スペクトラルパストレーシングにおける不偏・一致性のある経路再サンプリング法

指導教員　シモセラ・エドガー　准教授

研究指導名 コンピューターグラフィックス研究

令和6年1月22日

提出者

早稲田大学 基幹理工学研究科 情報理工・情報通信専攻
学籍番号　5122F092-9
氏名　　　柳田 侑羽

概要

物理ベースレンダリングは1987年にKajiyaによって提案されて以降、現代に至るまで映画・ゲーム等の映像生産分野において写実的な表現を実現するために必須の技術となっている。近年、この物理ベースレンダリングにおいて光輸送問題を厳密に解く技術はリアルタイムレンダリングにも適用され、ゲーム分野においても写実的な表現が可能となった。この研究成果は従来のオフラインレンダリングにも応用可能であり、特にスペクトラルレンダリングのようなより物理的に忠実な表現が可能であるものの莫大な計算時間がかかる手法において、リアルタイムレンダリングの技術を応用することで、より高速なレンダリングが可能となる。その重要性にもかかわらずこの分野の研究はほとんど行われておらず、本研究ではこの分野における研究の必要性を示し、リアルタイムレンダリングの技術を応用することでオフラインスペクトラルレンダリングの高速化を実現する。我々は光輸送アルゴリズムの代表的な手法であるパストレーシングにおける再サンプリング手法をスペクトラルレンダリングに応用することを提案し、その定式化と実装を行った。これにより従来のスペクトラルパストレーシングと比較し、その品質に影響を与えることなく高速化を実現した。実用スペクトラルレンダリングで広く用いられている多波長サンプリングの技術や、リアルタイムレンダリングで今も発展し続けている再サンプリング手法をオフラインスペクトラルレンダリングに応用することでより高速なレンダリングを行うためには、今後もさらなる研究が必要である。

Abstract

Since its proposal by Kajiya in 1987, physics-based rendering has been an indispensable technology for achieving realistic image-generation in the field of image production, such as movies and games, up to the present day. In recent years, the technique of strictly solving the light transport problem in physics-based rendering has been applied to real-time rendering, enabling realistic image generation in the game field as well. The results of this research can also be applied to conventional off-line rendering, especially in methods such as spectral rendering, which can produce more physically realistic representations but require enormous computation time. Despite its importance, little research has been done in this area, and this research demonstrates the need for research in this area and the application of real-time rendering techniques to achieve faster off-line spectral rendering. We propose to apply the resampling technique to spectral rendering in path tracing, a typical method of light transport algorithms, and formulate and implement the technique. Further research is needed to apply the multi-wavelength sampling technique widely used in practical spectral rendering and the resampling method still being developed in real-time rendering to off-line spectral rendering for faster rendering. The following is a brief summary of the research results of the project.

目 次

図目次

表 目 次

謝辞

本研究を進めるにあたり、適切に指導してくださった指導教員のシモセラ・エドガー准教授、ならびに研究室関係者の方々に感謝いたします。

第1章

序論

1.1　研究目的

近年、GPU 進化によりリアルタイムレンダリングの世界においても、レイトレーシング等を用いた物理ベースレンダリングが実用の領域に達しつつあり、CD Project RED の Cyberpunk 2077 [11] [12] や Remedy Entertainment の Alan Wake2 [13] などにおいて実験的ながらリアルタイムパストレーシングを導入するなど、この分野は急速に発展している。この背景にあるのは、GPU のハードウェア性能の進化やデノイザの発展、DLSS のような Neural Net を活用したアップサンプリング技術だけでなく、2020 年に発表された ReSTIR アルゴリズム [6] による、レイトレーシングによるサンプルを再利用する技術の急速な発展によるところが大きい。この再サンプリング技術は 2022 年の GRIS によってフルパス空間への拡張が可能となり、Bias と引き換えに速度を追求するリアルタイムレンダリングの世界だけでなく、映画等の高精細な映像制作においてその品質を追求するオフラインレンダリングの世界にも応用が可能となった。
オフラインレンダリングの世界では、より高いリアリティを追求するためにレンダリングの照明計算を RGB ではなくスペクトルデータで行うスペクトラルレンダリングが行われること [14] がしばしばある。再サンプリング技術は今日でも発展を続けているが、スペクトル領域への応用に成功した研究は本論文の執筆次点では存在せず、進行中の先行研究が 1 例あるのみである。
本論文の目的はこのオフラインレンダリングにおけるスペクトラルレンダリングにおいて、再サンプリング技術を用いたレンダリングを行い、その品質を向上させることである。

1.2　論文構成

第 1 章で研究の目的、第 2 章で本研究を支える既存技術・概念を説明する。第 3 章では本研究と関連のある技術を紹介する。第 4 章で本研究の提案技術を説明し、第 5 でその評価を行う。第 6 では本研究の総括を述べる。

第2章

背景

2.1　光と色の定量的な取り扱い

2.1.1　光量の測定とその単位

光の物理的伝達を定量評価するための単位とその定義について論じる。

電磁放射の測定における基本単位は放射束 $\boldsymbol{\Phi}$ であり、その単位時間あたりの量は $\boldsymbol{W}$ で測定される。単位面積 $\boldsymbol{A}$ あたりの放射束、すなわち放射束の密度は放射照度 $E(d\boldsymbol{\Phi}/d\boldsymbol{A})$ で表す。放射の強さは、放射束の単位立体角あたりの密度として、放射強度 $I(d\boldsymbol{\Phi}/d\omega)$ で表す。ここで立体角の単位はステラジアン sr とする。面積と立体角双方の放射束の密度は、放射輝度 $L(d^2\boldsymbol{\Phi}/d\boldsymbol{A}d\omega)$ で表す。[15]

表 **2.1:** 放射測定量とその単位

名前	記号	単位
放射束	$\boldsymbol{\Phi}$	$\boldsymbol{W}$
放射照度	$\boldsymbol{E}$	$\boldsymbol{W}/m^2$
放射強度	$\boldsymbol{I}$	$\boldsymbol{W}/sr$
放射輝度	$\boldsymbol{L}$	$\boldsymbol{W}/m^2 sr$

2.1.2　RGB とスペクトル

測光

放射測定では、人間の知覚を考慮せずに純粋な物理量の測定を行う。測光は、人間の知覚を考慮した重み付けを波長毎に行う。
この重みは人間の目の、可視光領域の波長への応答を表す 555nm を中心とする

釣鐘曲線、CIE 標準比視感度 $V(\lambda)$ を用いて測光単位に変換する。この V はより厳密には CIE 明所視スペクトル比感度曲線と呼ばれ、この明所所とは、3.4*nit* より明るい場所を指し、網膜の錐体細胞が活動する。0.034*nit* では網膜の桿体細胞が活動し、507nm にピークをもつ CIE 暗所視スペクトル比感度曲線 V‘(λ) を用いる。

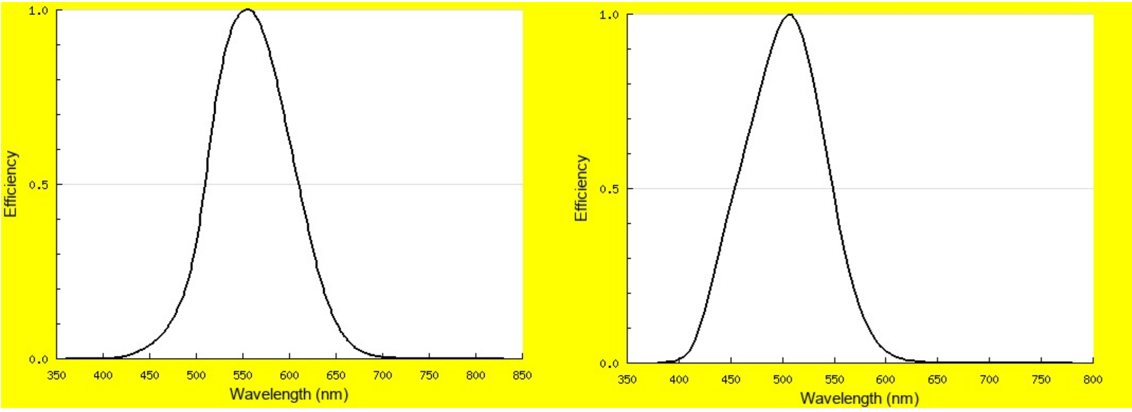

図 2.1: CIE 標準比視感度曲線。左が明所比視感度 V、右が暗所比視感度 V‘。CVRL [1] より引用。

放射測定と測光の違いは、この変換曲線の存在と測定単位の違いである。例として放射測定では、放射輝度は W/m^2sr で測定されるが、測光では cd/m^2 で測定される。この単位の関係を以下の表に記載する。

表 2.2: 放射測定量と測光量の単位

放射測定量: 名前	放射測定量: 単位	測光量: 名前	測光量: 単位
放射束	$\boldsymbol{W}$	光束	$\boldsymbol{lm}$
放射照度	$\boldsymbol{W}/m^2$	照度	$\boldsymbol{lx} = \boldsymbol{lm}/m^2$
放射強度	$\boldsymbol{W}/sr$	光度	$\boldsymbol{cd} = \boldsymbol{lm}/sr$
放射輝度	$\boldsymbol{W}/m^2sr$	輝度	$\boldsymbol{nit} = \boldsymbol{cd}/m^2$

測色

　様々な波長にわたる光のエネルギー分布を表したものをスペクトルパワー分布(SPD)と呼ぶ。このSPDと人間の色の知覚の関係を扱うのが測色である。
人間の網膜は、電磁波のうち波長およそ380*nm*〜780*nm*の可視光領域のみに反応する。ここで網膜の錐体細胞は3つの色受容体を持ち、この連続した波長の光のSPDを3つの信号に変換する。この性質により、例えば図のように典型的な野外照明を表すSPDと、レーザー光源によるSPDは、人間の目には同じ色に見える。
CIE(国際照明委員会)により、このSPDを3つの信号に変換する関数の特定のために、RGB3つの単色光源の重み付け混合光源と任意波長の比較光源Fが人間の近くにおいて同等となるような重みを求めるカラーマッチング実験が行われた。
この実験により、RGB等色関数r, g, bが定義されたが。この等色関数は負の値を持つ。これは、RGB3つの単色光源のみでは等色テストを通過できず、任意波長光源F側に単色光源を追加する必要があったことを意味する。
負の値を持つ等色関数は直感的でなく、また計算の簡単のためや後述のメリットのために、すべての可視光領域の波長において正の値を取る3つの仮想光源XYZが定義された。この光源のSPDは波長によっては負の値をもつ、現実には存在しない光源である。
この仮想光源における等色関数は図2.2のようになる。
ここで、任意のSPD$s(\lambda)$と等色関数$\bar{x}, \bar{y}, \bar{z}$の内積をとることで、そのSPDの

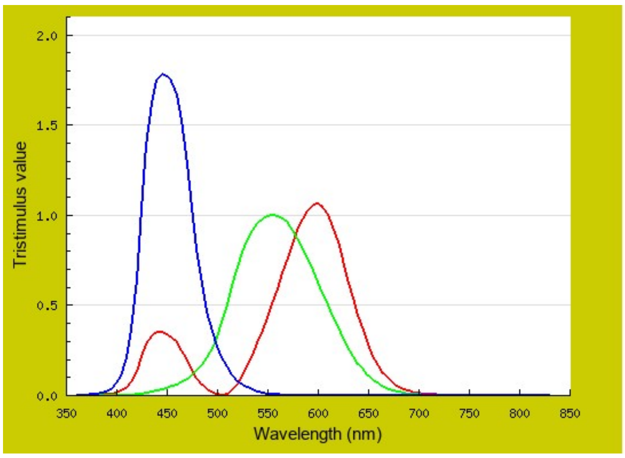

図 **2.2:** XYZ等色関数。CVRL [1] より引用。

XYZ値が求められる。

$$
\begin{aligned}
X &= \int_{380nm}^{780nm} s(\lambda)\bar{x}(\lambda)d\lambda \\
Y &= \int_{380nm}^{780nm} s(\lambda)\bar{y}(\lambda)d\lambda \\
Z &= \int_{380nm}^{780nm} s(\lambda)\bar{z}(\lambda)d\lambda
\end{aligned}
\tag{2.1.1}
$$

この三つの3刺激値 X, Y, Z はCIE XYZ空間における色の重みを意味する。色を輝度と色度に分けることで、色の知覚を定量的に扱うことができる。
この空間は3刺激値を $X + Y + Z = 1$ に投影したもので、次で計算される。

$$
\begin{aligned}
x &= \frac{X}{X+Y+Z} \\
y &= \frac{Y}{X+Y+Z} \\
z &= \frac{Z}{X+Y+Z} = 1 - x - y
\end{aligned}
\tag{2.1.2}
$$

