Brief Introduction to 2D Global Illumination
2D Global Illumination no Susume[*][†]

perim
@hi2p_perim

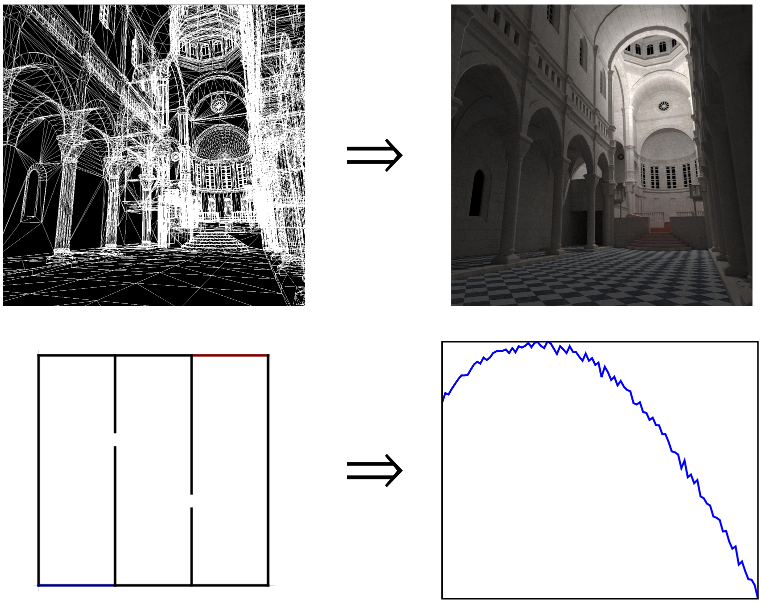

Figure 1: Input scenes and rendered results of 3D global illumination and 2D global illumination. In 3D global illumination (top), the scene is given by a set of *triangle meshes* and the rendered result is a 2D image. On the other hand in 2D global illumination (bottom), the scene is given by a set of *line segments* and the rendered result is a 1D histogram.

1 Introduction

It goes without saying that the development of global illumination techniques is important for many people. However, as many of you already might have noticed, their implementation is not so simple. Even a fundamental technique like *path tracing* [4], when you actually try to implement it, you may suffer from difficulties such as tricky bugs, long-lasting debugging, and difficulty in verification. One way to alleviate these difficulty is to utilize the *2D global illumination* proposed by Jarosz et al. [3]. In this technique, the dimension of the scene is reduced from 3D to 2D, which results to generate 1D

[*]This article is written for the advent calender of the ray-tracing camp 3
[†]`https://sites.google.com/site/raytracingcamp3/`

Quantity	3D expression	3D unit	2D expression	2D unit
Flux	$\Phi_{3D}(A) = \frac{dQ(A)}{dt}$	$W = J \cdot s^{-1}$	$\Phi_{2D}(\mathcal{L}) = \frac{dQ(\mathcal{L})}{dt}$	$W = J \cdot s^{-1}$
Irradiance	$E_{3D}(\mathbf{x}) = \frac{d\Phi_{3D}(A)}{da(\mathbf{x})}$	$W \cdot m^{-2}$	$E_{2D}(\mathbf{x}) = \frac{d\Phi_{2D}(\mathcal{L})}{dl(\mathbf{x})}$	$W \cdot m^{-1}$
Radiance	$L_{3D}(\mathbf{x}, \omega) = \frac{d^2\Phi_{3D}(A)}{(\mathbf{n}\cdot\omega)^+ d\omega da(\mathbf{x})}$	$W \cdot m^{-2} \cdot sr^{-1}$	$L_{2D}(\mathbf{x}, \theta) = \frac{d^2\Phi_{2D}(\mathcal{L})}{\cos\theta d\theta dl(\mathbf{x})}$	$W \cdot m^{-1} \cdot rad^{-1}$

Table 1: Representative radiometric quantities for 2D/3D models. Here $(\mathbf{n} \cdot \omega)^+$ is the clamped dot product with positive number, i.e. $(\mathbf{n} \cdot \omega)^+ \equiv \max(0, \mathbf{n} \cdot \omega)$.

histogram as a rendered result instead of 2D image. Actually it is substantially effective for verification of rendering techniques because of its easiness of visualization, and easiness of handling the result, and small computation time. In this article, I will introduce some basic concepts of the theory of 2D global illumination especially focusing on the difference between 3D formulation. Also, I will note some comments on the implementation of global illumination techniques in 2D, for the sake of helping those who actually try to implement them. In order to read and understand this article, you may want to have some knowledge on the theory of 3D global illumination (e.g. see PBRT [6], edupt [1], etc.).

