
layout: post title: "基于梯度域渲染方法的分析 (表面) " subtitle: "Gradient Domain based rendering" author: "ls" header-img: "img/post-gradient-cover.jpg" header-mask: 0.2 tags:

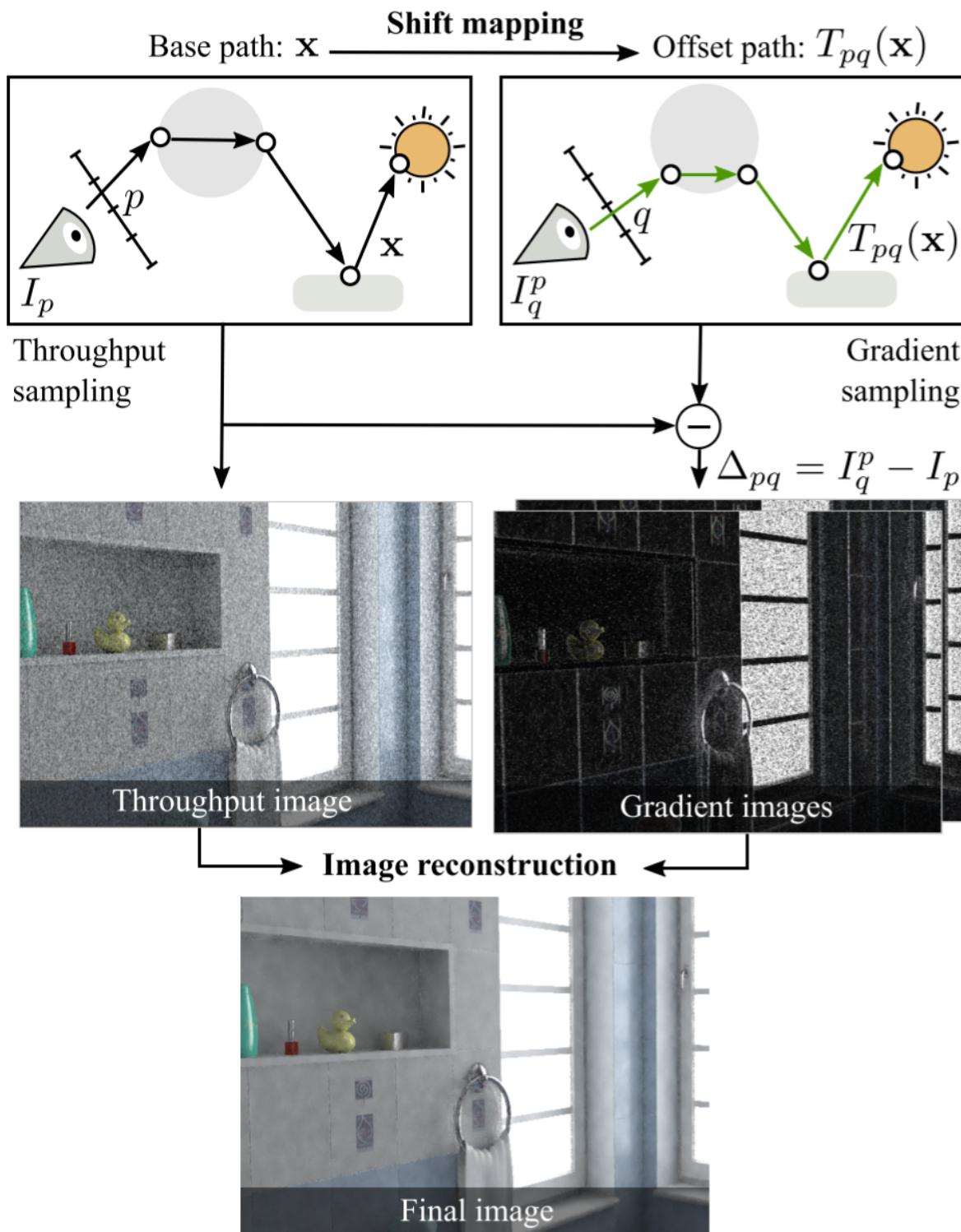
- MonteCarlo
 - Gradient Domain
 - Rendering
-

一、梯度域方法概述

梯度域方法是一种“降噪”方法，其主要目的是加快基于 MonteCarlo 采样的收敛效率，基于梯度域的方法主要通过以下流程实现：

1. 计算初始图像
2. 计算梯度图像
3. 对初始图像和梯度图像进行图像重构
4. 得到最终结果

其具体例子可见图1



二、关于基于梯度域方法的三大要素

初始渲染算法

初始渲染算法决定了路径的生成和初始图像的计算。

\$Shift\$函数

Shift 函数决定如何根据 base path 来生成 offset path。

雅可比行列式

雅可比行列式用来校正在 Offset Path 相对于 Base Path 的路径密度改变。

其中 Offset 路径的贡献值的计算公式如下：

$$\begin{aligned}
 & \$ \begin{aligned} l_{j+1} & \&= \int_{\Omega} h_{j+1}(\bar{x}) f^{}(\bar{x}) d\mu(\bar{x}) \\
 & \&= \int_{T_{1,0}} (-1)^j (\Omega) h_{j+1} \left(T_{1,0}(\bar{x}) \right) f^{}(\bar{x}) d\mu(\bar{x}) \\
 & \&= \int_{\Omega} h_{j+1} \left(T_{1,0}(\bar{x}) \right) f^{}(\bar{x}) d\mu(\bar{x}) \\
 & \&= \frac{\int_{\Omega} h_{j+1} \left(T_{1,0}(\bar{x}) \right) f^{}(\bar{x}) d\mu(\bar{x})}{\int_{\Omega} h_j(\bar{x}) f^{}(\bar{x}) d\mu(\bar{x})} \\
 & \&= \end{aligned} \tag{1} \\
 & \$
 \end{aligned}$$

则路径 \overline{x} 的路径差 $|l_j - l_{j+1}|$ 为：

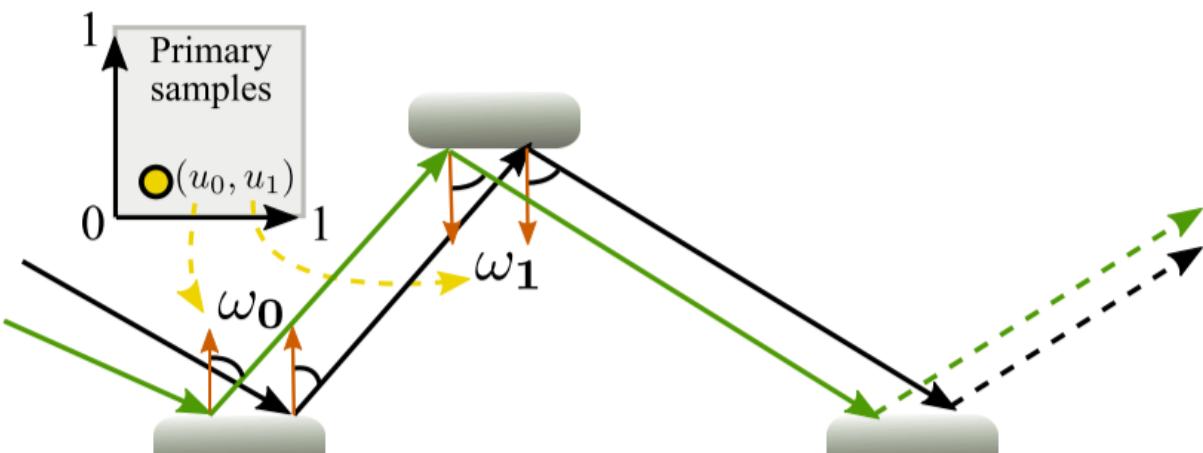
$$\begin{aligned}
 & \$ \begin{aligned} l_j - l_{j+1} & = \int_{\Omega} h_j(\bar{x}) \left[f^{}(\bar{x}) - \frac{\int_{\Omega} h_{j+1}(\bar{x}) f^{}(\bar{x}) d\mu(\bar{x})}{\int_{\Omega} h_j(\bar{x}) f^{}(\bar{x}) d\mu(\bar{x})} \right] d\mu(\bar{x}) \\
 & = \end{aligned} \tag{2} \\
 & \$
 \end{aligned}$$

三、Shift 函数

根据 [GD-Path Tracing](#) 中的频率分析，offset path 与 base path 关联性越大，其重构后的方差就越小，最终呈现出的结果也就越好。