ここで、色度座標 x と y のプロットはCIE 1931色度図と呼ばれ、図2.3のようになる。
この図において、色度座標 (x, y) は、波長 λ の光の色を表し、また点x,y=(0.31271,0.32902)をCIE標準光源D65における白点と呼ぶ。
ここでD65光源は平均的な昼光を表す光源であり、図2.4のようなSPDを持ち、その相関色温度はおよそ6500Kであることが知られている。
XYZ空間からRGB空間への変換は、CIE XYZ空間の3つの基底をRGB空間の3つの基底に変換することで行われる。色空間は、しばしば色度図上の三角形によって表され、その基底を決めることができる。
つまり、XYZ空間とRGB空間の関係は3x3の変換行列によって与えることができる。

$$
\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}
\tag{2.1.3}
$$

この変換行列 M は色空間ごとに決められた白点と3つの原色の XYZ 空間における座標によって決まる。
以下に代表的な色空間とその変換行列を示す。

IECが1998年に定めたsRGB色空間は標準的な多くのディスプレイがサポートし、現在でも広く使われる。AdobeRGBはAdobe社が1998年に定めた色空間であり、sRGBよりも広い色域を持つ。

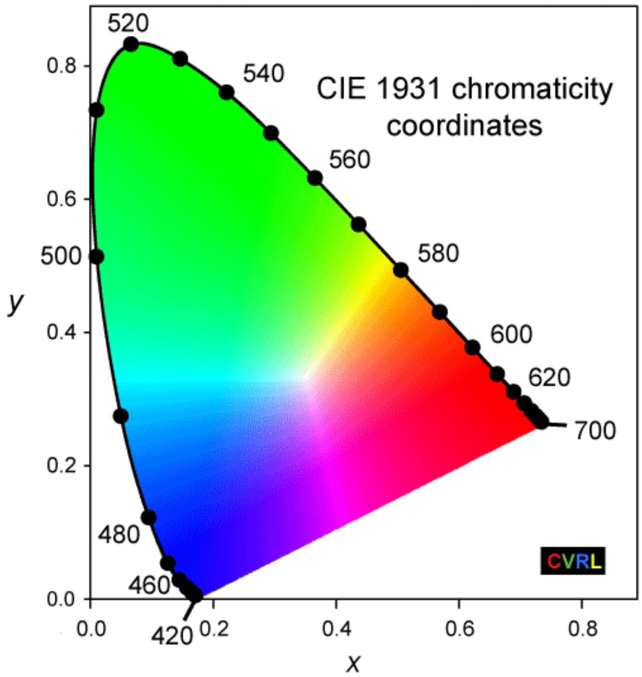

図 **2.3:** CIE 1931 色度図。CVRL [1] より引用。

図 **2.4:** D65 光源の SPD

DCI-P3 はデジタルシネマイニシアティブが定めた色空間であり、映画館のデジタルプロジェクターで使われる。Rec.2020 は ITU が定めた色空間であり、HDR 映像の標準として使われる。

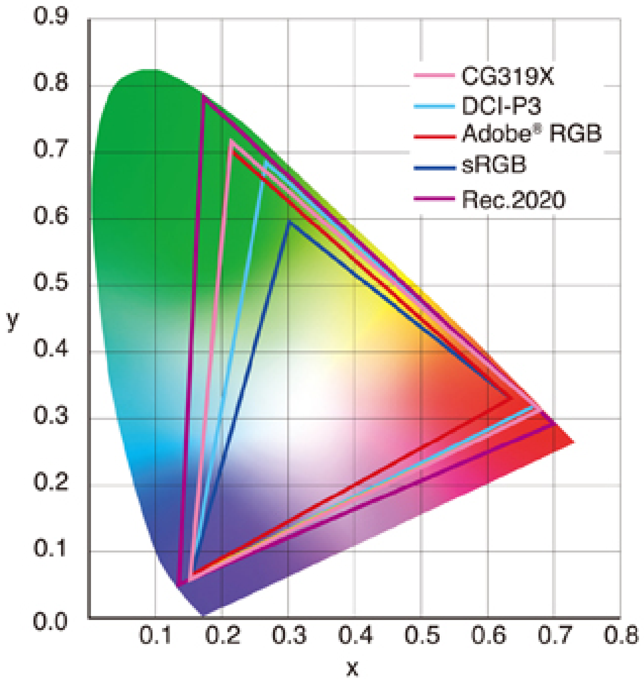

図 **2.5:** 様々な RGB 色空間。CVRL [1] より引用。

表 **2.3:** 色空間とその変換行列

色空間	白点	RGB to XYZ M	XYZ to RGB M^{-1}
sRGB	D65	$\begin{bmatrix} 0.412391 & 0.357584 & 0.180481 \\ 0.212639 & 0.715169 & 0.072192 \\ 0.019331 & 0.119195 & 0.950532 \end{bmatrix}$	$\begin{bmatrix} 3.240970 & -1.537383 & -0.498611 \\ -0.969244 & 1.875968 & 0.041555 \\ 0.055630 & -0.203977 & 1.056972 \end{bmatrix}$
Adobe RGB	D65	$\begin{bmatrix} 0.576669 & 0.185558 & 0.188229 \\ 0.576669 & 0.185558 & 0.188229 \\ 0.027031 & 0.070689 & 0.991338 \end{bmatrix}$	$\begin{bmatrix} 2.041588 & -0.565007 & -0.344731 \\ -0.969244 & 1.875968 & 0.041555 \\ 0.013444 & -0.118362 & 1.015175 \end{bmatrix}$
DCI-P3	D65	$\begin{bmatrix} 0.486571 & 0.265668 & 0.198217 \\ 0.228975 & 0.691739 & 0.079287 \\ 0.000000 & 0.045113 & 1.043944 \end{bmatrix}$	$\begin{bmatrix} 2.493497 & -0.931384 & -0.402711 \\ -0.829489 & 1.762664 & 0.023625 \\ 0.035846 & -0.076172 & 0.956885 \end{bmatrix}$
Rec.2020	D65	$\begin{bmatrix} 0.636958 & 0.144617 & 0.168881 \\ 0.262700 & 0.677998 & 0.677998 \\ 0.000000 & 0.0280727 & 1.060985 \end{bmatrix}$	$\begin{bmatrix} 1.716651 & -0.355671 & -0.253366 \\ -0.666684 & 1.616481 & 0.015769 \\ 0.017640 & -0.042771 & 0.942103 \end{bmatrix}$

RGB によるレンダリング

コンピューターグラフィックスにおいて、生成される画像の各ピクセルの値は 3D 仮想空間上にある光源 E から発射された光が空間内の物体表面 $\mathcal{A}$ と作用

し、仮想的なセンサーSに入力された放射輝度の値によって決まる。
このとき、光源Eがもつ光の量、物体表面を定義する属性、センサーの感度を$RGB:\mathbb{R}^3$空間で考えたレンダリング、すなわち光源からのRGBで表される放射輝度を加算や乗算その他演算で構成される推定機によって計算することを考える。

$$\langle I \rangle = estimate(\mathcal{A}|RGB, E|RGB, S|RGB) \tag{2.1.4}$$

このようなレンダリング方式はRGBレンダリングである。
現在、ほとんどのゲームアプリケーションや多くの映画において、RGBレンダリングが用いられている。

スペクトラルレンダリング

RGBレンダリングアルゴリズムは乗算のような操作を頻繁に行う。この操作は色空間によって結果が異なることが知られており、RGBレンダリングの問題点である。
例えばsRGB空間において、ある輝度の値$[0,0,1]$が、あるアルベド値$[0,0,1]$をもつ物体表面で反射したとき、成分ごとの乗算は$[0,0,1]\times[0,0,1]=[0,0,1]$となる。
ここで、色空間をより広いRec.2020に切り替えたとき、$[0,0,1]^{sRGB}\neq[0,0,1]^{Rec.2020}$である。
このとき、$[0,0,1]^{sRGB}\times[0,0,1]^{sRGB}=[0,0,1]^{sRGB}$であるが、$[0,0,a]^{Rec.2020}\times$

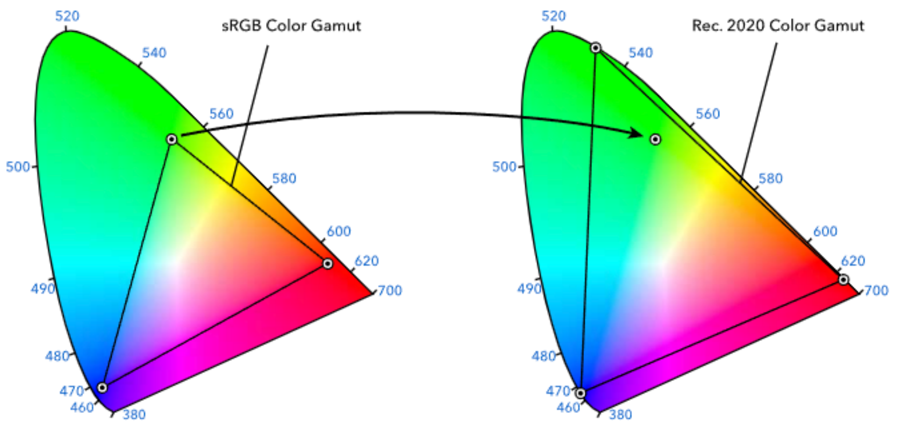

図 **2.6:** sRGB色空間における$[0,0,1]$のRec.2020色空間での対応。[2]より引用。

$[0, 0, a]^{Rec.2020} = [0, 0, a^2]^{Rec.2020}, a < 1$ となり、Rec.2020 では $1 - a^2$ ぶんの光が物体表面で吸収されることを意味するがこの結果は明らかに不適切である。
この問題は、色の演算は可視光のスペクトル領域 Λ で計算し、センサーに反応した後に RGB 空間に変換することで解決できる。
すなわち、

$$\langle I \rangle = SpectrumToRGB(estimate(\mathcal{A}|\Lambda, E|\Lambda, S|\Lambda)) \tag{2.1.5}$$

とすればよい。

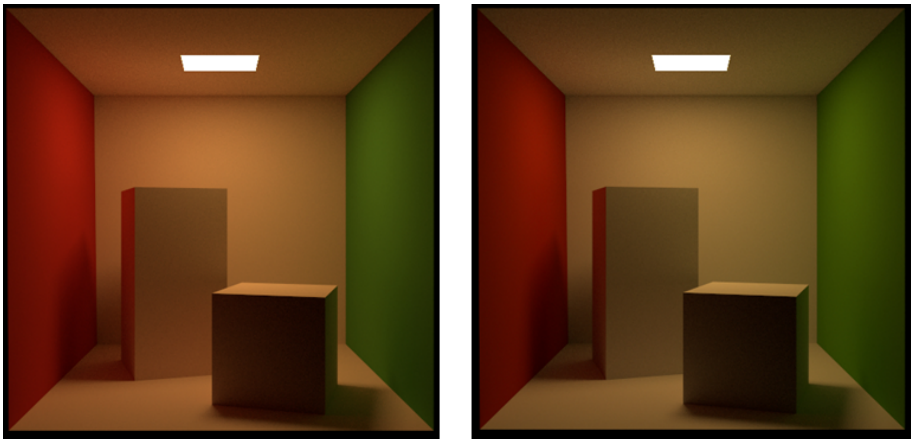

図 2.7: RGB によるレンダリング (左) とスペクトラルレンダリング (右)。前述の問題により、RGB によるレンダリングでは彩度が不正な値になっていることが確認できる。図は [2] より引用。

また、分光現象など何らかの波長依存の光学現象を考慮したい場合、RGB レンダリングではなくスペクトルレンダリングを行う必要があることが多い。

図 2.8: 波長依存の光学現象の例。ホイヘンスの原理による屈折物体における分光現象の他、可視波長より微細な構造をもった物体表面でも波長依存の現象は発生する。図は [3] より引用。

2.2　モンテカルロ積分

2.2.1　確率分布の台、関数の台

ある確率変数 X とその確率密度関数 p を考える。このとき、X の台 (Random variable support) は以下のように定義される。

定義 2.1 確率変数の台 *(Random variable support)*

$$supp(X) = \{x \in \mathbb{R} \quad | \quad p(x) > 0\}$$

すなわち、X の台とは、その X が 0 でない正の値を取りうる範囲のことである。また、関数 f の台 (Function support) は以下のように定義される。

定義 2.2 関数の台 *(Function support)*

$$supp(f) = \{x | f(x) \neq 0\}$$

すなわち、f の台とは、その f が 0 でない値を取りうる範囲のことである。

2.2.2　モンテカルロ法

ある関数 $f : \mathbb{R} \to \mathbb{R}$ を区間 $[a, b]$ で積分することを考える。このとき、その積分地は以下のように表される。

$$I = \int_a^b f(x) dx \tag{2.2.1}$$

また、ある連続した確率分布 p について、関数 f の期待値は以下のように表される。

$$E[f(x)] = \int_{\mathbb{R}} f(x) p(x) dx \tag{2.2.2}$$

ここで、$[a, b]$ で一様分布するそれぞれ独立な N 次元の確率変数 $\boldsymbol{x} = [x_1, x_2, x_3, ..., x_N]$ について、式 2.2.1, 式 2.2.2 から I の推定値 $\langle I \rangle$ を求めることができる。

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \tag{2.2.3}$$

このように、確率変数 $\boldsymbol{x}$ を用いて I を推定する手法をモンテカルロ法と呼ぶ。
モンテカルロ法の推定値 $\langle I \rangle$ は、その期待値が真値に一致する。

$$E[\langle I \rangle] = I \tag{2.2.4}$$

また、$N \to \infty$ のとき、$\langle I \rangle$ は真値 I に収束する。

モンテカルロ法は、高次元の関数 $f : \mathbb{R}^n \to \mathbb{R}^n$ にも拡張できる。

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i^1, x_i^2, x_i^3, ..., x_i^n)}{p(x_i^1, x_i^2, x_i^3, ..., x_i^n)} \tag{2.2.5}$$