2 Theory of 2D Global Illumination

2.1 2D Radiometry

In order to derive 2D counterpart of 3D rendering equation, we need to define radiometric quantities for the 2D world. In this section I will describe several 2D counterparts of the 2D radiometric quantities following the definition by Jarosz et al. [3]. The definition of the quantities are almost same as the 3D version, major difference is the dimension of the measurement target. For instance, In 3D radiometry, *radiant power* or *flux* is defined as a function of *surface area*: $\Phi_{3D}(A) = \frac{dQ(A)}{dt}$, which represents the energy flowing from/to/through A per unit time. On the other hand, its 2D counterpart is defined as a function of a *curve*: $\Phi_{2D}(\mathcal{L}) = \frac{dQ(\mathcal{L})}{dt}$, which represents the energy flowing from/to/through $\mathcal{L}$ per unit time. Also, these spatial dependence can cause the change in units for some quantities. For instance, in 3D formulation, the *irradiance* is defined as the incident radiant power per unit *surface area*. On the other hand, in 2D radiometry, the irradiance is define as the incident radiant power per unit *arc length*. As a result of the difference of dimensions in differentials, we can see that the unit of 3D/2D quantity is different; $J \cdot s^{-2}$ for 3D irradiance, and $J \cdot s^{-1}$ for 2D irradiance. Brief comparison of 2D and 3D radiometric quantities are summarized in Table 1.

2.2 2D BRDF

As for 3D global illumination, 2D counterpart of the *bidirectional scattering distribution function* (BRDF) is also defined as a function describing the relationship between differential irradiance and reflected differential radiance:

$$f_{2D}(\mathbf{x}, \theta \to \theta') \equiv \frac{dL_{2D}(\mathbf{x} \to \Phi')}{dE_{2D}(\mathbf{x} \leftarrow \Phi)}. \quad (1)$$

Various properties of BRDF held for 3D definition such as *range property*, *reciprocity* or *energy conservation* are also established in the 2D formulation. Similar to 3D version, the reflected radiance L^r_{2D} can be obtained with converting the definition of BRDF (Equation 1) to

$$L^r_{2D}(\mathbf{x} \to \theta') = \int_\Theta f_{2D}(\mathbf{x}, \theta \to \theta') L_{2D}(\mathbf{x} \leftarrow \theta) \cos\theta d\theta. \tag{2}$$

2.3 2D Rendering Equation

The rendering equation formulates equilibrium of light energy distributed in the scene. This equation is a fundamental formulation of all rendering technique because in order to compute global illumination, every rendering techniques somehow needs to solve the equation. The derivation of the 2D counterpart of the rendering equation is almost same as 3D. From the conservation of energy, for all point $\mathbf{x}$ in the scene surface (on a curve, in 2D configuration), the outgoing radiance is equal to a sum of emitted and reflected radiance:

$$L_{2D}(\mathbf{x} \to \theta') = L^e_{2D}(\mathbf{x} \to \theta') + L^r_{2D}(\mathbf{x} \to \theta'). \tag{3}$$

Next, by applying Equation 2, we obtain the 2D version of the *rendering equation*:

$$L_{2D}(\mathbf{x} \to \theta') = L^e_{2D}(\mathbf{x} \to \theta') + \int_\Theta f_{2D}(\mathbf{x}, \theta \to \theta') L_{2D}(\mathbf{x} \leftarrow \theta) \cos\theta d\theta. \tag{4}$$

3 Implementation of 2D Global Illumination Techniques

3.1 General Tips and Pitfalls

In this section I will introduce some tips and pitfalls when we try to implement 2D global illumination techniques utilizing the formulation described in Section 2, based on the author's experience.

Scene Description Change in dimensions changes the scene description that the rendering system must support. Theoretically the scene in the 2D framework is a set of 2D curves, it is generally acceptable for verification to utilize only a set of *line segments*, which is analogous to utilizing triangle meshes as scene description for the 3D framework. Although 2D line segments are easier to handle intuitively, scene creation is one of the difficult parts when you want to create a scene with a number of line segments. However, there is no counterparts of 3D mesh modeler which is usable for 2D scene creation. One approach is to implement a plugin for vector graphics editors such as Illustrator or Inkscape, or create a converter from some vector graphics formats. However generally vector graphics is designed to handle not only segments, and in this case apparently over specification. Considering requirements and time resources, I decided to create a specialized software for scene creation (Figure 2). Developing this kind of software is much easier for 2D framework than for 3D framework.