1、随机数复用

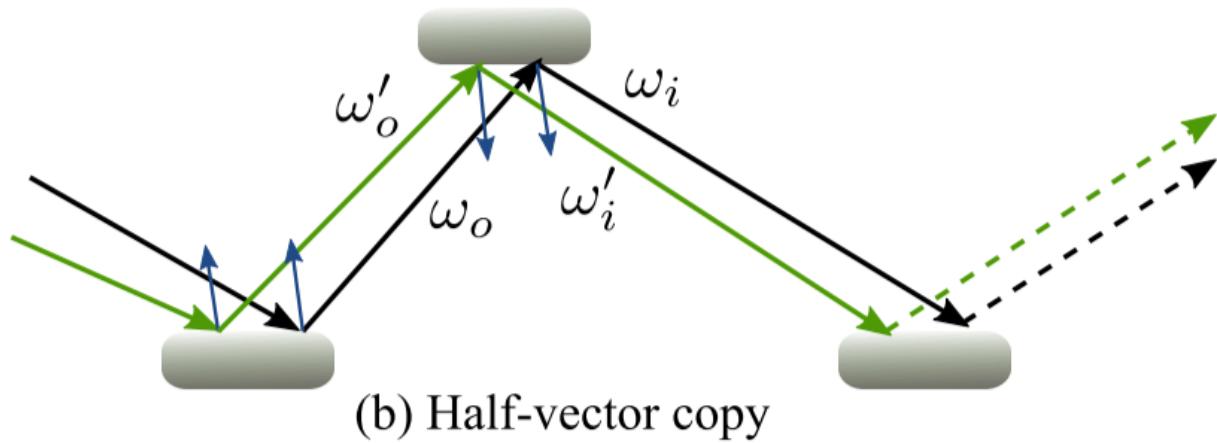
像素 i 产生路径 \overline{x} 所使用的随机序列记为 \overline{U} ，重复使用该随机序列来产生对应的 offset path \tilde{x} 。



(a) Random sequence replay

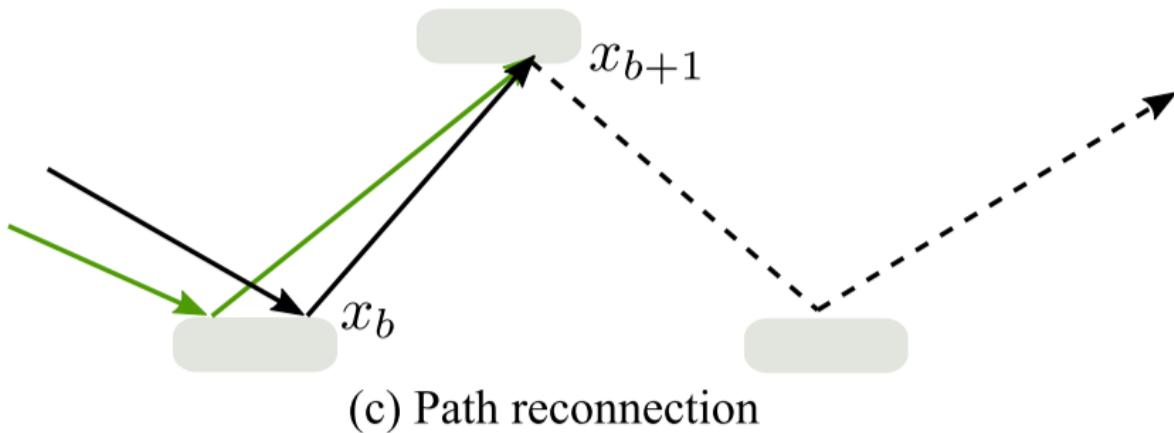
2、半向量保留

保留 base path 中每个顶点的半向量（局部空间），根据半向量，入射向量，可以直接计算得到出射方向。

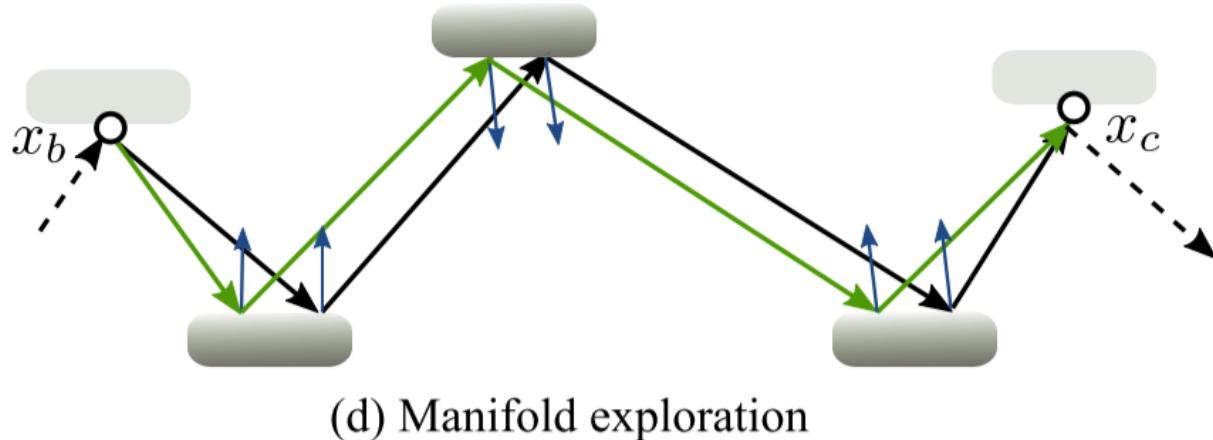


3、路径重连（Vertex reconnection）

只在 S 表面上进行 offset (S 表面能量分布的lobe过窄，直接连接其路径能量概率可能携带能量为0)，对于 D 表面，直接重连这些 D 顶点。



4、Manifold Perturbation （详见 Manifold Exploration）

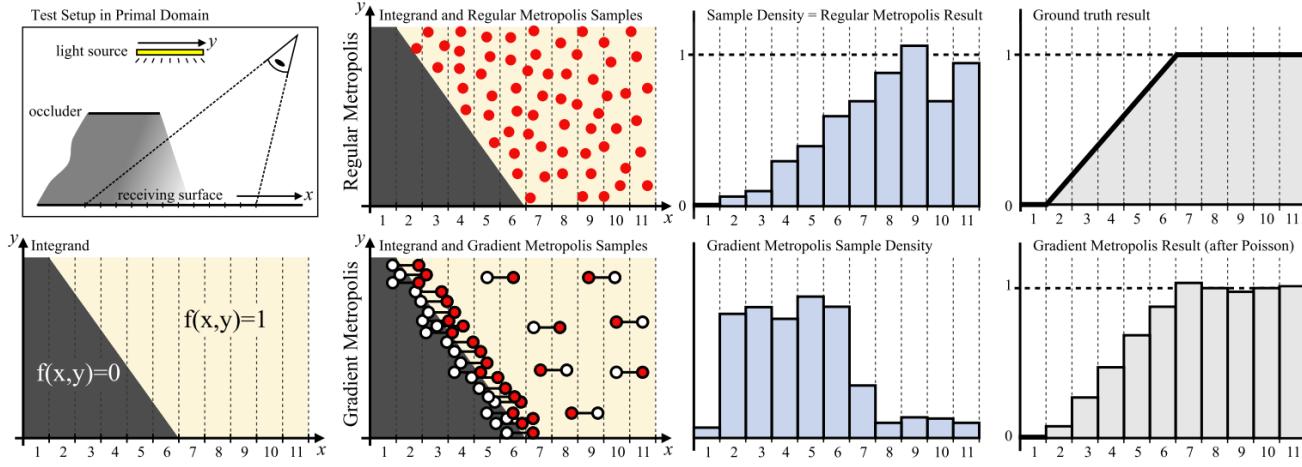


几种基于 Gradient - Domain 的方法在构建 offset path 上都采用了上述的一种或多种（主要为后三种）Shift 函数，并根据选择的基础算法不同进行了一定的修改。

四、不同算法中Shift函数的应用以及具体调整

1、GD - MLT

使用 Markov Chain Monte Carlo 的方法对目标函数 $f(\overline{x})$ 进行采样。传统的 MLT 方法对于目标函数只考虑了像素路径的能量贡献，但是在 GD - MLT 中的目标函数还考虑了梯度的变化。



在梯度变化大的地方着重采样能明显提高效率，上图中，GD - MLT 将采样重心更多的放在了阴影处。

1.1、Shift 函数

1. 对于像素 i 的 base path \overline{x} 寻找前三个 D 顶点 X_a, X_b, X_c
2. 根据像素 j (i 像素在屏幕空间的偏移像素) 从 X_a 处生成一条新的光线
3. 在 $X_a \sim X_b$ 的 Specular Chain 中进行 ray tracing，以此生成新的 $X_a \sim X_b^{off}$ 路径。
4. 对于 $X_b^{off} \sim X_c$ 路径，采取 Manifold Exploration 来确保 X_c 点不变