2.2.3　不偏・一致推定量

ある関数 f をドメイン Ω で積分することを考え、その積分値を I とする。

$$I = \int_a^b f(x)dx \tag{2.2.6}$$

このとき、f の積分値を推定するための関数 $\hat{I}$ を考える。このとき、$\hat{I}$ が I の不偏推定量であるとは、以下の式が成り立つことである。

$$E[\hat{I}] = I \tag{2.2.7}$$

推定関数 $\hat{I}$ の期待値と真値の間にずれ $\epsilon = E[\hat{I}] - I$ があるとき、それを推定関数の偏り (Bias) と呼ぶ。
また、$\hat{I}$ が I の一致推定量であるとは、以下の式が成り立つことである。

$$\lim_{N \to \infty} \hat{I} = I \tag{2.2.8}$$

2.3　物理ベースのレンダリング

物理ベースレンダリングで解くべき問題を簡潔に表現すると、画像のピクセルに対応するカメラビューの方向に沿って、仮想的なセンサーに入射する放射輝度を計算することである。この章では物理ベースレンダリングの定式化とその評価手法について簡潔に述べる。

2.3.1　物体材質とその反射の定式化

カメラに入る放射輝度は、カメラ位置を $\boldsymbol{c}$ とし、カメラビュー レイの方向を $-\boldsymbol{v}$ とすると、$L_i(\boldsymbol{c}, -\boldsymbol{v})$ と書ける。

また、オブジェクト間を満たす媒質が、吸収・散乱を通じて放射輝度に大きな影響を与える場合も考慮しない。これは、ほとんどのレンダリングではこの媒質は比較的綺麗な空気であり、その影響を無視できるからである。この仮定のもとでは、カメラビューレイと最初に交差するオブジェクトの交差点 $\boldsymbol{p}$,$\boldsymbol{l}$ の,$-\boldsymbol{l}$ 放射輝度と等しくなる。

$$L_i(\boldsymbol{c}, -\boldsymbol{v}) = L_o(\boldsymbol{p}, \boldsymbol{v})$$

すなわち、ピクセルの色を決定するためには $L_o(\boldsymbol{p}, \boldsymbol{v})$ を計算しなければならない。また、このカメラビューレイをプライマリレイと呼ぶ。

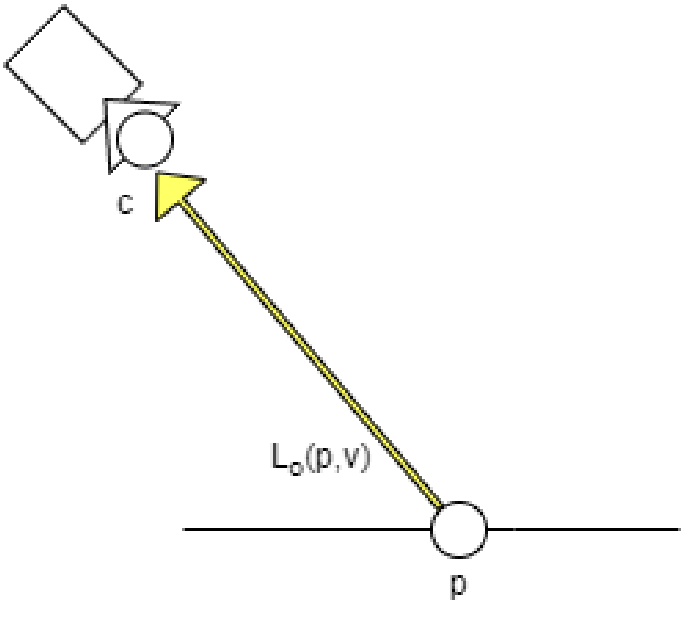

図 **2.9:** カメラビューと最初の衝突点

2.3.2　レンダリング方程式

レンダリング方程式 [16] はシーン全体の大域照明についてより一般的な記述である。

つまり、ある物体の表面上の点 $\boldsymbol{p}$ において、カメラへむかうレイを $\boldsymbol{v}$ とした

ときの放射輝度 L_o を求める上で、点 $\boldsymbol{p}$ から v に向けて放射される放射輝度を $L_e(\boldsymbol{p}, \boldsymbol{v})$ とした時の式は

$$L_o(p, v) = L_e(p, v) + \int_{l \in \Omega} f_r(\boldsymbol{l}, v) L_i(p, \boldsymbol{l})(n, \boldsymbol{l})^+ d\boldsymbol{l} \qquad (2.3.1)$$

となる。
ここで、$\int_{l \in \Omega}$ はある表面上の点において、そこへ向かう全てのベクトル $\boldsymbol{l}$ すなわちその点を中心として覆う半球 Ω を通過して点に向かうベクトルの積分を表す。
また、f は BRDF、$(\boldsymbol{a}, \boldsymbol{b})^+$ は $max(0, \boldsymbol{a} \cdot \boldsymbol{b})$ を表す。この式の視覚的な説明を図 2.10 に示す。

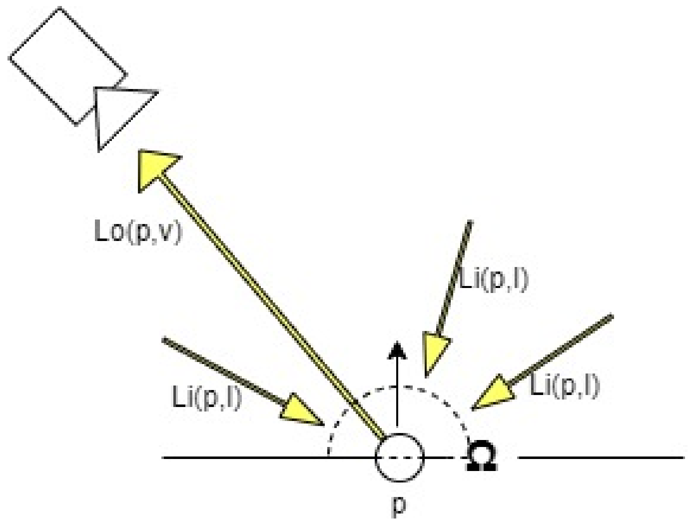

図 **2.10:** レンダリング方程式の概略図

レンダリング方程式の積分項は点 $\boldsymbol{p}$ で反射される放射輝度を表しており、$L_i(p, l) = L_o(r(\boldsymbol{p}, \boldsymbol{l}), -\boldsymbol{l})$ である。$r(\boldsymbol{a}, \{b)$ は点 $\boldsymbol{a}$ から方向 $\boldsymbol{b}$ 向きに最も近いオブジェクトの点の位置を返すレイキャスト関数であり、$r(\boldsymbol{p}, \boldsymbol{l})$ は再帰的である。つまり、この項を評価するためには図 2.11 レイキャストを光源、すなわち自己発光する物体にたどり着くまで繰り返す必要がある。
この式は、現実世界において光が物体を照らし、また衝突した物体ごとに跳ね返ったり吸収・屈折する直感的な解釈において、光線がとる可能な全ての経路をまとめて表している。
コンピューターグラフィックスにおいて、この光がとる経路を記述する表記スキームに Heckbert の記述法 [17] がある。

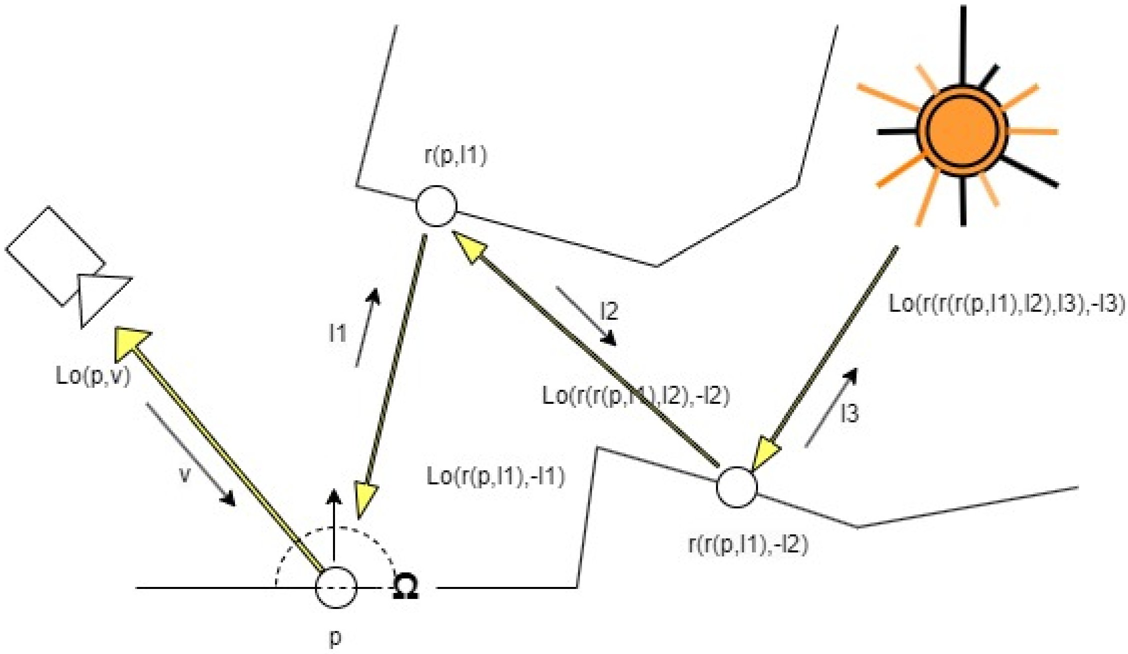

図 **2.11:** 再帰的なレイキャスト

heckbert の記述法では、光源は *L*、目やセンサーは *E*、拡散表面に *D*、鏡面に *S* のようにラベル付けして使用する。表2.4 に、その要約を載せる。図2.12 にその例を示す。この、光源からカメラに到達する光の経路を、光輸送経路と呼ぶ。

表 **2.4:** 光輸送経路の正規表現表記

演算子	記述	例	説明
*	0 以上	S*	0 回以上の Specular 反射
+	1 以上	D+	1 回以上の Diffuse 反射
?	0 または 1	S?	0 回か 1 回の Specular 反射
|	どちらか	D|SS	一回の Diffuse 反射もしくは、2 回の Specular 反射
()	グループ	(D|S)*	0 回以上の Diffuse 反射もしくは Specular 反射

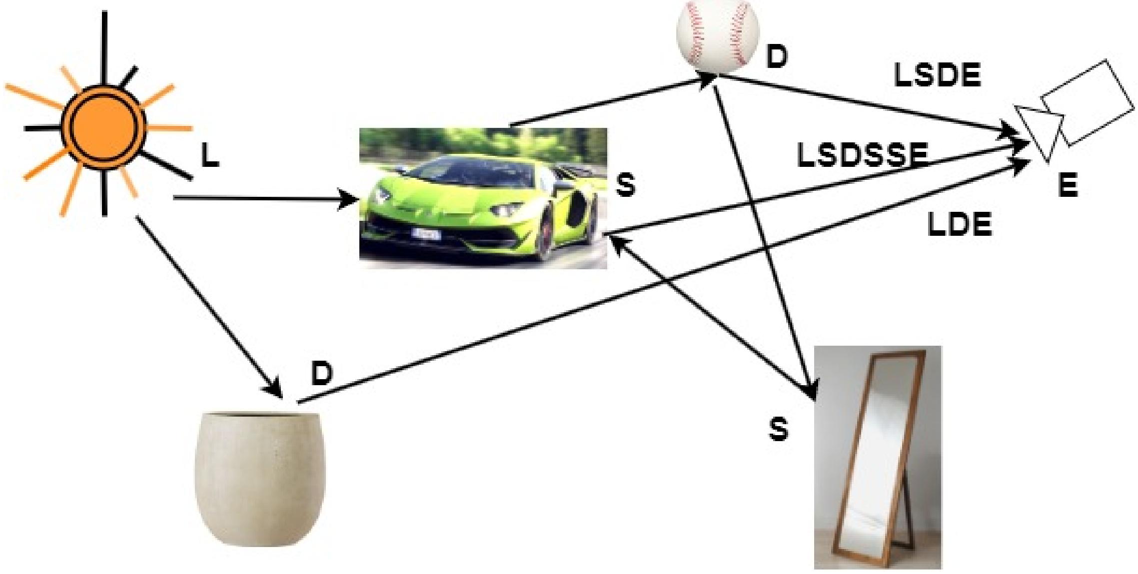

図 **2.12:** 光源からカメラに到達する経路とその表記の例

2.3.3 経路積分を考慮したレンダリング方程式

3次元空間における微小面積と微小立体角を変換することを考える。
ここで、ある点xとそれに関する微小立体角$d\omega$、点x'とその法線ベクトル$\vec{n'}$を考える。
このとき、xからx'に向かう微小立体角$d\omega$は、点x'の微小面積がx上の微小立体角を占める面積は距離の二乗に反比例することを考慮して以下のように表される。

$$d\omega = \frac{|(\vec{x'} - \vec{x}) \cdot \vec{n'}|}{\|\vec{x'} - \vec{x}\|^2} V(x' \leftrightarrow x) dA \tag{2.3.2}$$