Implementation Cost Rendering framework is often a complex software system. As for 3D rendering system, in addition to the essential part for rendering technique, we need to implement various components such as ray-triangle intersection, scene loading, or handling input/output images, or need to use

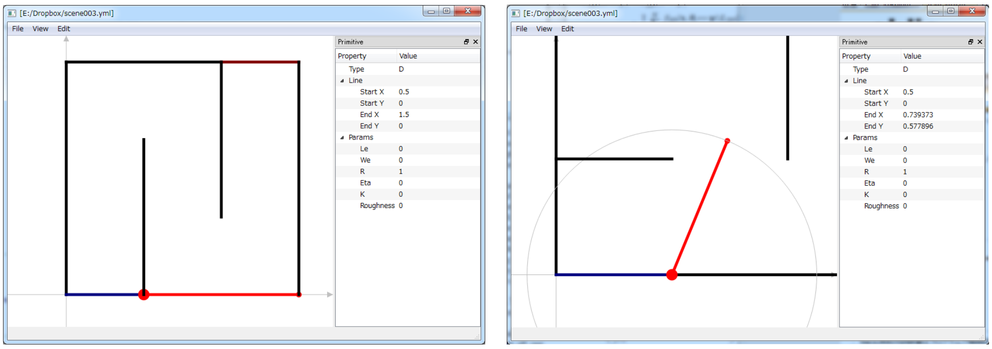

Figure 2: 2D scene editor. I assume the scene is composed of line segments only. A dark blue segment shows area sensor and a dar red segment shows area light. Bright red segment is the selected one.

libraries for them. One of the potential benefits for the implementation of 2D global illumination is that the implementation of these component could be much simpler than 3D version, which is effective for keeping entire system simple.

Easiness of Visualization In 2D framework, use of visualization is stronger part than 3D framework. Visualization in 2D is generally much simpler than in 2D. For example, visualization of path mutations in MCMC based rendering techniques such as Metropolis light transport [9], it is often useful to visualize a trajectory of the mutations to find bug-inducing unnatural movements of light paths such as unexpected sequence of rejections. With 2D framework, such a visualization can be implemented extremely easily because we just need to draw some sequence of lines (Figure 3). The strength is much more explicit when we consider the visualization of *volumes*. In 3D environment, visualization of 3D volumes is difficult in nature because the output device is restricted to the display which is only capable to display 2D image. On the other hand, 2D volumes can be easily visualized as a shading of the region excluding line segments.

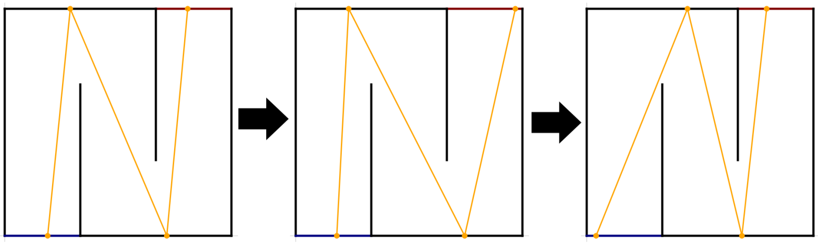

Figure 3: An example of visualization in 2D light paths. This figure indicates a sequence of mutated light paths with a mutation strategy in MLT.

Handling symmetry of light transport Theoretical study of light transport states that the formulation of light transport can be *symmetric* [7]. So calculating intensity of a pixel by tracing light path from a sensor (radiance transport) is theoretically equivalent to by tracing light path from a light source (importance transport). Designing rendering system keeping this symmetry has several benefits. For example, the verification of the rendering techniques based on bidirectional light transport such as the *bidirectional path tracing* (BDPT) [5] [8], or MLT [9] are simpler. In symmetrical formulation, the simplest combination of *emitters* (i.e., light sources and sensors) is to utilize finite-area emitters, e.g., *area light* and *area sensor* because we do not have to consider special handing of degenerated emitters which is likely to be a cause of bugs in the implementation of bidirectional techniques. For instance, calculating weighting function for the multiple importance sampling (MIS) in BDPT requires to exclude some strategies (probability distribution that generates lights paths) with which an evaluated path is sampled in zero probability. This kind of abstraction using finite-area emitters seems a natural choice for 2D environment because when we record contribution to the histogram, we can use a length from a edge of a line segment to a query position on the sensor. Actually 2D scene depicted in Figure 2 or 3 utilized finite-area sensors. I note that of course it is possible to implement "area sensor" in 3D by placing a quad in 3D and associate UV coordinates with pixel index, although it is questionable that it is a natural choice.