上述过程即为 **Manifold Perturbation**。

1.2、目标函数

定义 \overline{z} 为：

$$\overline{z} = \left\{ \overline{x}, f_x, f_y \right\}$$

其中 f_x 与 f_y 为 base path 在 x, y 方向上对应的 offset path。则 GD-MLT 的目标函数为

$$f(\overline{z}) = \left| f \right| \overline{z} + \alpha \left(\frac{1}{4} \left| f^2 \right| \overline{z} \right)^2$$

其中第一项考虑了路径的梯度变化，第二项考虑了路径的能量贡献，而 α 代表自定义系数。

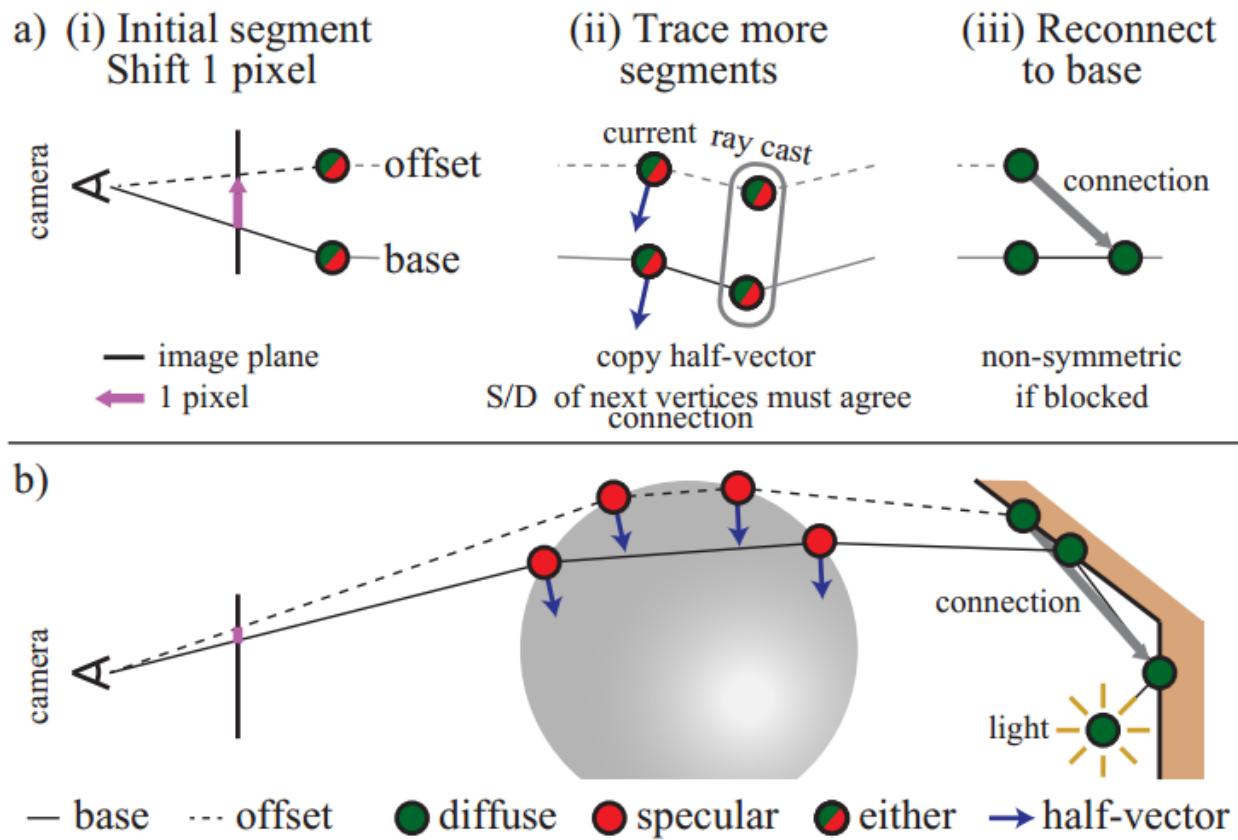
2、GD - PT

2.1、Shift 函数

不采用 manifold exploration，而是采用半向量保留和路径顶点重连的方案。设 j 为像素 i 在屏幕空间上移动 1 单位后的像素。从 j 开始进行 ray tracing。分为以下两种情况：

1. base path 中的当前顶点或者下一个顶点类型为 S，则在半向量保留的前提下进行 ray tracing 并生成下一个 offset 顶点

2. offset path 的当前顶点以及 base path 的当前及下一个顶点的类型都为 D 类型，则将当前 offset path 的顶点与 base path 中的下一个顶点相连



为了降低误差，需要 offset path 和 base path 的关联性较高，同时也要避免路径的能量分布极具变化（即 offset path 携带的能量可能为0）

定义 base path 上的二元组 $P = (V_c^b, V_n^b)$ ，其中 c 为路径中的当前点， n 为下一个顶点， V 代表顶点类型（D 或者 S）。同时对于当前 offset 顶点记为 V_c^o ，则路径顶点重连一共分为四种情况：

- \$(S_c^o, D_n^b)\$ 该处的重连会使路径能量变得极小（完美 Specular 时为0），这是由于 SD 相连得到的出射方向与 S 处的入射方向符合 S 顶点上 BSDF 分布的概率极小
- \$(S_c^o, S_n^b)\$ 与情况1相同
- \$(D_c^o, D_n^b)\$ D 上的方向变换不会引起 BSDF 的太大变化
- \$(D_c^o, S_n^b)\$ 在 S 上的能量会发生明显变化

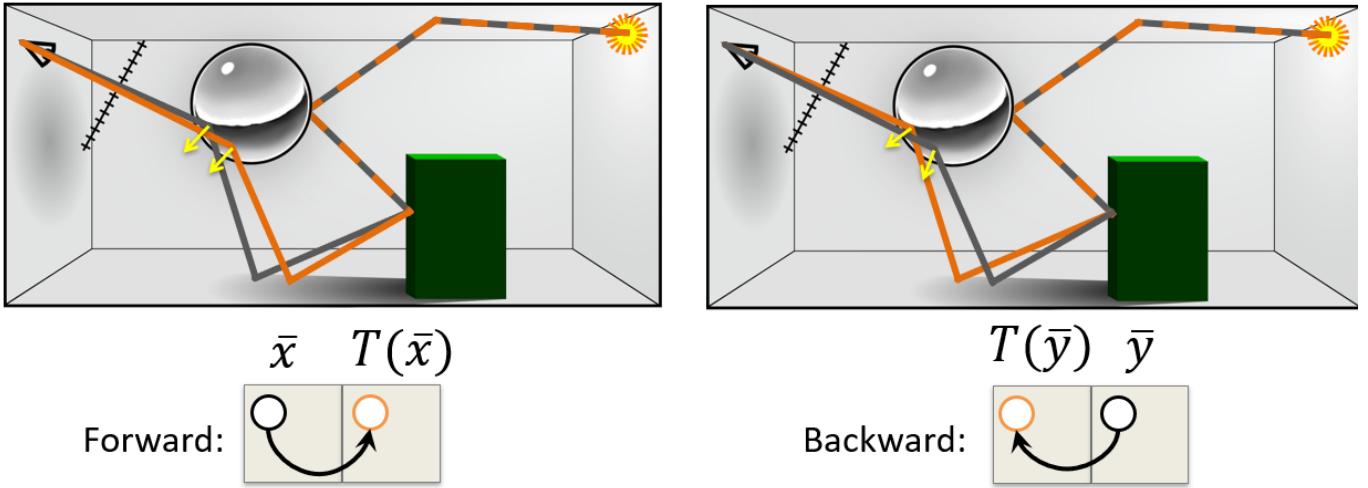
综上所示，只有在 base path 中遇到 DD 类型的两个顶点时才会发生路径顶点重连

2.1、多重重要性采样

对于像素 i 其对应的路径 \overline{x} 与 offset path \overline{y} 可以从两个方向得到：

- 正向采样 从 i 采样得到 \overline{x} ，从 $i+1$ 得到 offset path $\overline{y} = T(\overline{x})$
- 反向采样 从 $i+1$ 采样得到 \overline{y} ，然后得到 $\overline{x} = T^{-1}(\overline{y})$

可以这两种不同方向的生成方式视为两种不同的采样策略，然后使用 MIS 连接起来，这样还解决了当雅克比值过大时的进度问题（从低能量区转换到高能量区）



前向采样 MIS 权值为:

$$\$ w_{\text{forward}}(\overline{x}) = \frac{p(\overline{x})}{p(\overline{x}) + p(T(\overline{x}))} \left| \frac{d \mu(T(\overline{x}))}{d \mu(\overline{x})} \right| \$$$

