A Close Look At Path Space

Zichen Wang

Summer 2023

1 Introduction

The notion of path space and the path integral formulation of the light transport equation was first introduced by Veach in 1997 [3]. Since then, it has shown up in numerous important works [1][2][5][4]. However, due to its abstract nature, the concept of path space shows up more often in theory than in practical computation, for one rarely knows how to directly compute inside the path space. This manuscript thus tries to study the computational side of the path space.

2 Methods

2.1 A First Look At Path Space

To start off, we can model the scene as a camera at position $x_0 \in \mathbb{R}^3$, a point light at position $z \in \mathbb{R}^3$, and a collection of objects as 2-manifolds $\mathcal{M} = \bigcup_i \mathcal{M}_i$. For simplicity reasons, we consider only a single point light for now, but it should not be difficult to generalize to multiple points lights or area lights after the reader has gone through the manuscript.

Then, a *light path* describes how a ray of light bounces between objects inside the scene. Formally, we define a light path of length k to be a vector $(x_0,\ x_1,\ ...,\ x_k,\ z)$, where each $x_i \in \mathcal{M}_j$ for some index j. Note that we always start from the camera x_0 and end at the point light z.

The *path space of length* k is the set of all light paths of length k, that is,

$$\Omega_k = \{(x_0,\ x_1,\ ...,\ x_k,\ z) : x_i \in \mathcal{M}\}$$

The *path space* is then

$$\Omega = \bigcup_{k=1}^{\infty} \Omega_k$$

2.2 Computing The Velocity

Recall that the path space of length k is the union of the scene manifold for k times

$$\Omega_k = \underbrace{\mathcal{M} \cup \mathcal{M} \cup \ldots \cup \mathcal{M}}_{k \text{ times}}$$

And a path $\vec{x} \in \Omega_k$ can be expressed as $\vec{x} = (x_1, x_2, \ldots, x_k)$, where $x_i \in \mathcal{M}$. Note here x_i refers to an abstract point on the manifold. The velocity is then

$$\frac{\partial \vec{x}}{\partial p} = (\frac{\partial x_1}{\partial p}, \frac{\partial x_2}{\partial p}, \ldots, \frac{\partial x_k}{\partial p})$$

Thus it suffices to know how to compute the derivative of a point x_i on the manifold $\mathcal{M}$, and this is something that we are very familiar with.

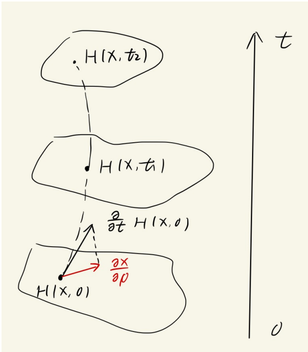

Figure 1: Illustration of an evolving manifold.

An evolving manifold can be characterized as a homotopy

$$H : \mathcal{M} \times \mathbb{R} \to \mathbb{R}^3$$

where $\mathcal{M}$ is the initial manifold (material space) when the parameter $p = p_0$. $H(\cdot, p)$ describes the manifold when the parameter is p, and $H(x, \cdot)$ describes the trajectory of a point $x \in \mathcal{M}$ as the parameter changes. The velocity of x on this evolving manifold is then $\frac{\partial}{\partial p}H(x, p_0)$. Eventually, we want the velocity of x on $\mathcal{M}$, so we project the velocity of the evolving manifold back onto the manifold. That is,

$$\frac{\partial x}{\partial p} = \frac{\partial H(x, p_0)}{\partial p} - \langle \frac{\partial H(x, 0)}{\partial p}, n(x) \rangle n(x)$$

In practice, say $x = (a, b, c)$ in $\mathbb{R}^3$, then $\frac{\partial}{\partial p}H(x, p_0) = (\frac{\partial a}{\partial p}, \frac{\partial b}{\partial p}, \frac{\partial c}{\partial p})$. The normal is simply the surface normal. In other words, we directly compute the derivative and project it back onto the tangent plane. We introduce the concept of the evolving manifold here because p might also change $\mathcal{M}$. Note that when $\mathcal{M}$ is independent of p, the derivative must be a tangent vector over the surface and no projection is needed. Finally, the path velocity we discuss here is similar to the scalar normal velocity in [5], albeit we rephrase the language from continuum mechanics to topology.