ここで、$V(x' \leftrightarrow x)$は可視性項であり、xとx'が互いに可視であるとき1、そうでないとき0となる。
このとき、レンダリング方程式2.3.1は以下のように書き換えられる。

$$\begin{aligned} L_o(\boldsymbol{x_1} \to \boldsymbol{x_0}) = L_e(\boldsymbol{x_1} \to \boldsymbol{x_0}) + \int_{\mathcal{A}} (\prod_{i=1}^{D-1} f(\boldsymbol{x_{i+1}} \to \boldsymbol{x_i} \to \boldsymbol{x_{i-1}}) G(x_{i+1} \leftrightarrow x_i) \\ V(x_{i+1} \leftrightarrow x_i)) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}) d\boldsymbol{x_2} ... d\boldsymbol{x_D} \end{aligned} \tag{2.3.3}$$

ここで、シーン中の全ての物体表面の集合を$\mathcal{A}$とする。

2.3.4 光輸送経路の構築と評価

レンダリング方程式 2.3.1 を評価し、ピクセルごとの色を決定するための放射輝度を計算するアルゴリズムを、光輸送アルゴリズムと呼ぶ。本節では光輸送アルゴリズムについて、特に本研究で使用する手法を中心に解説する。

パストレーシング法

現在でも研究から実応用分野で広く使われいる光輸送アルゴリズムに、Kajiya の提案したパストレーシング法 [16] がある。
レンダリング方程式は再帰的なレイキャストを持ち、経路は任意の回数のバウンスを取りうる。パストレーシングは、この可能な光輸送経路をカメラから光源に向かって構築し、モンテカルロ積分によって経路の積分を評価する光輸送アルゴリズムである。
パストレーシングが構築する光輸送経路 (以下、簡単のために光輸送経路をパスと呼ぶ) と、その経路積分について考えるために式 2.3.1 をパス空間について拡張する。
ここで、パスとはシーン中の全ての物体表面の集合を $\mathcal{A}$ としたとき、任意の物体表面 $x_i \in \mathcal{A}$ からなる列であり、例えばバウンス回数 $\mathcal{D}-1$ 回のパスは以下のようにあらわせる。

$$[\boldsymbol{x_0}, \boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_{\mathcal{D}}}] \tag{2.3.4}$$

また、パス空間とは、バウンス回数 $\mathcal{D}-1$ 回のパスが持ちうるサーフェスの集合を $\mathcal{A}^{D-1}$ としたとき、$\cup_{\mathcal{D}=2}^{\infty}\mathcal{A}^{D-1}$ で表される積空間の和である。
このとき、式 2.3.1 は微小面積に関する積分を考えることで以下のように拡張できる。

$$\begin{aligned} L_o(\boldsymbol{x_1} \to \boldsymbol{x_0}) = L_e(\boldsymbol{x_1} \to \boldsymbol{x_0}) + \sum_{D=2}^{\infty} \int_{\mathcal{A}^{D=1}} \Big(\prod_{i=1}^{D-1} f(\boldsymbol{x_{i+1}} \to \boldsymbol{x_i} \to \boldsymbol{x_{i-1}}) G(x_{i+1} \leftrightarrow x_i) \\ V(x_{i+1} \leftrightarrow x_i) \Big) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}) d\boldsymbol{x_2} ... d\boldsymbol{x_D} \end{aligned} \tag{2.3.5}$$

ここで、G は $G(x_\alpha \leftrightarrow x_\beta) = |\vec{\omega}_{\alpha\beta} \cdot \vec{n}_\alpha||\vec{\omega}_{\beta\alpha} \cdot \vec{n}_\beta| / ||\boldsymbol{x_\alpha} - \boldsymbol{x_\beta}||$ で表される幾何項であり、V は可視性項である。
式 2.3.5 は、パス空間において、点 $\boldsymbol{x_1}$ から点 $\boldsymbol{x_0}$ に向かう放射輝度を、あるバウンス回数の可能な全てのパスの積分について、全てのバウンス回数の総和を意味する。この式をコンピューターが計算可能な形式に置き換えることを考える。

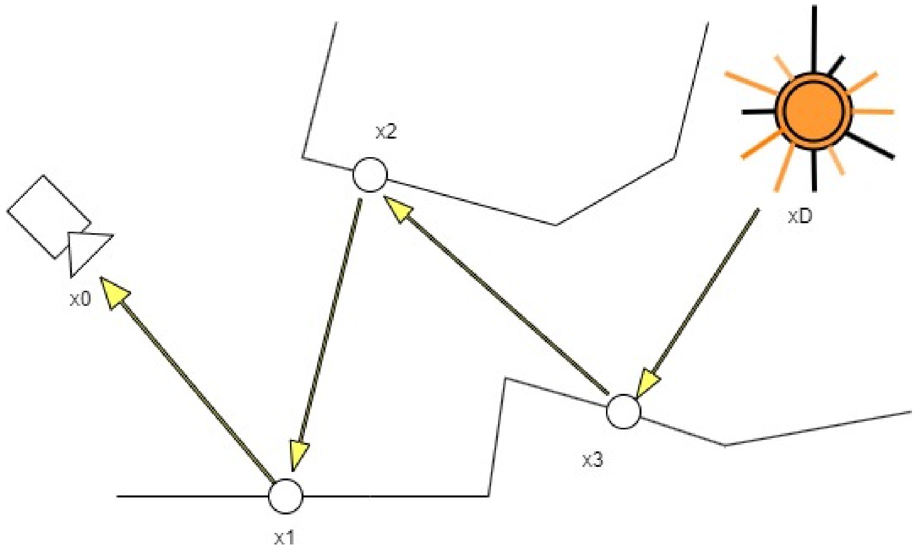

図 **2.13:** 一本のパス

まず、パスの積分をモンテカルロ積分によって近似すると、以下の式 2.3.6 が得られる。

$$\langle L_o(\boldsymbol{x_1} \to \boldsymbol{x_0})\rangle = L_e(\boldsymbol{x_1} \to \boldsymbol{x_0}) + \sum_{D=2}^{\infty} \frac{1}{N} \sum_{i=1}^{N} Thp(\boldsymbol{x_i}) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}) \tag{2.3.6}$$

Thp は以下の形式で表されるスループット関数であり、$p(\alpha \to \beta)$ は点 $\boldsymbol{\alpha}$ から点 $\boldsymbol{\beta}$ をサンプリングする確率であり、モンテカルロ被積分関数のサポートのとるサーフェスの集合において、その確率は必ず 0 より大きい。

$$Thp(\boldsymbol{x}) = \prod_{j=1}^{D-1} \frac{f(\boldsymbol{x_{j+1}} \to \boldsymbol{x_j} \to \boldsymbol{x_{j-1}}) G(x_{j+1} \leftrightarrow x_j) V(x_{j+1} \leftrightarrow x_j)}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}})} \tag{2.3.7}$$

$$p(x_i) \neq 0, \forall x_i \in supp(Thp(\boldsymbol{x_i}) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}))$$

次に無限和をロシアンルーレットによってその期待値を真値に一致させながら打ち切る。
このとき、スループット関数 Thp にロシアンルーレットの確率 P_j を導入し、以下のように書き換える。

$$Thp(\boldsymbol{x}) = \prod_{j=1}^{D-1} \frac{f(\boldsymbol{x_{j+1}} \to \boldsymbol{x_j} \to \boldsymbol{x_{j-1}}) G(x_{j+1} \leftrightarrow x_j) V(x_{j+1} \leftrightarrow x_j)}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}}) P_j} \tag{2.3.8}$$

$$
\begin{aligned}
\langle L_o(\boldsymbol{x_1} \to \boldsymbol{x_0})\rangle &= L_e(\boldsymbol{x_1} \to \boldsymbol{x_0}) \\
&\quad + \sum_{D=2}^{\infty} \begin{cases} \frac{1}{N}\sum_{i=1}^{N} Thp(\boldsymbol{x_i}) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}) & \text{Probability of } P_j \text{ (Not killed yet)} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{2.3.9}
$$

ここで、$L_e(\boldsymbol{x_1} \to \boldsymbol{x_0})$ は総和に無関係であり、また式 2.3.9 中の総和は入れ替え可能なため、以下のように式を変形できる。

$$
\begin{aligned}
\langle L_o(\boldsymbol{x_1} \to \boldsymbol{x_0})\rangle &= \frac{1}{N}\sum_{i=1}^{N}\Big(L_e(\boldsymbol{x_1} \to \boldsymbol{x_0}) \\
&\quad + \sum_{D=2}^{\infty} \begin{cases} Thp(\boldsymbol{x_i}) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}) & \text{Probability of } P_j \text{ (Not killed yet)} \\ 0 & \text{otherwise} \end{cases}\Big)
\end{aligned}
\tag{2.3.10}
$$

　式 2.3.5 から式 2.3.10 の変形を、同様に微小面積に関する積分の形式について考えると以下の式 2.3.11 が得られる。

$$
\begin{aligned}
\langle L_o(\boldsymbol{x}, \vec{\omega_o})\rangle &= \frac{1}{N}\sum_{i=1}^{N}\Big(L_e(\boldsymbol{x}, \vec{\omega_o}) \\
&\quad + \sum_{D=2}^{\infty} \begin{cases} \prod_{j=1}^{D-1} \frac{f(\boldsymbol{x_j}, \vec{\omega_{ji}}, \vec{\omega_{jo}})|\omega_{ji} \cdot \vec{n_j}|}{p(\boldsymbol{x_j}, \omega_{ji}) P_j} L_e(\boldsymbol{x_{D-1}}, \vec{(}\omega_{D-1,D})) & \text{Probability of } P_j \text{ (Not killed yet)} \\ 0 & \text{otherwise} \end{cases}\Big)
\end{aligned}
\tag{2.3.11}
$$

この式 2.3.11 をコンピューター上で計算する疑似コードは以下 1 のように書け、これをパストレーシング法と呼ぶ。

　パストレーシング法の視覚的な説明を図 2.14 に示す。
図のように、ロシアンルーレットもしくは無限遠方へのレイトレースによってパスが終了するまで、パスの深さを一つずつ増やしながら、最終的に D 個のパスを構築し、その評価をする。

　また、式 2.3.10 およびコード 1 におけるモンテカルロ積分の総和上限 N は、一般にサンプル数と呼ばれ、spp などと書かれる。
モンテカルロ積分の性質により、推定値 $\langle L_o\rangle$ の標準偏差は、サンプル数の平方

```
struct {
    float3 origin,direction,thp
} Ray
struct {
    bool hit
    float3 wo
    Vertex x
    float3 n
    Bsdf bsdf
} HitResult
begin
    I ⟵ 0.0
    for i = 0toN do
        ray ⟵ getViewRay(p)
        ray.thp ⟵ 1.0
        result ⟵ traceRay(ray)
        while result.hit = True do
            if result.x.hasEmission then
                I ⟵ I + ray.thp ∗ Le(result.x, result.wo)
            end
            bsdfSample ⟵ bsdf.sample(result)
            ray.thp ⟵ (thp ∗ bsdf.eval(result.x, result.wo, bsdfSample.wi) ∗
              dot(result.n, bsdfSample.wi))/bsdfSample.pdf
            Prr ⟵ russianRoulette()
            if random() ≥ Prr then
                break
            end
            ray.thp ⟵ ray.thp/Prr
            ray.origin ⟵ result.x
            ray.direction ⟵ bsdfSample.wi
            result ⟵ traceRay(ray)
        end
    end
    return I/N
end
```

Algorithm 1: パストレーシング法の疑似コード

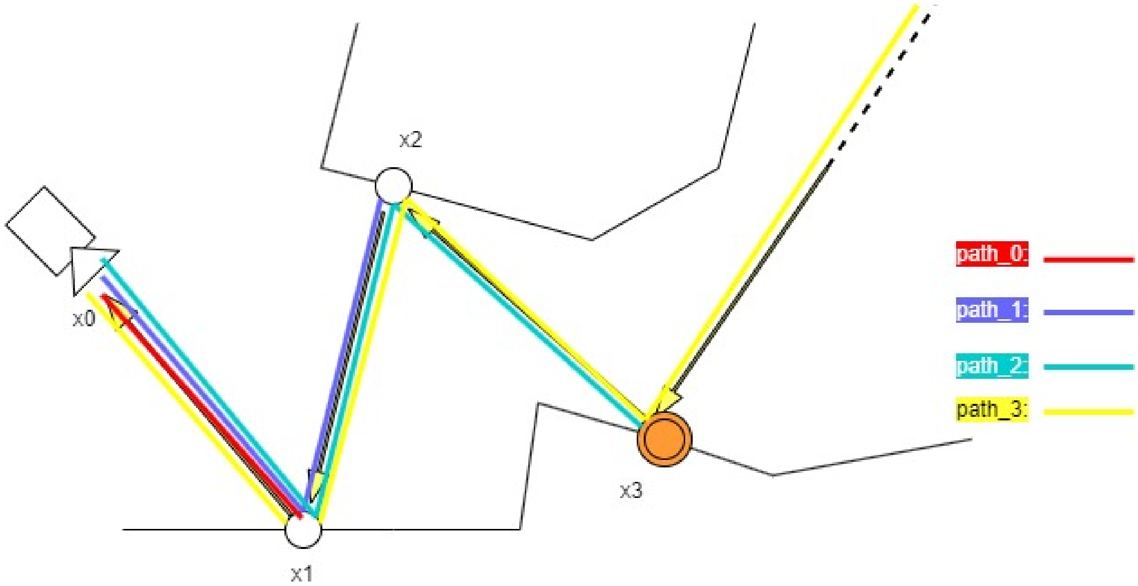

図 **2.14:** パストレーシング法の概略図

根に反比例し、これはパストレーシング法によって計算される画像のノイズ、すなわち真値との誤差を $\frac{1}{N}$ 倍にしたいとき、サンプル数を N^2 倍にしなければならないことを意味する。サンプル数の総和は全ての項の外にいるため、一般に計算の処理時間も N^2 倍となる。
このように、パストレーシング法はサンプル数の増加に伴い、計算時間が二次的に増加するためサンプル数を単純に増やすことが実用的に難しい場合がある。パストレーシング法の分散低減手法はコンピューターグラフィックスにおける重要な課題であり今日に至るまで多数研究されている。