反向采样 MIS 权值为:

$$\$ w_{\text{backward}}(\overline{y}) = \frac{p(\overline{y})}{p(\overline{y}) + p(T^{-1}(\overline{y}))} \left| \frac{d \mu(T^{-1}(\overline{y}))}{d \mu(\overline{y})} \right| \$$$

3、GD-BDPT

3.1、Shift 函数

在 BDPT 中，每次迭代都生成一条相机子路径和一条光源子路径。记相机子路径为 x^E ，光源子路径为 x^L

$$\$ x^E = \{x_0^E, x_1^E, \dots, x_{s-1}^E\} \$ \$ x^L = \{x_0^L, x_1^L, \dots, x_{t-1}^L\} \$$$

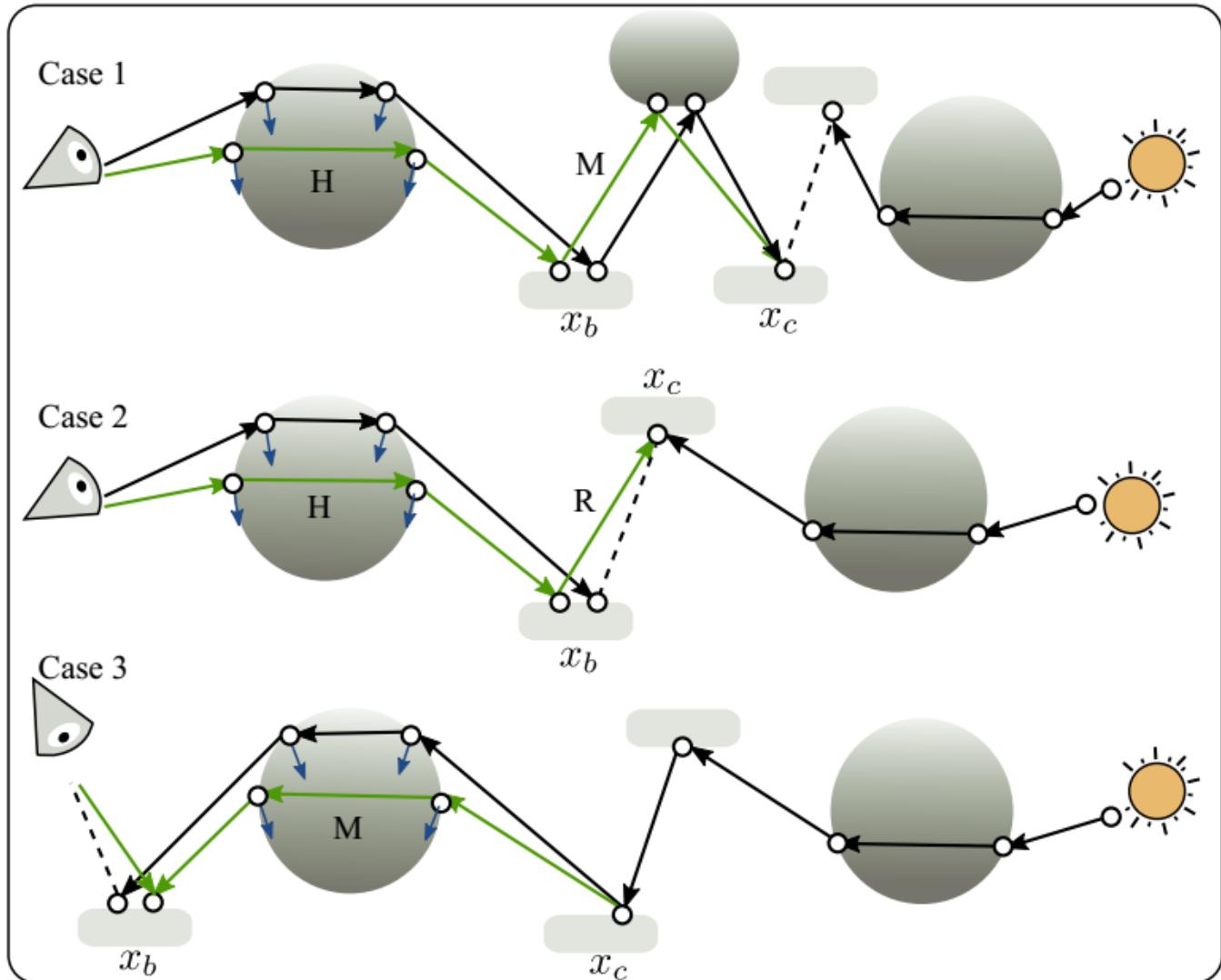
BDPT 中一次迭代会生成多条路径（对于子路径的不同连接方式）若每种连接方式产生的路径都产生对应的 offset path 将会产生大量的时间开销，所以 GD-BDPT 中约定：

S 点不作为连接点（在**S**处连接的 **BSDF** 值非常小，所以去掉这一部分影响不会很大）

对于连接后形成的路径 \overline{x} ，使用 Manifold Perturbation 进行扰动 对于前三个连续的 D 顶点 x_a, x_b, x_c ，GD-BDPT 规定 x_a 为相机，则有以下三种情况

1. $x_b, x_c \in x^E$ ，则对 $x_a \sim x_b \sim x_c$ 进行 Manifold Perturbation
2. $x_b \in x^E, x_c \in x^L$ 由于 S 点不能做连接点，所以 $c = b + 1$ 。此时由于 Manifold Perturbation 也在第一段 S 链也进行了半向量保留，且 x_b, x_c 为连续的 DD 点，所以退化成 GD-PT 中的情况
3. $x_b, x_c \in x^L$ 同理，所以 $b = a + 1$ ，该情况下，整个路径由 light tracing 得到

情况 1 2 的 Shift 操作在 x^E 进行，分别为 $x_a \sim x_b \sim x_c$ 和 $x_a \sim x_b$ 段。而情况 3 的 Shift 则全在 x^L 中进行



(b) Gradient-domain bi-directional path tracing

3.2、多重重要性采样

采用和 GD-PT 一样的策略，将两种不同方向的 offset path 生成方法视为两种不同的采样策略，且同时和 BDPT 中基于多种连接方案的 MIS 合并，得到：

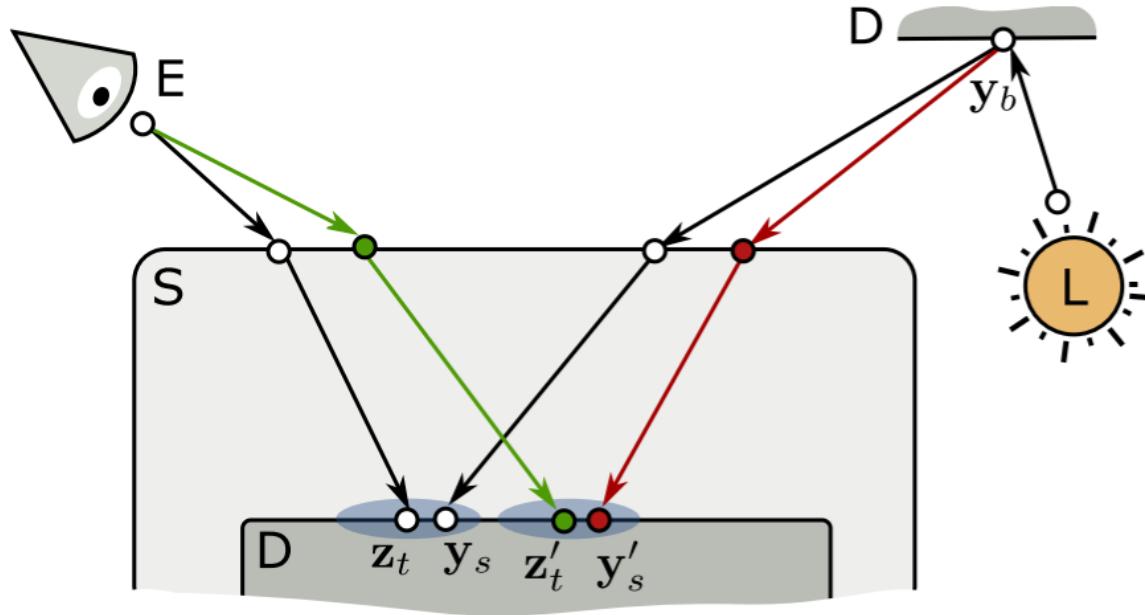
$$\$ w_{ij;st}(\overline{x}) = \frac{p_{s,t}(\overline{x})}{\sum_{k=0}^{s+t} p_{k,s+t-k}(\overline{x}) + p_{k,s+t-k}(T_{ij})(\overline{x})|T_{ij}|^{\prime}}, \$$$