Handling Geometry Term Some formulation of light transport involves in a function called *geometry term*. For example, it is utilized in the definition of the *measurement contribution function* in Veach's path integral formulation [7]. This function is originally derived from the measure conversion from $d\theta$ to $dl^{\perp}$ so essentially the function is a Jacobian:

$$G_{2D}(\mathbf{x} \leftrightarrow \mathbf{y}) \equiv \frac{d\theta(\mathbf{x})}{dl^{\perp}(\mathbf{y})} = \cos\theta_{\mathbf{x}} \frac{d\theta(\mathbf{x})}{dl(\mathbf{y})} = \frac{\cos\theta_{\mathbf{x}} \cos\theta_{\mathbf{y}}}{\|\mathbf{x} - \mathbf{y}\|}, \tag{5}$$

where $\mathbf{x}$ and $\mathbf{y}$ are points in the scene surface and $\theta_{\mathbf{x}}$ and $\theta_{\mathbf{y}}$ are the angles between the surface normal at $\mathbf{x}$ and $\mathbf{y}$ and the outgoing vectors respectively. On the other hand, the 3D counterparts of the geometry term is defined as

$$G_{3D}(\mathbf{x} \leftrightarrow \mathbf{y}) = \frac{|\cos\theta_{\mathbf{x}}||\cos\theta_{\mathbf{y}}|}{\|\mathbf{x} - \mathbf{y}\|^2}. \tag{6}$$

Confusing point is a difference in the numerators. You should be careful not to mistake a numerator in 2D definition for 3D definition, which often introduce tricky bugs.

3.2 Results

In this section I will introduce experimental results for some global illumination techniques implemented with 2D formulation describe above. All experiments are performed on a PC with Intel Core i7-3970X CPU @ 3.50GHz using 12 threads. I implemented several rendering techniques for comparison: *path tracing* (PT) [4], *light tracing* (LT) [2], and *bidirectional path tracing* (BDPT) [5] [8].

Figure 4 shows the estimated distributions for each technique with a different number of samples (10^7 and 10^9). In this experiment the maximum number of path vertices are limited to 5, and the bin size of the output histogram is fixed to 100. The scenes are designed to increase its complexity from top to bottom in Figure 4. We can observe that the effectiveness of bidirectional techniques from the third scene. However the result of the fourth scene shows that even a bidirectional technique have still have noticeable error on the result with a larger number of samples. Rendering such a scene can be improved utilizing MCMC based techniques which is of course possible to be implemented in the 2D framework. I also note that in this scene configuration the variance of a estimator used in PT and LT are theoretically same due to the symmetry of light transport.

4 Conclusion

In this article I introduced the 2D global illumination which is useful for the verification of the rendering techniques. In the first part of the article, I briefly introduced the theoretical concepts of 2D global illumination such as 2D radiometry and 2D rendering equation and pointed out its similarities between 3D counterparts. Next I described the implementation details of the 2D global illumination techniques focusing on some tips and pitfalls.

References

[1] edupt. `http://kagamin.net/hole/edupt/index.htm`. Accessed: 2015-06-22.

[2] J. Arvo and D. Kirk. Particle transport and image synthesis. *Computer Graphics (Proc. of SIGGRAPH 90)*, 24(4):63–66, 1990.

[3] W. Jarosz, V. Schonefeld, and L. Kobbelt. Theory, analysis and application of 2d global illumination. *ACM Transactions on Graphics*, 31(5):Article 125, 2012.

[4] J. T. Kajiya. The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH 86)*, pages 143–150, 1986.

[5] E. Lafortune and Y. D. Willems. Bi-directional path-tracing. *Compugraphics 93*, pages 145–153, 1993.

[6] M. Pharr and G. Humphreys. *Physically Based Rendering*. Morgan Kaufmann, 2010.

[7] E. Veach. Robust monte carlo methods for light transport simulation. *Ph.D. thesis, Stanford university, Department of Computer Science*, 1997.

[8] E. Veach and L. J. Guibas. A ray tracing solution for diffuse interreflection. *Fifth Eurographics Workshop on Rendering*, pages 147–162, 1994.