Next Event Estimation による分散低減

式 2.3.10 における Thp 項と L_e 項の積の評価について考える。
この積が 0 になるとき、それは放射輝度のモンテカルロ推定値 $\langle L_e \rangle$ に寄与しないため無駄である。コード 1 では 25 行目にて、衝突点が発光していた場合、すなわち L_e 項が非ゼロ値の場合にのみ計算をしているが、一般の 3D シーンにおいては空のような IBL を除き、発光する物体のサーフェスは少ないためできるだけ発光しているサーフェスを評価することで分散の低減につなげることができる。
そのための代表的で有用な手法として Next Event Estimation(NEE) がある。
コード 1 では新しいレイの出射方向 ω_i について、BRDF のサンプリングを行っている。ここで式 2.3.10 の微小面積形式のレンダリング方程式に注目すると、

これを発光するサーフェスのサンプリング、すなわち$\boldsymbol{x_D}$が必ず発光しているようなサンプリングに置き換えることができるとわかる。

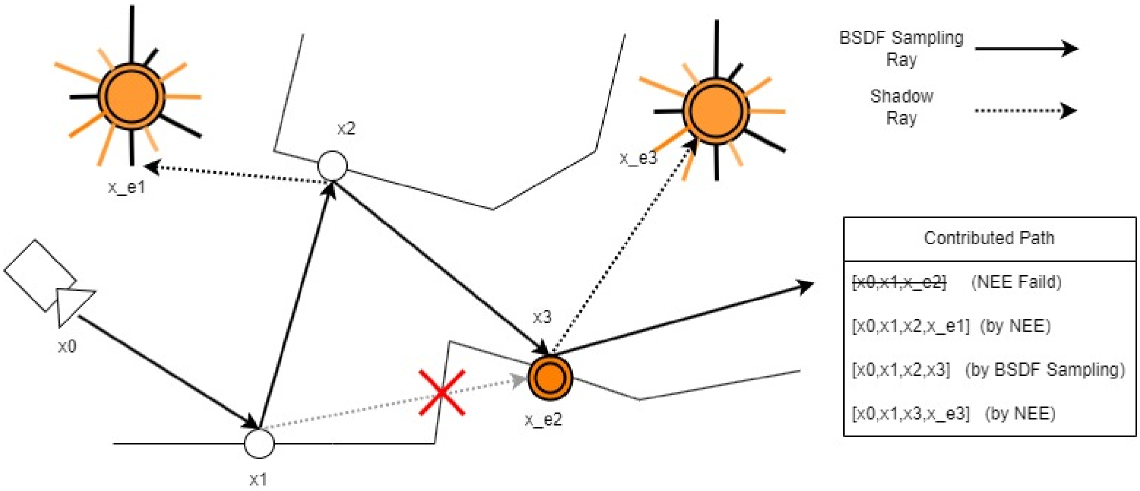

図 **2.15:** NEEの概略図

このようなサンプリングを行うことで、L_e項が非ゼロ値である状況を明示的に作ることができ、分散の低減につながる。
ただし、サンプリングした$\boldsymbol{x_D}$についてBSDF$f(\boldsymbol{x_D} \rightarrow \boldsymbol{x_{D-1}} \rightarrow \boldsymbol{x_{D-2}})$が0の場合は寄与が0となり、そのサンプリングは無駄であることに注意する必要がある。このようなケースは完全鏡面のようなBSDFにおいて簡単に発生する。また、接続した点について可視性が認められない場合、同様にその点からの寄与は0となる。
すなわち、NEEではパスの構築にBSDFサンプリングを使用した上で、光源とパスを明示的に接続して寄与を計算する。接続した点への可視性の確認には、一般にShadow Rayの発射と呼ばれる衝突判定結果のみを返すレイキャストを行う。以下に、NEEを行うコード2を示す。
また、図2.16にNEEパストレーシングによる生成画像の例を示す。

その他の光輸送アルゴリズム

Kajiyaが提案したパストレーシング法はカメラから光源に向かってパスを構築していく、単方向(Uni-directional)パストレーシングだが、カメラと光源の双方向(Bi-directional)のパストレーシング[18]も提案されている。
双方向光輸送アルゴリズムに他に代表的なものにフォトンマッピング法[19]があり、これはパストレーシングでは表現が困難なコースティクス現象の再現が

```
struct {
    ...
} Ray
struct {
    ...
} HitResult
struct {
    float3 x,n,Le
    float pdf
} LightSample
begin
    I ⟵ 0.0
    for i = 0toN do
        ray ⟵ getViewRay(p)
        ray.thp ⟵ 1.0
        result ⟵ traceRay(ray)
        while result.hit = True do
            if result.x.hasEmission then
                I ⟵ I + ray.thp ∗ Le(result.x, result.wo)
            end
            if bsdf.hasNonDeltaReflection then
                ls ⟵ sampleLight()
                wi ⟵ normalize(ls.x − result.x)
                shadowRay ⟵ Ray(result.x, wi, 1.0f)
                anyHit ⟵ traceShadowRay(shadowRay)
                if !anyHit then
                    f ⟵ bsdf.eval(result.x, result.wo, wi)
                    G ⟵ dot(result.n, wi) ∗ dot(ls.n, −wi)/dot(result.x, ls.x)
                    thp ⟵ (ray.thp ∗ f ∗ G)/ls.pdf
                    I ⟵ I + thp ∗ ls.Le
                end
            end
            bsdfSample ⟵ bsdf.sample(result)
            ray.thp ⟵ (thp ∗ bsdf.eval(result.x, result.wo, bsdfSample.wi) ∗
              dot(result.n, bsdfSample.wi))/bsdfSample.pdf
            Prr ⟵ russianRoulette()
            if random() ≥ Prr then
                break
            end
            ray.thp ⟵ ray.thp/Prr
            ray.origin ⟵ result.x
            ray.direction ⟵ bsdfSample.wi
            result ⟵ traceRay(ray)
        end
        return I/N
    end
end
```

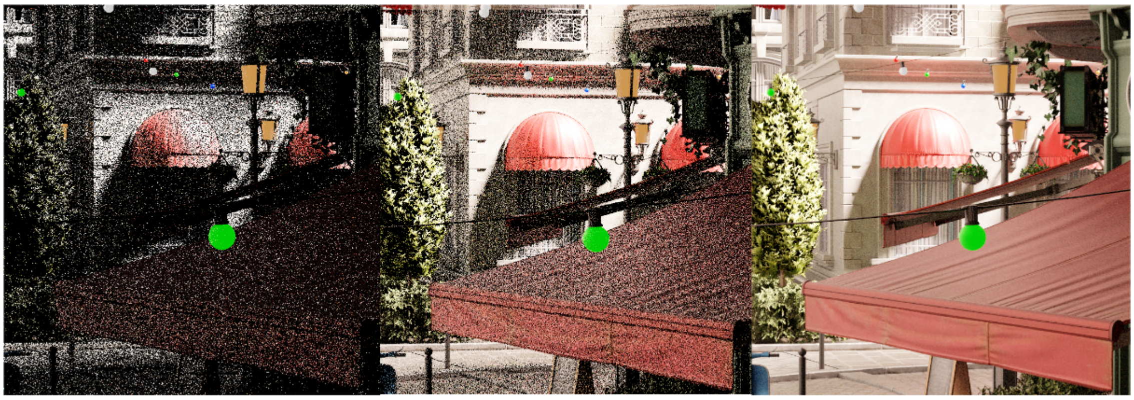

図 **2.16:** パストレーシングによる写実的な CG 画像。左からサンプル数 1, サンプル数 10, サンプル数 1024。

容易であることが知られている。フォトンマッピング法は空間計算量が N^3 に比例しメモリ消費が激しいが、これを改善したプログレッシブフォトンマッピング法 [20] も過去に提案されている。
パストレーシング法の改善手法として著名なものに、Veach らが開発した Metropolis Light Transport(MLT) [21] がある。これはメトロポリス-ヘイスティングス法を光輸送の問題に適用し、シーン中に存在する光源の面積が非常に少ない、間接光の寄与が大きいシーンでもノイズに対し頑健な結果を得ることができる。
近年では、光の干渉等の波動性を考慮したレンダリング方程式の拡張、Physically Light Transport が提案されている [22]。

2.3.5 スペクトラルパストレーシング

スペクトラルパストレーシングにおいても通常の RGB パストレーシングと同様に式 2.3.1 を解くことを考える。
ただし、明示的に波長の積分を考える必要があり、式 2.3.1 は以下のように拡張される。

$$L_{\lambda o}(p, v, \boldsymbol{\lambda}) = L_{\lambda e}(p, v\boldsymbol{\lambda}) + \int_{l\in\Omega,\lambda\in\Lambda} f_s(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{\lambda}) L_{\lambda i}(p, \boldsymbol{l}, \boldsymbol{\lambda})(n, \boldsymbol{l})^+ d\boldsymbol{l} \tag{2.3.12}$$

$$L_o(p, v) = \int_{\lambda\in\Lambda} L_{\lambda e}(p, v, \boldsymbol{\lambda}) + \int_{l\in\Omega,\lambda\in\Lambda} f_s(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{\lambda}) L_{\lambda i}(p, \boldsymbol{l}, \boldsymbol{\lambda})(n, \boldsymbol{l})^+ d\boldsymbol{l} \tag{2.3.13}$$

経路積分方式のレンダリング方程式 2.3.5 についても、同様に波長の積分について考えればよい。

このとき、構築したパスが与える寄与は、そのパスが持つ波長の積分になる。ゆえに、2.1.2 で述べたように等色関数による畳み込みを行うことで値を得ることができる。
すなわち、ピクセル j の XYZ 成分は

$$I_{X,j} = \frac{1}{\int_\Lambda f_{\bar{y}}(\lambda)d\lambda} \int_\Lambda \bar{x}(\lambda) \int_{l\in\Omega} f_s(\boldsymbol{l}, v_j, \lambda) L_{\lambda i}(p_j, \boldsymbol{l}, \lambda)(n_j, \boldsymbol{l})^+ d\boldsymbol{l} d\lambda \tag{2.3.14}$$

$$I_{Y,j} = \frac{1}{\int_\Lambda f_{\bar{y}}(\lambda)d\lambda} \int_\Lambda \bar{y}(\lambda) \int_{l\in\Omega} f_s(\boldsymbol{l}, v_j, \lambda) L_{\lambda i}(p_j, \boldsymbol{l}, \lambda)(n_j, \boldsymbol{l})^+ d\boldsymbol{l} d\lambda \tag{2.3.15}$$

$$I_{Z,j} = \frac{1}{\int_\Lambda f_{\bar{y}}(\lambda)d\lambda} \int_\Lambda \bar{z}(\lambda) \int_{l\in\Omega} f_s(\boldsymbol{l}, v_j, \lambda) L_{\lambda i}(p_j, \boldsymbol{l}, \lambda)(n_j, \boldsymbol{l})^+ d\boldsymbol{l} d\lambda \tag{2.3.16}$$

をそれぞれ解くことによって求まる。
スペクトラルレンダリングは RGB レンダリングにおける輝度分散に加えて波長による色分散も発生させることになる。

2.4 サンプリング

レンダリング方程式を評価する際、式 2.3.7 の確率密度関数 p の評価のように、ある集合から要素を取り出すサンプリング操作を行う必要がしばしば発生する。
この章では、物理ベースレンダリングにおけるサンプリングの基礎について簡単に解説し、また本研究の根幹をなす再サンプリングについてもここで説明する。

2.4.1 再サンプリング

重点的サンプリングは、サンプルの生成に使用される確率密度関数 p によってその有効性が大きく変わる。しかし、理想的なサンプルを生成する確率密度関数を定式化することは一般に難しい。
また、定式化してもその確率密度関数を評価することが困難であったり実用上無意味な場合もある。
本節ではそのような確率密度関数の定式化が困難な場合において、モンテカルロ積分による不偏な推定値を与えることを考える。
まずそのような確率密度関数の代替として機能する不偏寄与度重み (Unbiased Contribution Weight) について解説し、次にその不偏寄与度重みを生成する手法の一つである再サンプリングについて解説する。

不偏寄与重み (UCW)

式 2.2.1 で示したように、ある関数 f の積分値 I があるとき、

$$I = \int_{\Omega} f(\boldsymbol{x}) d\boldsymbol{x}$$

それをモンテカルロ法を用いて近似計算した場合、その期待値は以下のように表すことができた。

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(\boldsymbol{x_i})}{p_i(\boldsymbol{x_i})}$$

