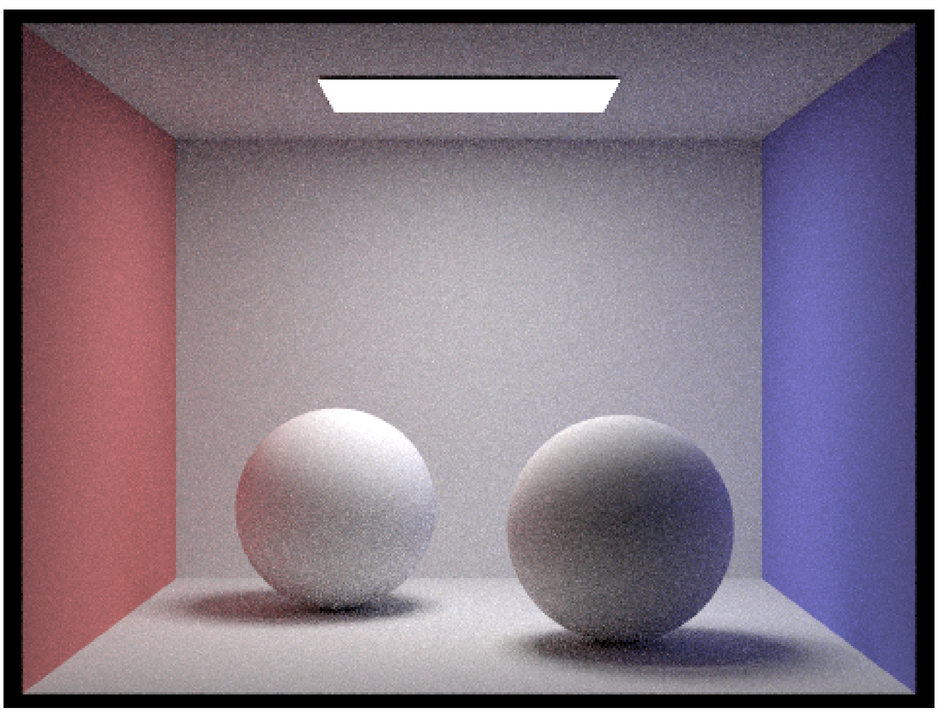
(c) Pixel sample rate 4



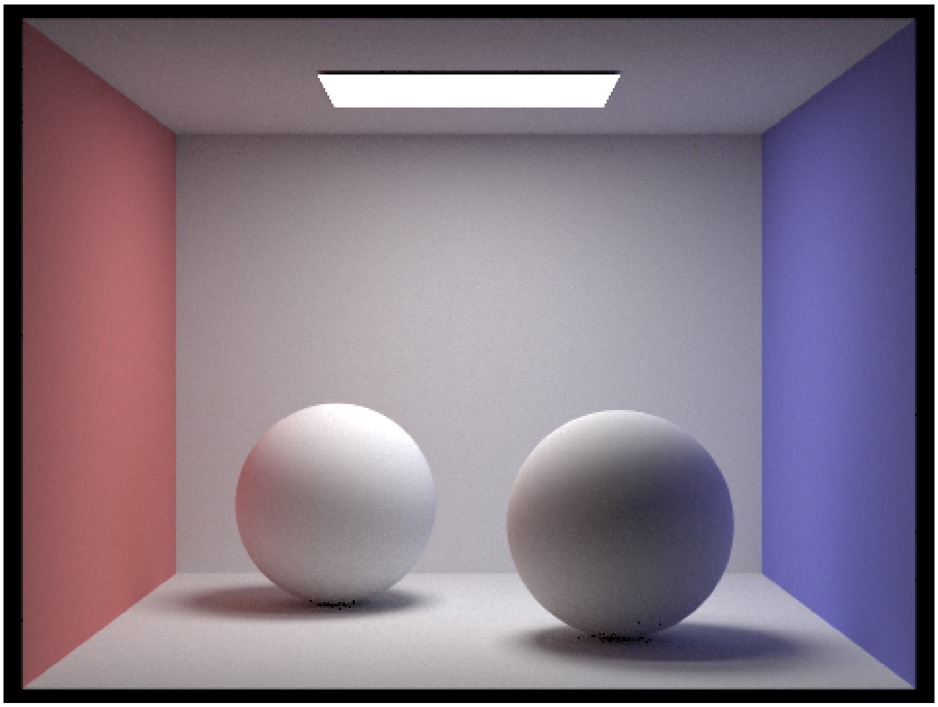
(d) Pixel sample rate 8



(e) Pixel sample rate 16

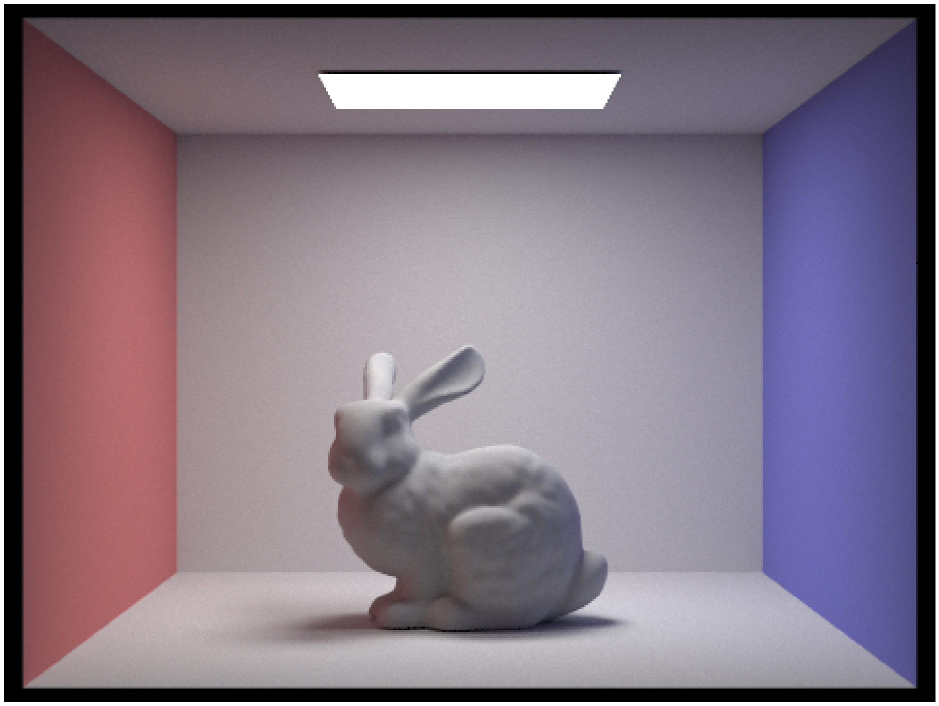


(f) Pixel sample rate 64

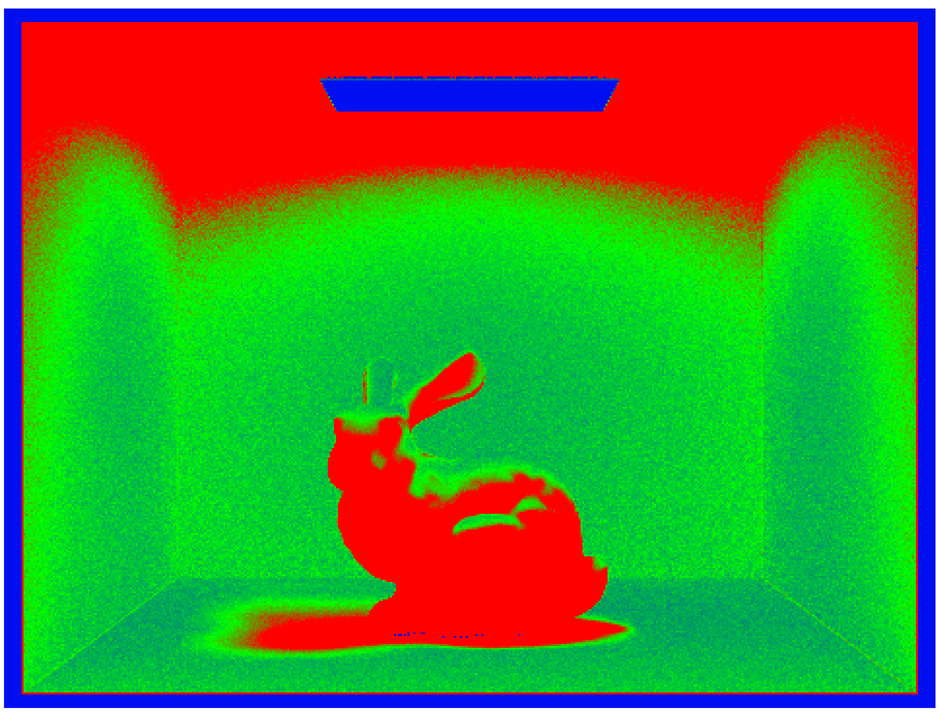


(g) Pixel sample rate 1024

Figure 14: Renderings of `sky/CBspheres_lambertian.dae` with global illumination using lighting sampling of various pixel sample rates. The light sample rate is 4. The following flags are used:
Pixel sample rate 1: `-t 8 -s 1 -l 4 -m 5 -r 480 360`
Pixel sample rate 2: `-t 8 -s 2 -l 4 -m 5 -r 480 360`
Pixel sample rate 4: `-t 8 -s 4 -l 4 -m 5 -r 480 360`
Pixel sample rate 8: `-t 8 -s 8 -l 4 -m 5 -r 480 360`
Pixel sample rate 16: `-t 8 -s 16 -l 4 -m 5 -r 480 360`
Pixel sample rate 64: `-t 8 -s 64 -l 4 -m 5 -r 480 360`
Pixel sample rate 1024: `-t 8 -s 1024 -l 4 -m 5 -r 480 360`



(a) Rendered image



(b) Sample rate buffer

Figure 15: Rendering of `sky/CBbunny.dae` with global illumination using adaptive lighting sampling. The light sample rate is 4. The following flags are used:
`-t 8 -s 2048 -a 64 0.05 -l 1 -m 5 -r 480 360`