Rendering

Sung-eui Yoon
KAIST
January 19, 2017

Chapter 5

Gradient-Domain Rendering

In this chapter, we discuss rendering techniques that compute and utilize gradients of images. Since gradients of images are more sparse than the radiance representation of images, working with gradients can be more sparse and thus efficient. Nonetheless, various issues arise from the very nature of the sparsity of gradients. We see how these issues are addressed in related techniques.

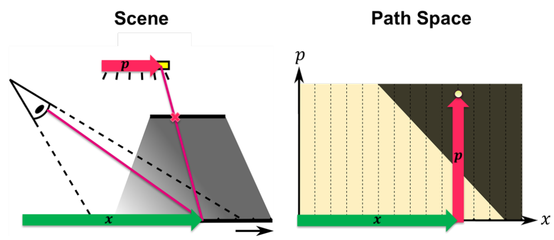
5.1 Gradient-Domain Metropolis Light Transport

Initially, gradient-domain approaches, sampling and utilizing gradients of an image, were designed for Metropolis light transport (Sec. **??** YOON's comment: Need to discuss this part), but extended into many different rendering methods. In this section, we briefly discuss ideas of the gradient-domain approach for the Metropolis light transport, and move to gradient-domain path tracing (Sec. 5.2).

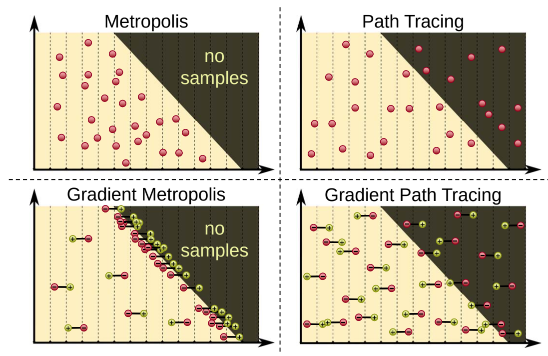
The Metropolis approach samples mainly on regions with high contributions, i.e., bright regions, to the final image. While this approach has been demonstrated to handle difficult cases by exploring rare events well, it also needs to generate many samples on bright and smoothly varying regions.

The gradient-domain approach is based on an intuitive idea that most information of an image is around edges and other variations and thus focuses on those variations. This intuition is well explored in related fields. For example, gradients of natural images tend to be sparse, and thus can serve as a compact image representation [SO01]. Also, the power spectrum of such natural images is approximately inversely proportional to the squared frequency. This indicates that gradients carry less energy than the original image values, and thus can be less noise YOON's comment: Need to be delivered in a intuitive way .

The technical difficulty of sampling such regions with edges and variations requires us to know such regions in advance before the sampling process. To tackle this problem, the Metropolis approach was adopted, to explore such regions effectively, Surprisingly, this gradient-domain approach has been identified useful even for other rendering methods including path tracing. We thus study this approach in a detailed manner in a next section.



(a) Scene configuration



(b) Sampling patterns

Figure 5.1: The left image of (a) shows a scene configuration parameterized by x and p, which are 1D spatial coordinates in the receiver and occluder, respectively. The right image of (a) shows the path space representation parametrized by those two parameters. (b) shows schematic difference between gradient-domain and primary-domain rendering methods. This image is excerpted from a talk slide of [KMA$^+$15].

5.2 Gradient-Domain Path Tracing

While the gradient-domain approach was mainly designed for Metropolis sampling, it is also demonstrated to work well with path tracing. This development is critical since the idea is demonstrated to work well in a wider set of algorithms and scenes. For example, Metropolis sampling is good for identifying and exploring difficult-to-find light paths, but shows too dark or bright regions in a chunky way. Furthermore, its convergence rate of error reduction can be undesirable, since its sampling pattern is highly non-uniform. On the other hand, path tracing shows robust performance in many practical scenes that we encounter.

Fig. 5.1 shows sampling patterns of different methods. Suppose a simple scene configuration shown in the left image of the figure. To compute an image, we generate a ray passing through the eye, hitting a point in the floor, which is parameterized in a x coordinate. To compute the

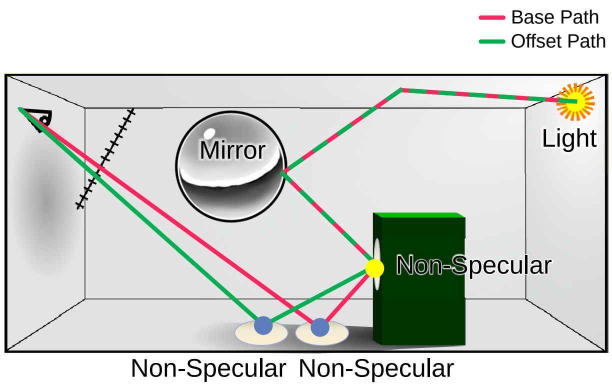


Figure 5.2: For gradient-domain rendering, we also generate offset paths that are coherent to the base light paths. This way we generate less-coherent gradients than ones computed from arbitrary ray paths. This image is excerpted from a slide of the work [KMA$^+$15].

incoming radiance to the hit point, we also generate another ray, i.e., pink rays, toward the yellow light source. Some of those secondary rays intersect with the blocker located in the middle of the scene. This intersection point with the black blocker object is denoted p in this example.