しかし、光輸送アルゴリズムの複雑さゆえに、あるサンプルを生成した x について理想的な確率密度関数 p を評価することが非常に困難な場合がある。
そのような場合でも x について、p の逆数と期待値が一致するような確率変数 W_x がわかっていれば、不偏なモンテカルロ積分は評価することができる。この W_x のことを不偏寄与重み (Unbiased Contribution Weight, UCW) と呼び、以下のように定義する。

定義 2.3 不偏寄与重み *(Unbiased Contribution Weight, UCW)*

$$\begin{aligned} &\langle I \rangle = f(x) W_x \\ &\textit{with} \quad \mathbb{E}[f(x)W_x] = \mathbb{E}[f(x)/p(x)] = \int_{\Omega} f(x) dx = I \end{aligned}$$

再サンプリングされた重点的サンプリング (RIS)

UCW を求める手法の一つに、再サンプリングされた重点的サンプリング (Resampled Importance Sampling, RIS) [23] がある。
Talbot による最も基本的な RIS では、UCW を求めるために、対応する確率密度関数の評価が困難な目標関数 $\hat{p}$ があるとき、まず容易に評価できる確率密度関数 p の分布に従い候補サンプルを M 個生成 $\boldsymbol{x} = [x_1, x_2, x_3, ..., x_M \in \Omega]$ する。
次に、それぞれの候補サンプルに対し、リサンプリング重み w_{x_i} を以下のように計算する。

$$w_{x_i} = \frac{\hat{p}(x_i)}{p(x_i)} \tag{2.4.1}$$

このリサンプリング重みに比例した確率で、候補サンプルから一つのサンプル x_s をサンプリングする。

このとき、不偏寄与重み W_x は以下のように計算される。

$$W_x = \frac{1}{\hat{p}(x_z)}\Big(\frac{1}{M}\sum_{i=1}^{M} w_{x_i}\Big) \tag{2.4.2}$$

この W_x は評価が困難な確率密度関数の代替として機能し、$M \to \infty$ の時、W_x の分布は目標関数 $\hat{p}$ に完全に比例した分布を持つ。
この基本の RIS は候補サンプルが共通のドメインに属し、共通の分布から生成されるときにのみ成り立つ。

Talbot は候補サンプルが異なる分布から生成される場合の RIS も提案している。本論文では Lin らの精神に基づき、リサンプリング MIS 重みを導入し、この RIS を以下のように定義する。

1. 共通のドメイン Ω に属する候補サンプルを M 個生成 $\boldsymbol{x} = [x_1, x_2, x_3, ..., x_M], (x_i \in \Omega)$ する。
2. 全ての候補サンプルについて、

$$m_i \geq 0 \quad and \quad \sum_{i=1}^{M} m_i(x) = 1, \forall x \in supp(\hat{p})$$

   を満たすようなリサンプリング MIS 重み $m_i(x_i)$ を評価する。
3. 目標関数 $\hat{p}$ に対し、リサンプリング重み $w_i = m_i(x_i)\hat{p}(x_i)W_{x_i}$ を評価する。
4. リサンプリング重み w_i に比例した確率で、候補サンプルから一つのサンプル x_s をサンプリングする。
5. 不偏寄与重み $W_x = \frac{1}{\hat{p}(x_s)} \sum_{i=1}^{M} w_{x_i}$ を評価する。

このとき、モンテカルロ法を適用する被積分関数 f に対し、$\hat{p}$ と p が以下の条件

$$supp(f) \subset \Big(supp(\hat{p}) \cap (\cup_{i=1}^{M} supp(x_i))\Big) \tag{2.4.3}$$

を満たすとき、RIS のモンテカルロ積分の推定量 $\langle I \rangle = f(x)W_x$ は不偏となる。

$$\mathbb{E}[\langle I \rangle] = \int_{\Omega} f(x)dx = I \tag{2.4.4}$$

言い換えれば RIS の推定量を不偏に保つためには被積分関数の台、つまり 0 以上の”推定量に寄与する値をもつ範囲”は、目標関数の台と初期候補サンプルの生成範囲双方によって完全に覆われている必要がある。

共通のドメインΩに属する候補サンプルをM個生成するとき、容易に評価できある分布Dに従う確率密度関数pを用いて、$p(d_i), (1 \leq i \leq M)$を評価することで生成する場合、リサンプリングMIS重みのUCWであるW_{x_i}は、pが容易に評価できることにより$W_{x_i} = \frac{1}{p(x_i)}$となる。
このとき、リサンプリングMIS重みの定式化として以下のようなバランスヒューリスティックスに類似したものが提案されている。

例 1 共通ドメインからのサンプル生成時の再サンプリング*MIS*重み

$$m_i(x) = \frac{p_i(x)}{\sum_{j=1}^{M} p_j(x)}$$

また、分布Dが一様なら、$m_i(x_i) = \frac{1}{M}$となる。
このとき、式2.4.3の条件は次のように書くこともできる。

$$\hat{p}(x_i) > 0, p(x_i) > 0, \forall x_i \in supp(f)$$

このRISを評価する疑似コードを以下3に示す。
このRISを使用する上で特別に注意すべき点として、先に述べた条件2.4.3がある。
これが意味することは、被積分関数fが0より大きい値をとる領域Ω_{sub}について、目標関数$\hat{p}$が0より大きい値をとる領域$\hat{\Omega_{sub}}$に含まれている必要がある。
またΩ_{sub}が含む全てのxはpによってサンプリングされる可能性がある必要がある。つまりあるxをサンプルする確率が0であるような分布はpとして不適切であり、偏った結果をもたらす。

```
struct {
    Sample s
    float ucw
} RISSample
Function RandomIndex(float[] w):
    float r = random()
    for i = 0 to M do
        if w[i] > 0.0f then
            r− = w[i]/sum(w)
            if r <= 0.0f then
                return i
            end
        end
    end
    return ∅
Function ResampliedImportanceSampling(M):
    float[] w(M)
    Sample[] x(M)
    for i = 0 to M do
        [x_i, ucw] ⟵ sampleGen()
        x[i] ⟵ x_i
        w[i] ⟵ m_i(x_i) ∗ p(x_i) ∗ ucw
    end
    RISSample    s
    s.s ⟵ ∅
    s.ucw ⟵ 0.0f
    index ⟵ RandomIndex(w)
    if index ≠ ∅ then
        s.s ⟵ x[index]
        float w_sum ⟵ 0.0f
        for i = 0 to M do
            w_sum+ = w[i]
        end
        s.ucw ⟵ (1/p(x[index])) ∗ w_sum
    end
    return s
```

Algorithm 3: RIS の疑似コード

RIS の一般化 (GRIS)

前節での RIS は、生成する候補サンプルが共通のドメイン Ω に属するという制約があった。
ここで、候補サンプルが共通のドメインに属する必要がない一般化された RIS(Generalized Resampled Importance Sampling, GRIS) が Lin らによって提案されている [4]。
これは、潜在的に相関のある入力サンプル列 $[(x_i \in \Omega)_{i=1}^{M}]$ を任意のソースドメイン Ω_i からサンプリングすることを許す。このとき、各 x_i について、不偏寄与重み W_{x_i} があれば不偏なモンテカルロ積分の推定量を与えることができる。
この手順を以下に示す。

1. サンプル毎のドメイン Ω_i に属する候補サンプルと、その不偏寄与重みの組を M 個用意 $\boldsymbol{x} = [(x_1, W_{x_1}), (x_2, W_{x_2}), (x_3, W_{x_3}), ..., (x_M, W_{x_M})]$ する。
2. シフト関数 T_i を用いて、各 x_i を $y_i = T_i(x_i), y_i \in \Omega$ として目標ドメイン Ω に写像する。
3. 全ての写像済み候補サンプル $\boldsymbol{y} = [y_1, y_2, ..., y_M]$ について、

$$m_i \geq 0 \quad and \quad \sum_{i=1}^{M} m_i(y_i) = 1, \forall y \in T_i(supp(\boldsymbol{x}))$$

   を満たすようなリサンプリング MIS 重み $m_i(y_i)$ を評価する。
4. 目標関数 $\hat{p}$ に対し、全ての写像済み候補サンプル $\boldsymbol{y}$ と対になるリサンプリング重み $w_i = m_i(y_i)\hat{p}(y_i)W_{x_i}|\frac{\nabla T_i}{\nabla x_i}|, (0 \leq i < M)$ を評価する。
5. リサンプリング重み w_i に比例した確率で、写像済み候補サンプルから一つのサンプル y_s をサンプリングする。
6. 不偏寄与重み $W_y = \frac{1}{\hat{p}(y_s)} \sum_{i=1}^{M} w_{y_i}$ を評価する。

GRIS におけるリサンプリング MIS 重みの設計として、写像後の目標ドメイン Ω とサンプル $\boldsymbol{y}$ について、その確率密度関数 p_{y_i} が定式化でき容易に評価できれば以下のように書ける。

$$m_i(y_i) = \frac{p_{y_i}(y_i)}{\sum_{j=1}^{M} p_{y_j}(y_j)} \tag{2.4.5}$$

しかし実際には $y_i = T_i(x_i)$ のように T で写像したのちのサンプルについてその確率密度関数 p_{y_i} を評価することは困難である。そこで Lin らは”i 番目における

代理の $\hat{p}$"を以下のように定義し、

$$\hat{p}_{\leftarrow i}(y) = \begin{cases} \hat{p}_{x_i}\Big(T_i^{-1}(y)\Big)|\frac{\nabla T_i^{-1}}{\nabla y}| & \text{if } y \in T_i(supp(x_i)) \\ 0 & \text{otherwise} \end{cases} \tag{2.4.6}$$

GRIS におけるリサンプリング MIS 重みとして、以下の提案を行っている。

$$m_i(y_i) = \frac{\hat{p}_{\leftarrow i}(y_i)}{\sum_{j=1}^{M} \hat{p}_{\leftarrow j}(y_j)} \tag{2.4.7}$$

RIS と同様にこのリサンプリング MIS 重みは条件

$$\sum_{i=1}^{M} m_i(y_i) = 1, (y_i \in supp(x_i)) \tag{2.4.8}$$

を満たさなければならない。
また、被積分関数 f に対し、条件

$$supp(f) \subset \Big(supp(\hat{p}) \cap (\cup_{i=1}^{M} T_i(supp(x_i)))\Big) \tag{2.4.9}$$

を満たすとき、GRIS のモンテカルロ積分の推定量 $\langle I \rangle = f(\mathbf{y})W_{\mathbf{y}}$ は不偏となる。ここで、式 2.4.6 における条件は、単にある $y \in \Omega$ について候補サンプル x_i 固有のドメイン Ω_i への写像が不可能であるか x_i がゼロ確率でサンプリングされる場合に 0 を返す、という意味である。推定量を不偏に保つためには条件 2.4.9 を満たす必要があるため、このとき Ω_i への写像は必ず可能であるため、この条件は単に x_i が評価できないならば 0 を返す、という意味である。

式 2.4.8 と 2.4.9 に対する視覚的・直感的な解釈として、目標ドメイン Ω は候補サンプルのドメイン Ω_i の和集合であるが、重複して覆われる領域が存在する場合がある。このとき、重複した領域においては、その領域を覆うドメイン全てから合計で正確に 1 つぶんのサンプルによってカバーされるように重み付けをしなければならないという意味である。

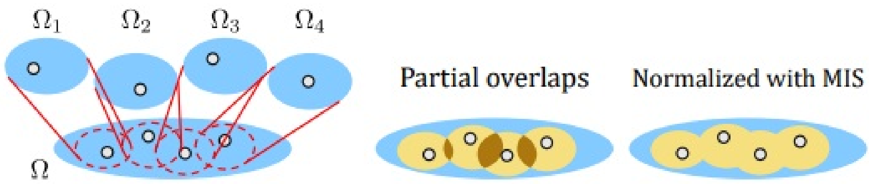

図 2.17: 式 2.4.8 と 2.4.9 の解釈 ([4] より引用)。複数のドメインから不偏な統合を行うには、重複領域のための適切な再サンプリング MIS 重みが必要になる。

第3章

関連研究

3.1　再サンプリングの照明計算への応用

この章では、2.4.1で触れた再サンプリングを光輸送問題、特にパストレーシング法による直接照明と間接照明の計算に応用した研究について述べる。

3.1.1　再サンプリング手法の効率的なストリーム入力

アルゴリズム3に示した再サンプリング手法は、全ての候補サンプルを事前に生成しそのすべてをメモリ上に保持する必要がある。これは光輸送アルゴリズムにとって、特にGPUのような並列計算環境では、メモリの使用量が大きく効率的な計算ができない。また任意長のストリーム入力を与えることもできず、実応用的な実装に厄介な問題となる。
Chaoらによる Weighted Reservoir Sampling(WRS) [24] は、1つ分のサンプルを保持するメモリ量のみで任意長の入力ストリームから一つ以上の要素をサンプリングすることができる手法である。これはRISに完全に適合する。
WRSを使用したRISの疑似コードをアルゴリズム4に示す。