其中 $\sum_{k=0}^{s+t} p_{k,s+t-k}(\overline{x})$ 与不同的路径连接方式有关，而 $p_{k,s+t-k}(T_{ij})(\overline{x})|T_{ij}|^{\prime}$ 则与梯度有关。

4、GD-PM

4.1、Shift 函数

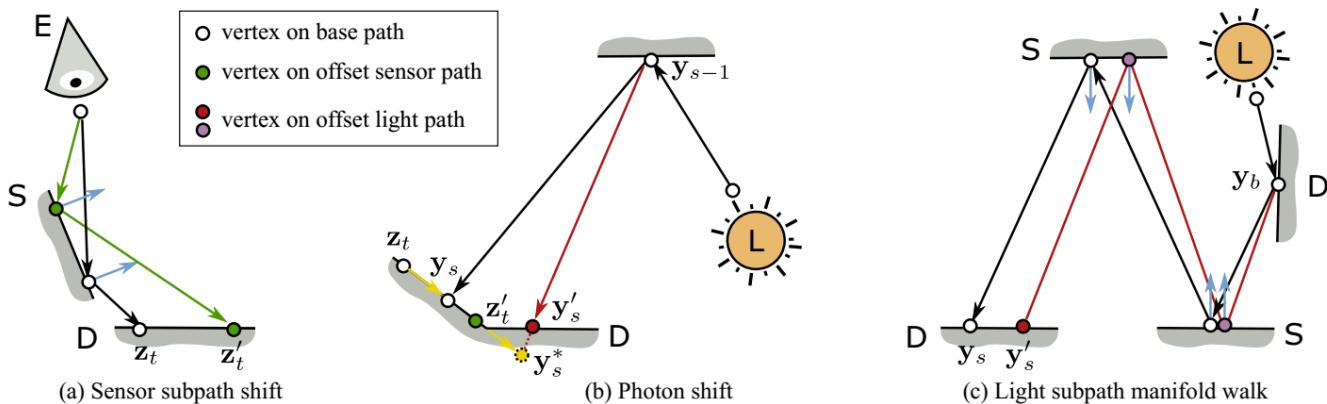
基于 PM 的算法存在两条子路径，相机路径 x^E 和光子路径 x^L ：



对于 $x^E = \{x_0^E, x_1^E, \dots, x_{s-1}^E\}$, x_{s-1}^E 必定落在 D 顶点上且 $x_0^E \sim x_{s-1}^E$ 之间必定不存在 D 表面 (PM 在生成 Visible Point 时的定义), 所以 x^E 的路径类型为 LS^D 。对于 $x^L = \{x_0^L, x_1^L, \dots, x_{t-1}^L\}$, x_{t-1}^L 为光子, 必定落在 D 上, 但 $x_0^L \sim x_{t-1}^L$ 之间可能存在 D , 所以路径类型为 $L(S|D)^{\cap}D$ 。

综上对于 x^E , 使用半向量保留和路径顶点重连, 由于 x_{s-1}^E 处使用 Density estimation, 所以不需要连续的 DD 路径 (或者说已经存在了 DD 路径, 因为后续点是一个光子, 而光子必定落在同一个 Diffuse 表面上)。对于 x^L 进行 manifold perturbation。

实际上, 在 GD-PM 中, 对于 x^E 的抖动会产生新的端点 $x_{s-1}^{E,\text{off}}$, 将 x_{s-1}^E 和 x_{t-1}^L 的相对关系不变, 以此来产生 $x_{t-1}^{(\prime, L,\text{off})}$, 由于 $x_{t-1}^{(\prime, L,\text{off})}$, 不一定还在同一个表面上, 所以连接 x^L_{t-2} 与 $x_{t-1}^{(\prime, L,\text{off})}$, 然后进行 ray tracing 得到真正的 offset 光子 $x_{t-1}^{(L,\text{off})}$, 然后在 $x_{t-1}^{(L,\text{off})} \sim x_b^L$ 上进行 manifold exploration。



值得注意的是, 若 x^L_{t-2} 为 D (全局间接光子图), 则不再需要后续的对 x^L 的 manifold exploration, 而是直接相连即可。

5、GD-VCM

传统的 VCM 算法将 BDPT 和 PM 以 vertex connect 和 vertex merge 结合起来。

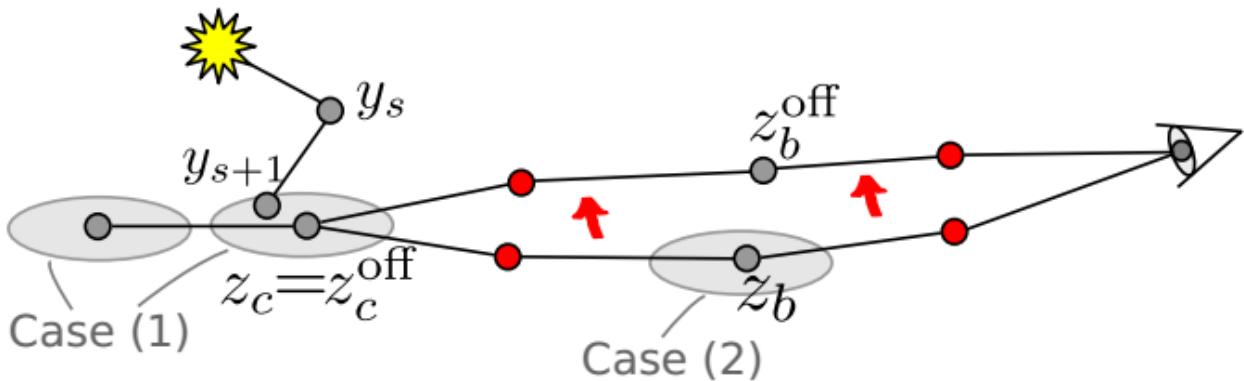
5.1、Shift 函数

G-VCM 中同样存在两条路径 x^E 和 x^L 。对于 x^E , 采用 manifold perturbation

VC 阶段的策略和 G-BDPT 基本类似, 只是在 MIS 上有所不同 (考虑了 VM)。

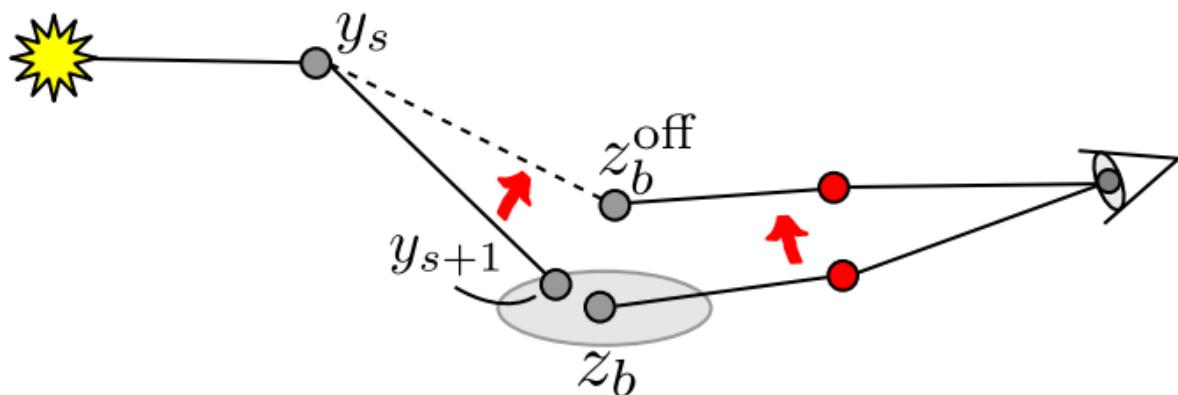
VM 阶段, 对于 x_a^E, x_b^E, x_c^E 处的 merge 操作有两种可能 (PM 的 merge 必须在 D 表面上进行) :