Given this setting of x and p, two intersection points, we can represent each ray in a path space parametrized with those two parameters. (b) of Fig. 5.1 visualizes how different sampling patterns different rendering methods have.

Path tracing generates uniformly in these two parameter spaces, while Metropolis sampling generates samples according to the contribution of each pixel value. On the other hand, the gradient-domain Metropolis sampling aims to mainly cover boundary cases so that we can robustly compute gradients of the image. In a similar direction, we would like to apply this idea to path tracing.

Most gradient-domain methods including gradient-domain path tracing have the following components:

- **Offset path generation.** Gradient-domain path tracing computes gradients in additional to pixel values, i.e, radiances. To do that, in addition to generating a regular sample, known as a base path, we also generate its offset path in a neighboring pixel, to compute a gradient between the pixel and its neighboring pixel. The radiance value from the neighboring pixel can be arbitrarily different from that of the pixel, resulting in a very noise gradient. To avoid such problem, it is desirable to construct the offset light path that is similar to the base light path. To achieve the goal, we commonly re-use prior sub light paths of the base light path, as shown in Fig. 5.2. While practical approaches of generating such offset light paths have been suggested, it cries further experiments to see how broadly and robustly this offset path generation works well.

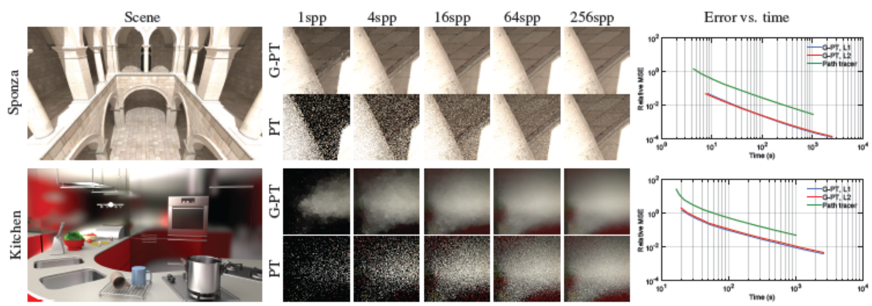


Figure 5.3: This image shows difference between path tracing and its gradient-domain version. Gradient-domain path tracing shows much better convergence rate over path tracing. This image is excerpted from the paper [KMA+15].

- **Screened Poisson reconstruction.** Once we compute gradients of the image, I^{dx} and I^{dy}, and sparse estimate of the final image, I^s, one can compute the final image I based on the following screened Poisson reconstruction problem [PGB03]:

$$\underset{I}{\operatorname{argmin}} \left(\left\| \begin{bmatrix} H^{dx} I \\ H^{dy} I \end{bmatrix} - \begin{bmatrix} I^{dx} \\ I^{dy} \end{bmatrix} \right\|_2^2 + \alpha \left\| I - I^s \right\|_2^2 \right), \tag{5.1}$$

where H^{dx} and H^{dy} are finite difference operators along X and Y directions, respectively. α is introduced to consider the sparse estimate of the final image I^2, and is commonly chosen as a small value (e.g., 0.2). This screened Poisson reconstruction method is shown to be unbiased [KMA+15]. Nonetheless, L^1 reconstruction shows visually better results over the L^2 reconstruction, while L^1 is biased.

Fig. 5.3 shows difference between path tracing and its gradient-domain path tracing in terms of different samples and running time.

Extensions. The gradient-domain rendering framework has been extended into different directions including the temporal domain to handle animation [**?**] and supporting bi-directional path tracing.

Gradient-domain image processing. Many image processing techniques work in spatial or frequency domains. However, working in the gradient-domain can be more natural [BZCC10]. Many applications including sharping, image deblocking, object insertion can be done nicely (Fig. **??**). This approach can be further studied, since human visual system is more sensitive to local contrast rather than pixel values themselves, and gradients arising from edges, textures,

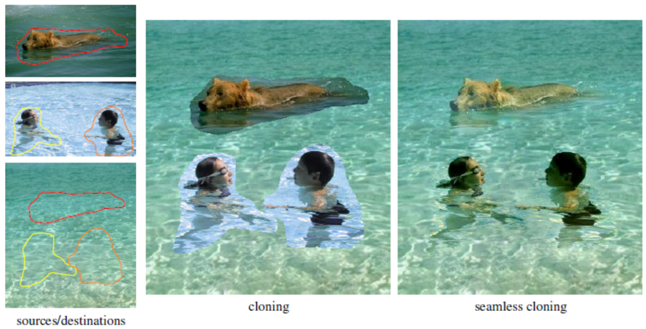


Figure 5.4: This shows simple cloning of an object into another image and seamless cloning in the gradient-domain by using the Poisson reconstruction. This image is excerpted from a slide of the work [PGB03].

shading, artifacts (e.g., noise and compression artifacts) provide various information on the images.