3.1.2　再サンプリングと直接照明

1回目のNEEサンプリング、つまり直接照明の計算でRISを使用し、画像の輝度分散を低減することを考える。
直接照明は、一回バウンスのパス、すなわち $\boldsymbol{x} = [x_0, x_1, x_2], x \in \mathcal{A}$ となる全てのパスに対して、そのパスの寄与を計算することで得られる。
すなわち、以下の式を画像の各ピクセルごとに評価すれば良い。

$$L(x_1 \to x_0) = \int_{\mathcal{A}} f(x_2 \to x_1 \to x_0) G(x_1 \leftrightarrow x_2) V(x_1 \leftrightarrow x_2) L_e(x_2 \to x_1) dx_2 \tag{3.1.1}$$

class {
  int M
  float w_{sum}
  float $\hat{p}$
  Sample sample
  Function `Update`(*Sample* s_i, *Float* w_i, *Float* $\hat{p}_i$, *Float rand01*)**:**
    $w_{sum} \leftarrow w_{sum} + w_i$
    if $rand01 < \frac{wi}{w_{sum}}$ **then**
      sample $\leftarrow s_i$
      $\hat{p} \leftarrow \hat{p}_i$
    end
    M++
    return *true*
} *Reservoir*
Function *WeightedReservoirSampling***:**
  Reservoir r
  r.M $\leftarrow 0$
  r.$w_{sum} \leftarrow 0$
  r.$\hat{p} \leftarrow 0$
  Sample sample $\leftarrow \emptyset$
  for $i \leftarrow 0$ **to** N **do**
    Sample s_i, **float** $ucw \leftarrow$ sampleGen()
    float $\hat{p}_i \leftarrow \hat{p}(s_i)$
    float $w_i \leftarrow m(s_i)\hat{p}_i ucw$
    float rand01 $\leftarrow$ rand()
    bool accepted $\leftarrow$ r.`Update`(s_i, w_i, $\hat{p}_i$, *rand01*)
  end
  return *r.sample*

Algorithm 4: WRS を使用した RIS

ここで、頂点 x_0 と x_1 は各ピクセル毎に固定されているため、この積分は x_2 についてのみの積分となり、またこの x_2 は光源サーフェスであるからこれを x_e とすると、この積分は以下のように書ける。

$$L(x_1 \to x_0) = \int_{\mathcal{A}} f(\boldsymbol{x_e}) d\boldsymbol{x_e} \tag{3.1.2}$$

Talbot らは、直接照明における目標関数 $\hat{p}$ として以下の式を設定した。

$$\hat{p}(x) = f_s(x)G(x)L_e(x) \tag{3.1.3}$$

可視項 V は式 3.1.1 で計算時間が支配的であるため、性能上の理由からこれを含めない。
また、RIS の対象は光源サンプリングである。
候補サンプルを一様分布からサンプリングするとき、このアルゴリズムは以下のように書ける。

1. シーンの光源サーフェスの集合 A に属する候補サンプルをある確率密度関数 p の分布に従って M 個生成 $\boldsymbol{x_e} = [x_{e1}, x_{e2}, x_{e3}, ..., x_{eM} \quad | \quad x_i \in \mathcal{A}]$ する。
2. 全ての候補サンプルについて、リサンプリング重み $w_i = \frac{1}{M} f_s(x_{ei})G(x_{ei})L_e(x_{ei})\frac{1}{p(x_{ei})}$ を評価する。
3. リサンプリング重み w_i に比例した確率で、候補サンプルから一つのサンプル x_s をサンプリングする。
4. 不偏寄与重み $W_x = \frac{1}{f_s(x_s)G(x_s)L_e(x_s)} \sum_{i=1}^{M} w_{x_{ei}}$ を評価する。

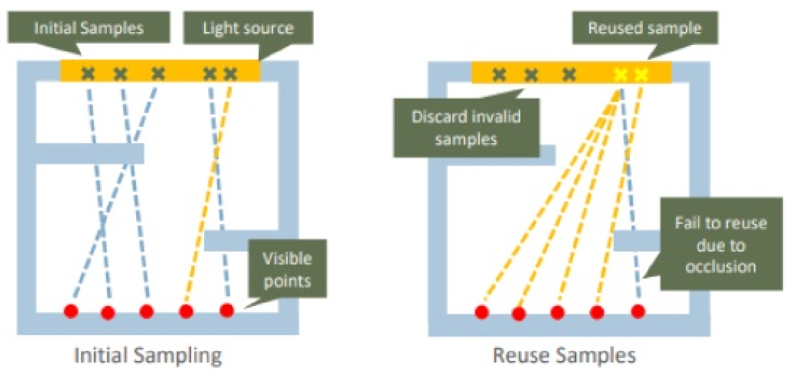

図 **3.1:** RIS によるサンプル再利用の図式 ([5] より引用)

BSDF サンプリングと NEE サンプリングの結合

前節では一様分布からの NEE サンプリングを候補サンプルの生成に使用したが、より複雑な例として PDFp_{BSDF} を持つ BSDF 重要度サンプラーから M_{BSDF} 個の候補を、PDFp_{NEE} を持つ NEE ライトサンプラーから M_{NEE} 個の候補を生成する場合を考える。
このとき、BSDF サンプリングに対するリサンプリング MIS 重み及びリサンプリング重みは次の通りである。

$$m_i(x) = \frac{p_{BSDF}(x)}{M_{BSDF}p_{BSDF}(x) + M_{NEE}p_{NEE}(x)} \tag{3.1.4}$$

$$\begin{aligned} w_i =& m_i(x_i)\hat{p}(x_i)W_{x_i} \\ =& \Big(\frac{p_{BSDF}(x_{BSDF})}{M_{BSDF}p_{BSDF}(x_{BSDF}) + M_{NEE}p_{NEE}(x_{NEE})}\Big)\hat{p}(x_{BSDF})\frac{1}{p_{BSDF}(x_{BSDF})} \end{aligned} \tag{3.1.5}$$

不偏寄与重み W_x は

$$W_{x_s} = \frac{1}{\hat{p}(x_s)}\sum_{j=1}^{M_{NEE}+M_{BSDF}} w_j \tag{3.1.6}$$

時空間サンプリング

Biterrli ら [6, 2020] は、直接照明の計算において WRS を用いて時空間方向に再サンプリングを行い、分散を低減する手法 ReSTIR(Reservoir based Spatio-Temporal Importance Resampling) を提案した。
これは過去フレームからのサンプル (時間方向)、現在フレームの近傍ピクセルからのサンプル (空間方向) を WRS の対象とするものである。
過去、近傍からのサンプルは異なるピクセル、すなわち 1 回バウンスパスのセンサー頂点 x_0、一次衝突頂点 x_1 が異なることに注意しなければならない。
このとき、直接照明における NEE 時空間サンプリングでは、サンプル x_e は全ピクセルで共通のドメイン $\mathcal{A}$ からサンプリングされ、シーンが静的であれば全時間においても共通のドメイン $\mathcal{A}$ からサンプリングされる。しかし、各サンプルの分布は各ピクセルにおいて、センサー頂点、一次衝突頂点が異なるためにその分布は異なる。すなわち、共通のドメイン、異なる分布からの RIS を適用する必要がある。
時空間サンプルの対象となる異なるピクセル j におけるセンサー頂点、一次衝突頂点をそれぞれ $x_{0,j}, x_{1,j}$ とすると、適切なリサンプリング MIS 重みとして以下のようなものがある。

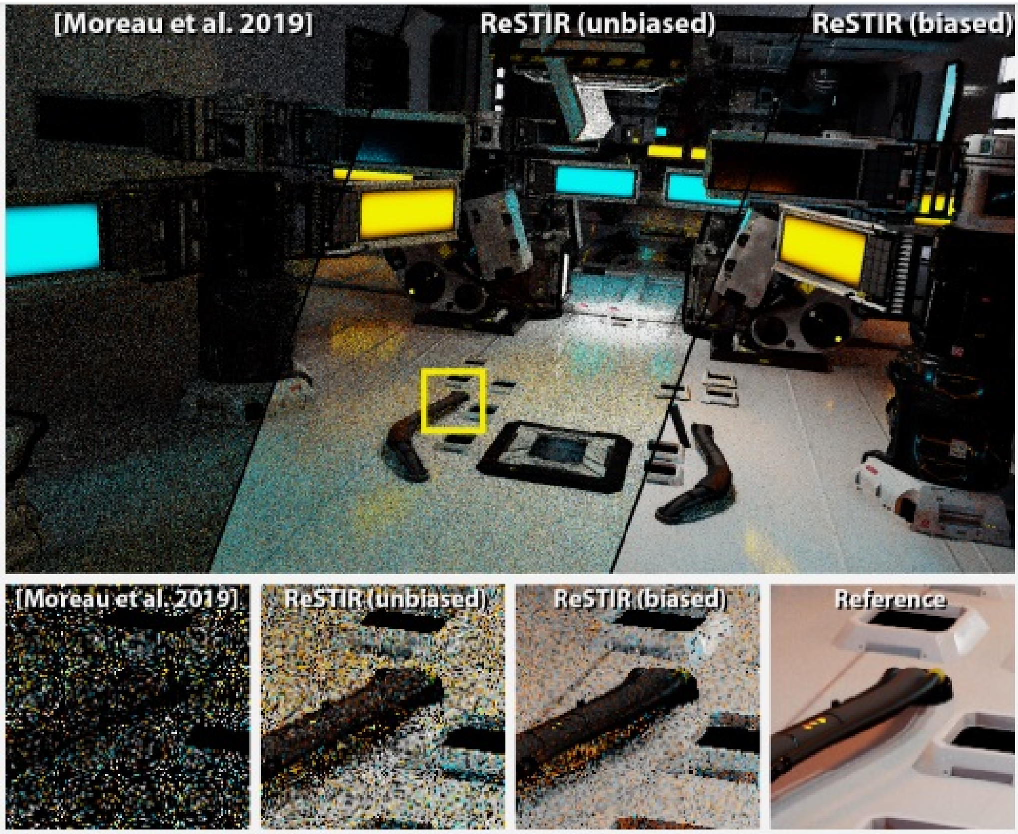

図 3.2: Moreau ら [2019] による Dynamic Many-Light Sampling for Real-Time Ray Tracing と ReSTIR アルゴリズムの比較 ([6] より引用)。Biterrli らは Spatial Resampling 時に、Bias を導入する代わりにより高速な MIS Weight を使用した手法も [6] で提案している。

$$
\begin{aligned}
m_i(\boldsymbol{x_e}) &= \frac{f(\boldsymbol{x_e} \rightarrow x_{1,i} \rightarrow x_{0,i})}{\sum_{j=1}^{M} f(\boldsymbol{x_e} \rightarrow x_{1,j} \rightarrow x_{0,j})} \\
with \quad & f = f_s \cdot G \cdot L_e
\end{aligned}
\tag{3.1.7}
$$

空間再利用:現在のピクセルの相対的な近傍、例えば現在ピクセルを中心とするある半径の円から適切な数のピクセルを選択し、そのピクセルからのサンプルを WRS の対象とする。

相対的な近傍の選択には、例えば G-Buffer からの深度情報やラフネス情報を用いることで類似したピクセルを選択できる。空間再利用によるサンプルで $\frac{1}{M}$ のリサンプリング MIS 重みを選択した場合、それは目標関数 $\hat{p}$ への収束保証を失い、バイアスの原因となる。

時間再利用:過去フレーム、一般には現在フレームの一つ前のフレームからのサンプルを WRS の対象とする。

現在フレームにおける対象ピクセル i に対応する前フレームのピクセル i' を特

定するため、Motion-VectorやOptical-Flowを用いたフレーム間移動の追跡が必要となる。
動的シーンにおいてはリサンプリングMIS重みの計算に可視性項Vの再計算が必要となる。

3.1.3 任意の光輸送経路の再サンプリング

直接照明だけでなく、シーン全体の大域照明についての再サンプリングを考える。式3.1.2を任意のパス長について考えると、それは面積測度におけるレンダリング方程式、式2.3.5についてのパス再利用となる。

$$L_o(\boldsymbol{x_1} \to \boldsymbol{x_0}) = L_e(\boldsymbol{x_1} \to \boldsymbol{x_0}) + \sum_{D=2}^{\infty} \int_{\mathcal{A}^{D-1}} (\prod_{i=1}^{D-1} f(\boldsymbol{x_{i+1}} \to \boldsymbol{x_i} \to \boldsymbol{x_{i-1}}) G(x_{i+1} \leftrightarrow x_i) \\ V(x_{i+1} \leftrightarrow x_i)) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}) d\boldsymbol{x_2} ... d\boldsymbol{x_D} \tag{3.1.8}$$

センサー頂点、一次衝突頂点が各ピクセル毎に固定されていると考えると、この積分はx_2からx_Dについての積分となり、以下の微小面積のベクトルに関する再サンプリングを行えばよい。

$$[x_0, x_1, \boldsymbol{x_2}, \boldsymbol{x_3}, ..., \boldsymbol{x_D}] \tag{3.1.9}$$

このパスに対してReSTIRアルゴリズムを適用する研究は近年活発に行われており、ReGIR [25, 2021]、ReSTIR GI [5, 2021]などがある。
それらの発展手法は任意のパスを扱うことが出来ず、何らかのBiasと無視できないアーティファクトを発生させていたが、GRISの登場[4, 2022]により完全なパス再サンプリングが可能となった。