1. 在 x_c^E 之后进行 merge 此种情况和 offset 之后的 S 链无关, 只需要直接相连即可。

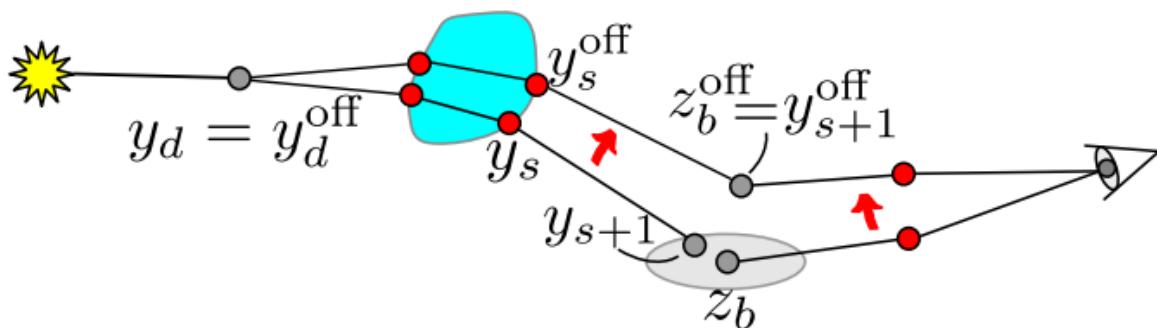


2. 在 x_b^E 处进行 merge, 此时需要考虑 x_{t-2}^L 的顶点属性

- x_{t-2}^E 为 D (全局间接光子图) 直接连接 x^L_{t-2} 与 x^L_{t-1} 形成了 DD



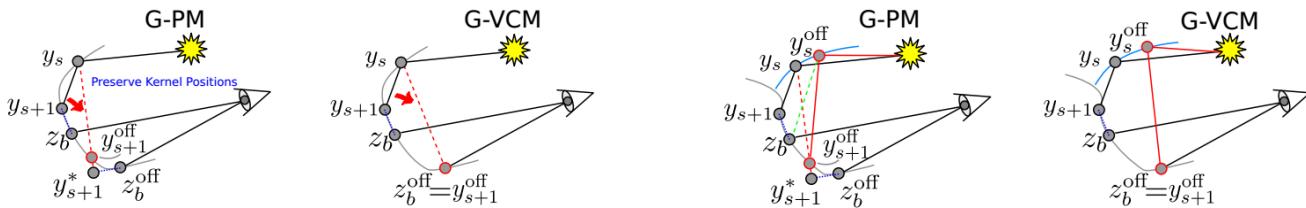
- x_{t-2}^E 为 S (焦散光子图) 将 $x_{t-1}^{L,\text{off}}$ 设为 $x_b^{E,\text{off}}$, 然后在 $x_b^L \sim x_{t-1}^{L,\text{off}}$ 进行 manifold exploration



5.2、和 GD-PM 的区别

G-PM 中，根据 $\$x_{\{t-1\}}^L$ 与 $\$x_{\{s-1\}}^E$ 的关系，形成新的 $\$x_{\{t-1\}}^{\{L, off\}}$ ，然后根据投影得到真正的 offset 光子 $\$x_{\{t-1\}}^{\{L, off\}}$

而在 G-VCM 中，直接将 $\$x^{\{L, off\}\{t-1\}}$ 设置为 $\$x^{\{E, off\}\{s-1\}}$ ，两者在收敛后是等价的，因为基于 PM 的算法具有一致性。



五、雅可比行列式

该部分的内容受限于本人的数学水平，可能会有较多理解不到位或者错误的地方，期待各位读者斧正！

1、基于 Manifold Exploration

限制项:

```

$$ \mathbf{c}_i \left( \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1} \right) = \mathbf{o} $$

$$ \mathbf{c}_i \left( \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1} \right) = T \left( \mathbf{x}_i \right) \\ h \left( \mathbf{x}_i, \overrightarrow{\mathbf{x}_i \mathbf{x}_{i-1}}, \overrightarrow{\mathbf{x}_i \mathbf{x}_{i+1}} \right) $$

$ h(\mathbf{x}, \mathbf{v}, \mathbf{w}) = \frac{\eta(\mathbf{x}, \mathbf{v})}{\mathbf{v} + \eta(\mathbf{x}, \mathbf{w})} \frac{\eta(\mathbf{x}, \mathbf{w})}{\mathbf{w} + \eta(\mathbf{x}, \mathbf{v})} $$

```

其中 \mathbf{T}_{i} 为在 x_i 处的切平面转换矩阵，由 \mathbf{x}_i 处的切线和副法线构成。 \mathbf{h} 为该点的半向量。 \mathbf{o} 为一个二维向量，当 x_i 为 Perfect Specular 时，由于半向量和法线垂直，所以 $\mathbf{o} = \mathbf{0}$ 。

整条路径 \overline{x} 的 manifold 限制为:

$\$ \$ \backslash mathcal{S} _ \backslash mathrm{o} \} = \{ \overline{\mathbf{x}} \mid C(\overline{\mathbf{x}}) = \mathbf{o} \} \$ \$$

综上, 利用 manifold perturbation 得到的 offset path \widetilde{x} 与 base path \overline{x} 的雅克比可以表示为:

$$\begin{aligned} & \left(\frac{\partial \tilde{\mathbf{x}}_i}{\partial \tilde{\mathbf{O}}_k} \right)_{j,l} = \left(\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_j} \right)_{l,k} \left(\frac{\partial \mathbf{x}_j}{\partial \mathbf{O}_k} \right)_{i,l} \\ & \quad + \sum_m \left(\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_j} \right)_{m,l} \left(\frac{\partial \mathbf{x}_m}{\partial \mathbf{O}_k} \right)_{i,k} \end{aligned}$$

其中：

$\$ \$ \begin{aligned} & \left(\mathbf{x}_1, \dots, \mathbf{x}_{c-1} \right) \equiv \left(\mathbf{o}_1, \dots, \mathbf{o}_{c-1} \right) := \mathbf{O} \left(\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{c-1} \right) \end{aligned}$

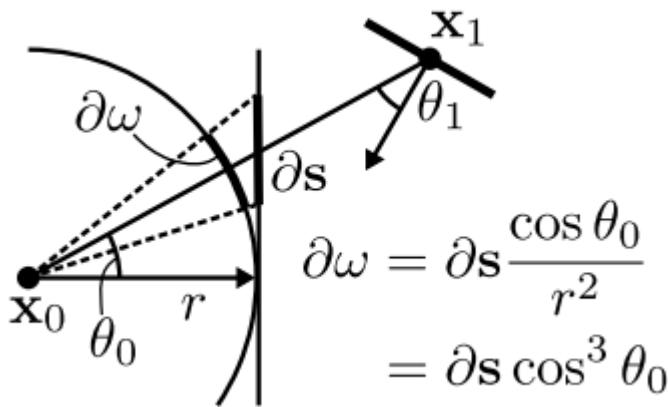
```
1}\right) & \equiv \left( \tilde{\mathbf{\sigma}}_1, \dots, \tilde{\mathbf{x}}_b, \dots, \tilde{\mathbf{o}}_{c-1} \right) := \tilde{\mathbf{O}} \end{aligned} $$
```

由于在 Manifold Exploration 中，对于同一对顶点 $\overline{\mathbf{x}}_i, \widetilde{\mathbf{x}}_i$ 采用了半向量保留，所以 $\overline{o}_i = \widetilde{o}_i$ ，所以在雅克比矩阵的中间项可以简化为 $\frac{\widetilde{\mathbf{x}}_b}{\overline{\mathbf{x}}_b}$

其中 $\$s\$$ 为屏幕空间上的像素位置

```
 $$ \begin{aligned} \frac{\partial \mathbf{x}\{b\}}{\partial \mathbf{s}} = \frac{\partial \mathbf{x}\{b\}}{\partial \mathbf{x}\{1\}} \frac{\partial \mathbf{x}\{1\}}{\partial \mathbf{s}} = \frac{\partial \mathbf{x}\{b\}}{\partial \mathbf{x}\{1\}} \frac{\partial \omega_0^{\perp}}{\partial \mathbf{x}\{1\}} & \&= \frac{\partial \mathbf{x}\{1\}}{\partial \mathbf{s}} = \frac{\partial \mathbf{x}\{1\}}{\partial \mathbf{x}\{0\}} \leftarrow \mathbf{x}\{1\} \rightarrow \mathbf{x}\{0\} \\ & \&= \frac{\partial \mathbf{x}\{1\}}{\partial \mathbf{s}} = \frac{\partial \mathbf{x}\{1\}}{\partial \mathbf{x}\{0\}} \leftarrow \mathbf{x}\{1\} \rightarrow \mathbf{x}\{0\} \end{aligned} $$
```

$$\$ \$ \frac{\partial \mathbf{x}\{1\}}{\partial s} = \frac{\left| \mathbf{x}\{1\} - \mathbf{x}\{0\} \right|^2}{\cos^3 \theta\{0\}} \cos \theta\{1\} \$ \$$$