2.3 Example 1

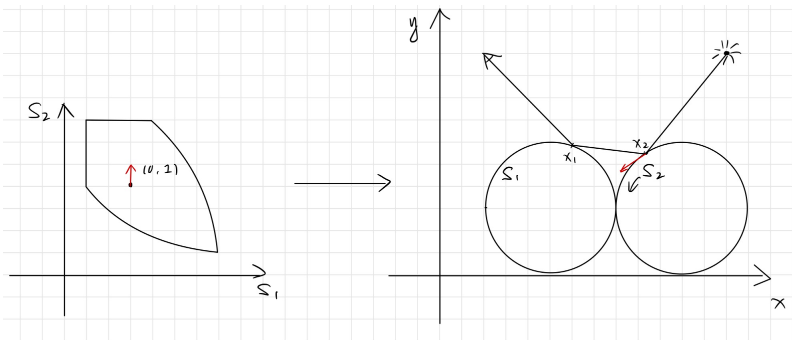

Figure 2: Example of computing path velocity

Consider the 2D scene with 2 spheres S_1, S_2 touching each other horizontally. We can parameterize each sphere by its radian. The path space of length 2 is then 2-dimensional. Now given a path from the camera to a point $x_1 \in S_1$ to a point $x_2 \in S_2$ to the point light, we can compute analytically its position in the path space. If we assign a velocity of $(0, 1)$ in the path space, this would be rotating S_2 counter-clockwise in $\mathbb{R}^2$ (the image in the wrong direction). x_1 remains static, so the velocity of x_1 is 0; x_2 rotates and has a velocity (v_x, v_y) tangential to S_2. Already we see $(0, 0, v_x, v_y)$ is the velocity of the path. We can further compute the local coordinate of $(0, 0,), (v_x, v_y)$ at x_1, x_2, which is simply the dot product of the velocity vector with the tangent vector. This tells us how to map $(0, 0, v_x, v_y)$ back to the path space and get exactly $(0, 1)$.

2.4 Example 2

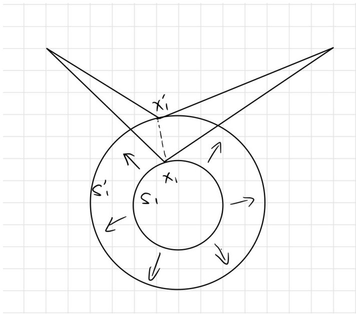

Figure 3: Another Example of computing path velocity

If you are not yet convinced by the above example, consider the scene where a sphere increases its radius with time. A path that goes from the camera to $x_1 \in S_1$ to the point light would have x_1 move along the normal direction at the surface. By our theory, the path velocity should be 0, since the velocity of x_1 in $\mathbb{R}^2$ is orthogonal to the tangent plane. And this is indeed the case: if we map a later point x_1' back to the initial manifold (material space), we see that the point stays at x_1 and does not move at all!

3 Conclusion

In this manuscript, we tried to define and justify the velocity of a path inside the path space. More generally, given that the path space is the Cartesian product of 2-manifolds, it seems that for any local property that we can compute on a 2-manifold, we can also compute in the path space. The vectors and manifolds are represented in $\mathbb{R}^3$, but we can transform them to path space by computing a local parametrization (tangent plane) of the manifold. The Cartesian product of the tangent planes would then give us a local parametrization of the path space in high dimensions, which enables the computation of local properties in the path space.

References

[1] Wenzel Jakob and Steve Marschner. "Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport". In: *ACM Trans. Graph.* 31.4 (July 2012). ISSN: 0730-0301. DOI: 10.1145/2185520.2185554. URL: https://doi.org/10.1145/2185520.2185554.

[2] Fujia Su, Sheng Li, and Guoping Wang. "SPCBPT: Subspace-based Probabilistic Connections for Bidirectional Path Tracing". In: 2022. URL: https://doi.org/10.1145/3528223.3530183.

[3] Eric Veach. "Robust Monte Carlo Methods for Light Transport Simulation". PhD thesis. Stanford University, 1997.

[4] Beibei Wang, Miloš Hašan, and Ling-Qi Yan. "Path Cuts: Efficient Rendering of Pure Specular Light Transport". In: *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: 10.1145/3414685.3417792. URL: https://doi.org/10.1145/3414685.3417792.

[5] Cheng Zhang et al. "Path-Space Differentiable Rendering". In: *ACM Trans. Graph.* 39.4 (2020), 143:1–143:19.