[9] E. Veach and L. J. Guibas. Metropolis light transport. *Proc. of SIGGRAPH 97*, pages 65–76, 1997.

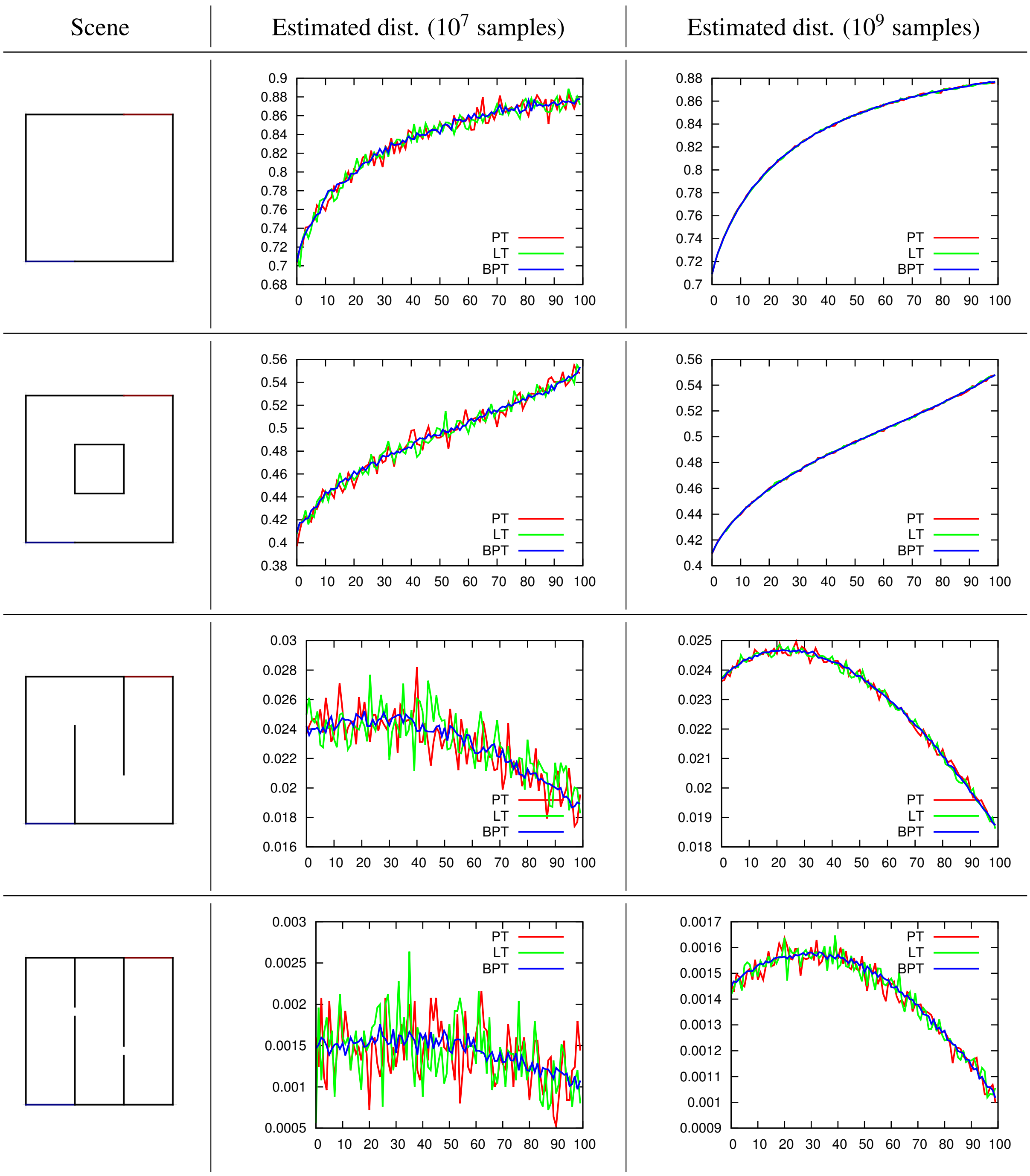

Figure 4: Estimated distributions with 2D framework. The left column shows the scenes which is composed of line segments. The red line segment shows a finite-area light source, and the blue line segment shows a finite-area sensor. The middle column shows the estimated distribution with 10^7 samples and the right column show the estimated distribution with 10^9 samples. The horizontal axis indicates a position on the sensor (i.e., a blue line segment) and the vertical axis indicates estimated luminance at the position.