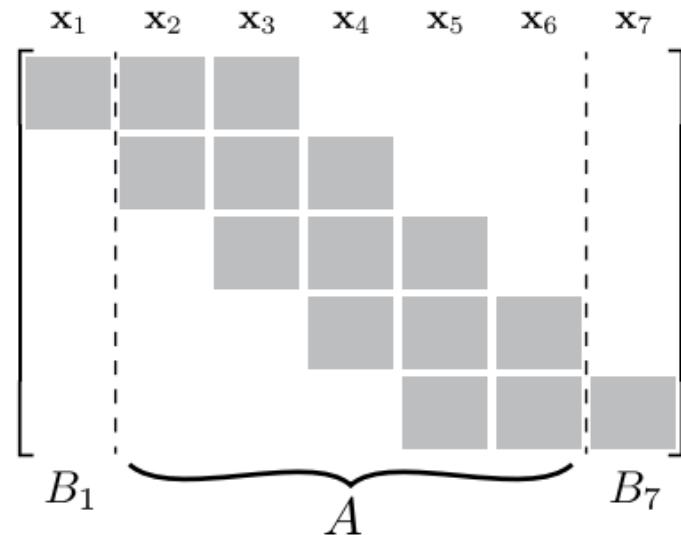


对于 $\frac{\partial \mathbf{s}}{\partial \tilde{\mathbf{x}}_b}$ 同理计算。

对于雅克比中的前后项，考虑路径的限制函数 S_o ，对其进行隐函数求导：

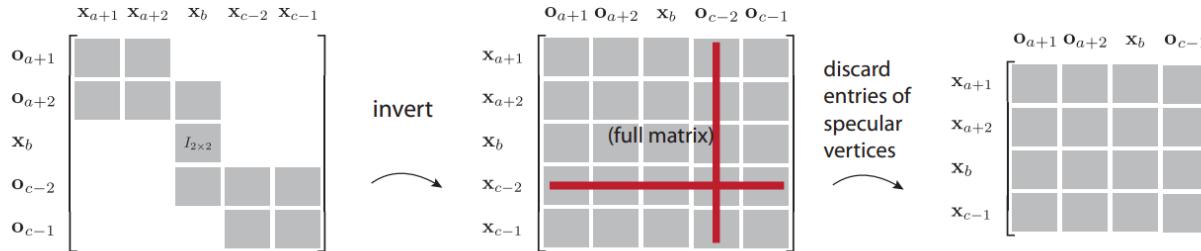
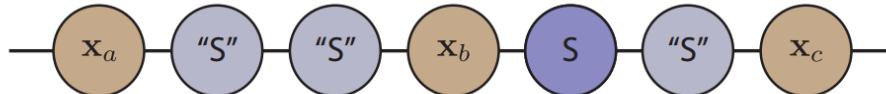
$$C(\bar{x}) = 0 \quad \text{where}$$

$$C = \begin{bmatrix} \mathbf{c}_2(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \\ \vdots \\ \mathbf{c}_6(\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7) \end{bmatrix} \quad \nabla C =$$



由于 $C_i = o_i$, 所以 A 矩阵就包含我们所需要的偏导, 只不过为倒数形式, 只需要将 A 矩阵求逆, 同时值得注意的是, 由于在 Perfect Specular 顶点时, $o = 0$ 恒成立, 所以需要舍弃这部分数据:

A perturbation with 3 glossy and 1 specular vertices



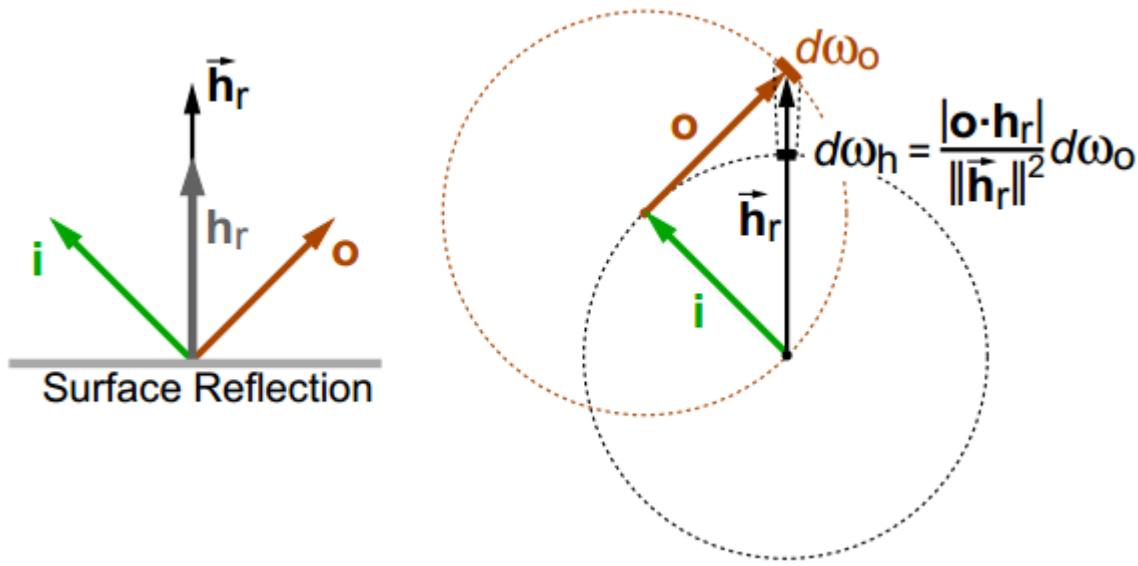
2、基于半向量保留和路径顶点重连 (GD-PT)

由于使用路径顶点重连, 所以重连后的路径的雅克比为 1, 不需要再进行计算, 只需要考虑其中的 S 链。

- 对于中间的 S 点, 用方向参数 ω_i 来表示 x_i , 则对于 base path 的顶点 \overline{x}_i 和 offset path 的顶点 \overline{y}_i , 其入射方向为 ω_i^x 和 ω_i^y , 对应的半向量为 h_i^x 和 h_i^y , 所以雅克比可以写成:

$$\frac{\partial \omega_i^y}{\partial \omega_i^x} = \frac{\partial \omega_i^y}{\partial h_i^x} \cdot \frac{\partial h_i^x}{\partial \omega_i^x} = \frac{\omega_i^y \cdot \mathbf{h}^y}{\omega_i^x \cdot \mathbf{h}^x}$$

其中半向量与入射光线的比, 可以根据[这里](#)得到几何上的解释。(关于代数角度上的推导, 还有些困惑, 所以这里不再列出)



- 对于路径顶点重连后的雅克比，记 base path 中连续的 DD 点为 x_1^x 和 x_2^x ，则雅克比为：

$$\left| \frac{\partial \omega_i^y}{\partial \omega_i^x} \right| = \left| \frac{\partial \mathbf{x}_2^y}{\partial \mathbf{x}_1^y} \right| = \frac{\cos \theta_2^y}{\cos \theta_1^y} \frac{\left| \mathbf{x}_1^x - \mathbf{x}_2^x \right|^2}{\left| \mathbf{x}_1^y - \mathbf{x}_2^y \right|^2}$$

这里使用了微元面和微元立体角之间的关系 ($d\omega = \frac{\cos \theta}{r^2} dA$)。值得注意的是，这个关系在不同类型 Shift 函数的雅克比的计算中频繁出现，是连接方向和点的“桥梁”

3、GD-PM 中的雅克比

在 GD-PM 中，相机路径的雅克比与 GD-PT 中的计算方法类似，对于光子路径，由于 offset 光子是根据相对位置产生，所以在该点的雅克比为：

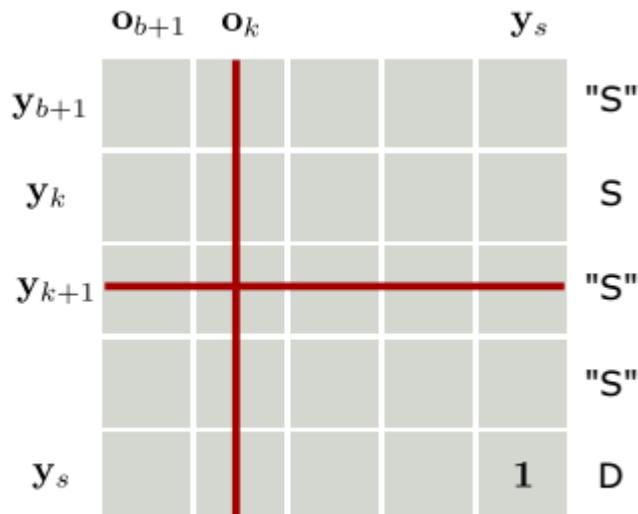
$$\left| \frac{\partial \mathbf{x}_{t-1}^{L, off}}{\partial \mathbf{x}_{t-1}^{L, off}} \right| = \left| \frac{\partial \mathbf{x}_{t-1}^{L, off}}{\partial \mathbf{x}_{t-1}^{\prime, L, off}} \right| = \frac{G(\mathbf{x}_{t-2}^L, \mathbf{x}_{t-1}^{\prime, L, off})}{G(\mathbf{x}_{t-2}^L, \mathbf{x}_{t-1}^{L, off})}$$

其中，

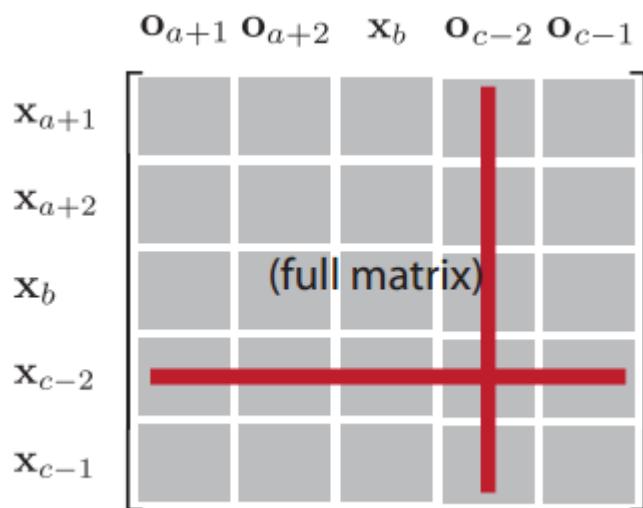
$$\left| \frac{\partial \mathbf{x}_{t-1}^{\prime, L, off}}{\partial \mathbf{x}_{t-1}^{L, off}} \right| = 1$$

由于 G-PM 在 x_{t-2}^L 为 D 时，直接进行了相连，不再对光子路径进行抖动，所以整个光子路径的雅克比即为 $\left| \frac{\partial \mathbf{x}_{t-1}^{L, off}}{\partial \mathbf{x}_{t-1}^{L, off}} \right|$ 。

当 x_{t-2}^L 为 S 时，需要进行 Manifold Exploration，可以采用 Manifold Perturbation 中雅克比相似的计算方法，但是值得注意的是，这里关于 $\left| \frac{\partial \mathbf{x}}{\partial \omega} \right|$ 的计算有些不同，对于 Perfect Specular 部分的舍弃不一致（关于这部分还在思考中）：



- GD-PM



- Manifold

六、Shift函数的开销和对比

6.1、Manifold

设路径长度为 k , x_a, x_b, x_c 为 D , 有 $k-3$ 个 S 顶点 则在 $x_a \sim x_b$ 阶段

- $(b-a-1)$ 次 ray tracing

在 $x_b^{off} \sim x_c$ 阶段 (N 为优化迭代次数)

- $N \times 1$ 次 牛顿迭代所需要的优化矩阵计算
- $N \times (c-b-1)$ 次 ray tracing

对于雅克比计算中的 $\left[\frac{\partial \mathbf{o}}{\partial \mathbf{x}} \right]$ 可以在最后一次迭代中, 通过优化矩阵计算得到

- 优点

足够 robust, 效果好, 符合相应的优化理论

- 缺点

时间开销大，需要整条路径构造完成后才能进行，且得到一条 offset path 需要多次迭代，同时当目标点距离初始点过远时，难以收敛，甚至无解

6.1、半向量保留和路径顶点重连

设路径长度为 k ，其中 S 链的长为 $k-3$

- $k-3$ 次 ray tracing
- 优点 容易实现，开销小，只需要路径的局部坐标类型（当前点和后继点）就可以构建 offset path
- 缺点 不能面对复杂的情况，需要连续 DD 的路径

七、GD- 算法分析

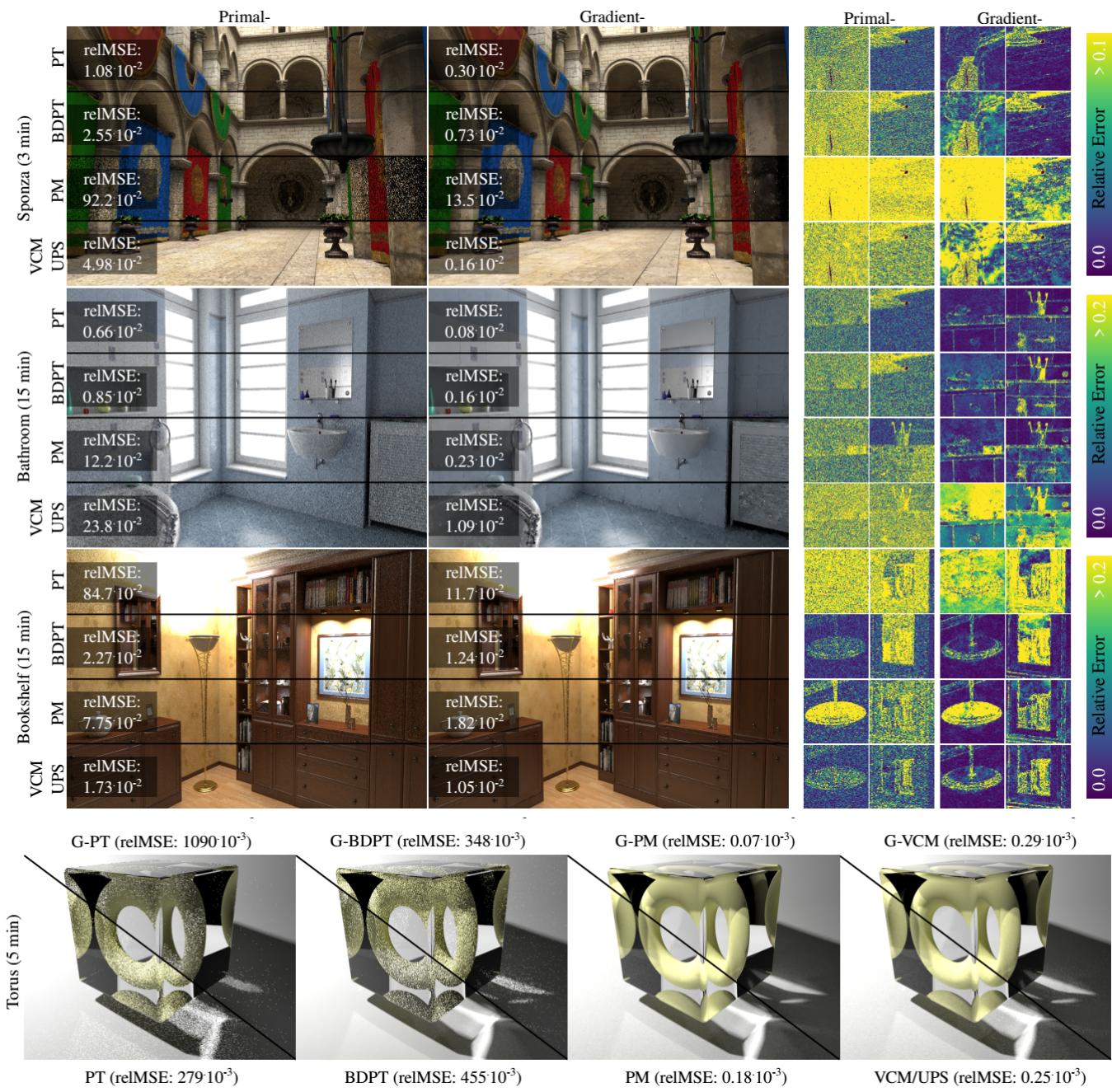
基于梯度域的渲染算法依然有其基础算法的缺点，例如：

- GD-PT 在焦散，光源遮挡上的问题
- GD-BDPT 在 SDS 上的问题
- GD-MLT 需要分布优秀的基础路径
- GD-PM Glossy 上效果较差
- GD-VCM 高频细节缺失和 Glossy - Spectrum 问题

GD 算法虽然引入了计算梯度的额外开销，但是大幅提高了收敛速度，即：

同采样数下，**GD** 算法更慢，但同时间下，**GD** 算法会有更好的效果

- 对于纯 Diffuse 传输的场景，例如 **Sponza**，梯度域方法能明显提高收敛速度。
- 对于包含复杂路径的场景，例如 **BATHROOM**, **BOOKSHELF**，梯度域方法依赖于：
 - 所使用的的基础渲染算法的效率
 - Shift 函数的开销
- 对于包含大量 \$SDS\$ 路径的场景，例如 **TORUS**，基于 **Density Estimation** 的方法明显优于基于 **MonteCarlo Path** 的算法



八、可研究点

1. GD 与其他基础渲染算法的结合 主要针对参与性介质的渲染方法，比如 GD-UPBP
2. 更好更高效的 Shift 函数
3. GD 与 Path Guide 的结合
4. GD 在工业界的应用 工业中，材质的属性过于复杂，不能简单的用 S 和 D 来区分
5. 其他领域 Temporal; 波动光学；自适应采样；梯度异常处理；高阶分析（黑塞矩阵）；重构方法