パス空間におけるパスの変換

あるパスxを、シフトマッピング関数Tを用いて別のパスyに変換することを考える。近年のパス再サンプリング研究の結果を踏まえ、Wymanら[2023]は以下の考察を与えている。

最も単純なTとして、面積測度においての同一シフトがある。これは、式3.1.2を単に任意のパスに拡張する。

$$T([x_0, x_1, \boldsymbol{x_2}, \boldsymbol{x_3}, ..., \boldsymbol{x_D}]) = [x_0, x_1, \boldsymbol{y_2}, \boldsymbol{y_3}, ..., \boldsymbol{y_D}] \tag{3.1.10}$$

このTは、y_1から$\boldsymbol{x_2}$を適切に接続するために点y_1と点$\boldsymbol{x_2}$のBSDF、ジオメトリ項G、可視性項Vを評価する。そして、例えば$Le(\boldsymbol{x_2} \to x_1)$を$Le(\boldsymbol{x_2} \to y_1)$に再利用できる。
Ouyangら[5, 2021]のReSTIR GIはこの戦略を利用したが、これは2点のBSDF

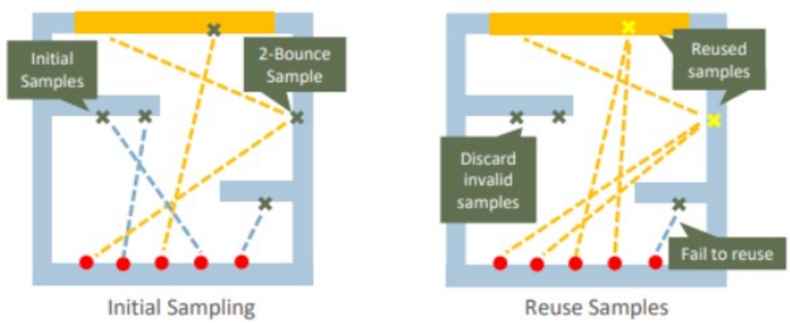

図 **3.3:** 単純な再接続([5]より引用)

ローブが鋭い、例えば完全鏡面などの場合は逆に分散を劇的に増加させる結果となる。

低分散で不偏な結果を与える理想的なシフトマッピング関数Tについて、以下の性質が考えられる。

$$\bar{p}_j(T(x))\left|\frac{\nabla T}{\nabla x}\right| \simeq \bar{p}_i(x) \tag{3.1.11}$$

ここで$\bar{p}_k$はピクセルkにおいてパスxをサンプリングする分布を与える確率密度関数であり、iは目標ピクセル、jは再利用されるピクセルである。
この式は、理想的なTはシフトしたサンプルの目標PDFを元の目標PDFとほぼ等しくすることを意味する。
すなわち、$\bar{p}_j(T(x))$と$\bar{p}_i(x)$が極端に異なる場合、Tは不適切であると言えるが、これはまさにパスxについて接続される2点のBSDFローブが鋭い場合に該当する。

図 3.4: ReSTIR GI によるパス再利用のアーティファクト ([7] より引用)。 透過マテリアルや Glossy マテリアルでは、偏った結果をもたらす

そのような場合はピクセル i にパスをシフトした場合、ソースピクセル i にお

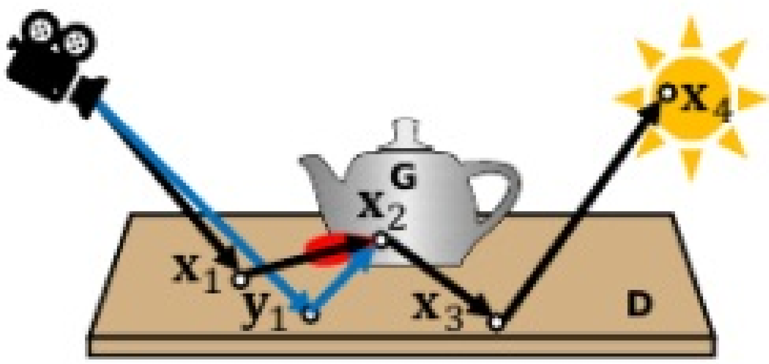

図 3.5: 頂点再接続に失敗する例。[7] から引用。

ける寄与がほぼ 0 になる場合が十分に考えられるためである。
また、$\|x_2 - x_1\|$ と $\|x_2 - y_1\|$ が極端に異なる場合も、幾何項 G の働きにより不適切な T を生み出す要因となる。
よって、このような単純な T は避けるべきである。

パスシフトマッピング関数 T の詳細な研究は、勾配領域レンダリング [26] において広範囲に研究されている。これら先行研究によれば、画像の勾配の局所

的変化がなめらかであれば、近傍ピクセルは類似した正規化係数をもつはずであるとされる。
この事実と目標関数が被積分関数に類似した設計をされる事実 $\hat{p} \simeq f$ に注目すれば、式 3.1.11 は以下のように書ける。

$$f_j(T(x))\left|\frac{\nabla T}{\nabla x}\right| \simeq f_i(x) \tag{3.1.12}$$

これは、勾配領域レンダリングで使用されるのと同じ条件であり、それら分野のシフトマッピングがパス再サンプリングにも適用できることを意味する。
Lin ら [4, 2022] はパス再サンプリングのためのシフトマッピング関数 T に有用なものとして、Lehtinen ら [27, 2013] の頂点コピー、多様体探索 [28, 2020], [29, 2012]、Kettunen ら [30, 2015] のハーフベクトルコピー、各頂点におけるサンプリング乱数再利用、パス方向コピーなどを挙げている。

ReSTIR PT におけるパスシフト戦略

Lin ら [2022] は勾配領域パストレーシング [2015] のシフト戦略に類似した、パス再サンプリングのためのハイブリッドシフトマッピング戦略 (Hybrid Shift Mapping) を提案した。
この戦略は、乱数再利用戦略と頂点コピー戦略を組み合わせたものである。パスの各頂点ごとに、距離条件と粗さ条件を確認し、有効であれば頂点コピーを行う。そうでなければ乱数コピーを行い、頂点コピーを遅延する。

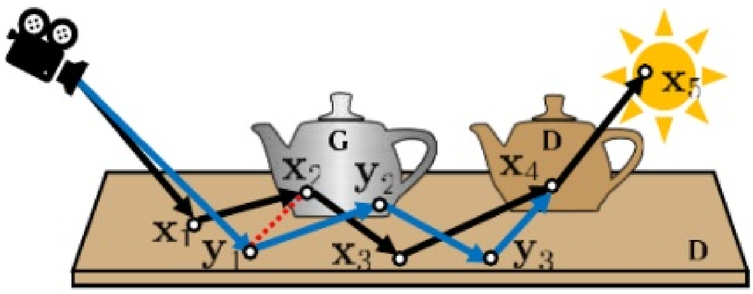

図 **3.6:** ハイブリッドシフト戦略。[4] から引用。

ここで、ベースパス $\boldsymbol{x}$ に含まれる頂点 $\boldsymbol{x}_k$ と、オフセットパス $\boldsymbol{y}$ に含まれる頂点 $\boldsymbol{y}_k$ について、

距離条件とは、

$$min(\|\boldsymbol{x_k} - \boldsymbol{x_{k-1}}\|, \|\boldsymbol{x_k} - \boldsymbol{y_{k-1}}\|) \geq d_{min} \tag{3.1.13}$$

を満たすことである。
粗さ条件とは、

$$min(\alpha_{x_{k-1}}(l_{k-1}), \alpha_{y_{k-1}}(l`_{k-1}), \alpha_{x_k}(l_k)) \geq \alpha_{min} \tag{3.1.14}$$

を満たすことである。$l_{k-1}, l`_{k-1}, l_k$ はそれぞれ頂点 $\boldsymbol{x_{k-1}}, \boldsymbol{y_{k-1}}, \boldsymbol{x_k}$ における BSDF ローブであり、$\alpha_v(l)$ は頂点 v における BSDF ローブ l についての粗さを返す。
つまり、パストレーサーによってベースパスを生成するとき、以下を満たす最小の $k(k \geq 2)$ を持つ頂点 $\boldsymbol{x_k}$ を頂点コピー戦略における再接続頂点として格納する。

$$min(\|\boldsymbol{x_k} - \boldsymbol{x_{k-1}}\|) \geq d_{min} \wedge min(\alpha_{x_{k-1}}(l_{k-1}), \alpha_{x_k}(l_k)) \geq \alpha_{min} \tag{3.1.15}$$

そして、RIS によってオフセットパスを生成するとき、頂点 y_{k-1} まで乱数コピー戦略を行い、y_{k-1} と x_k を頂点コピーにより接続する。
このとき、y_{k-1} と x_k について、

$$min(\|\boldsymbol{x_k} - \boldsymbol{y_{k-1}}\|) \geq d_{min} \wedge min(\alpha_{y_{k-1}}(l`_{k-1}), \alpha_{x_k}(l_k)) \geq \alpha_{min} \tag{3.1.16}$$

を満たす必要がある。
ReSTIR PT はこのハイブリッドシフト戦略によるパス再サンプリングのために、パスの定義を拡張してサンプリングを行う。
パスサンプルは頂点と BSDF ローブ $\boldsymbol{l}$、光源サンプリングテクニックのタグ $t \in \mathcal{T}$ を持ち、

$$\bar{\boldsymbol{x}} = [x_0, (x_1, \boldsymbol{l_1}), (\boldsymbol{x_2}, \boldsymbol{l_2}), (\boldsymbol{x_3}, \boldsymbol{l_3}), ..., (\boldsymbol{x_{D-1}}, \boldsymbol{l_{D-1}}), (\boldsymbol{x^t_D})] \tag{3.1.17}$$

のように表す。
この拡張パスに対する積分は可能な全ての BSDF ローブ $\boldsymbol{l}$ の集合を $\mathcal{L}$ とし、また拡張パスからローブだけを取り出したパスを $\bar{l} = [\boldsymbol{l_1}, \boldsymbol{l_2}, ...]$ とすると、

$$\begin{aligned} I &= \sum_{D=1}^{\infty} \int_{\Omega_D} \bar{f}(\bar{\boldsymbol{x}}) d\bar{\boldsymbol{x}} \\ &= \sum_{D=1}^{\infty} \sum_{\bar{l} \in \mathcal{L}} \int_{\mathcal{A}^D} m_t(\boldsymbol{x}) f_{\bar{l}}(\boldsymbol{x}) d\boldsymbol{x} \end{aligned} \tag{3.1.18}$$

となる。ここで m_t はサンプリングテクニック t に対する MIS 重みである。

3.1.4 ReSTIR PT アルゴリズムによるパス再サンプリングのパストレーシングへの応用

この章では、GRIS を用いた任意パスの再サンプリングをパストレーシングに応用した、ReSTIR PT アルゴリズム [4, 2022] について述べる。

パスサンプルの生成

NEE パストレーシングは、パス打ち切り時のパス長さを D_{end} とすると、最大で $D_{end} - 1$ 本のパスを一回のパストレーシング・ループの中で生成する。
このパスツリーの全てのパスを RIS の入力とし、初期候補サンプル $\boldsymbol{x} = [X_1, X_2, ..., X_{D_{end}-1}]$ とする。
このとき、各パス $X_i = [x_0, x_1, \boldsymbol{x_2}, \boldsymbol{x_3}, ..., \boldsymbol{x_i}]$ について、そのモンテカルロ推定量は

$$
\begin{aligned}
\langle I \rangle &= \frac{f(X_i)}{p(X_i)} = \\
&= \Big(\prod_{j=1}^{i} \frac{f_s(\boldsymbol{x_{j+1}} \to \boldsymbol{x_j} \to \boldsymbol{x_{j-1}}) G(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j}) V(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j})}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}}) P_j} \Big) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}})
\end{aligned}
\tag{3.1.19}
$$

また、パスツリー x の推定量は

$$
\begin{aligned}
\langle I \rangle &= \frac{f(\boldsymbol{x})}{p(\boldsymbol{x})} = \\
&= \sum_{D=2}^{D_{end}} \Big(\prod_{j=1}^{D-1} \frac{f_s(\boldsymbol{x_{j+1}} \to \boldsymbol{x_j} \to \boldsymbol{x_{j-1}}) G(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j}) V(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j})}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}}) P_j} \Big) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}})
\end{aligned}
\tag{3.1.20}
$$

となる。

生成した候補サンプル $\boldsymbol{x} = [X_1, X_2, ..., X_{D_{end}-1}]$ から RIS を用いて一つのサンプルを選択することを考える。
各サンプル毎にリサンプリング重み w_i と、選択サンプル X_s に対する不偏寄与重み W_s を定式化しなければならないが、各サンプル X_i はドメイン $\mathcal{A}^{\rangle+\epsilon}$ に属しているので GRIS を使用した再サンプリングを行う。ここで、ターゲットドメイン $\mathcal{A}$ と候補サンプルのドメイン $\mathcal{A}^{i+2}$ について、$\mathcal{A}^{i+2} \in \mathcal{A}$ であるからシフトマッピング関数は入力をそのまま返す $T(X) = X$ となり、そのヤコビアンは $|J_T(X)| = 1$ となる。

また、再サンプリング MIS 重みについて、式 2.4.5、式 2.4.8、式 2.4.9 より、各サンプル X_i は部分空間 $\mathcal{A}^{i+2}$ に属し、全ての部分空間 $\mathcal{A}^k \in \mathcal{A}$ はパス長について

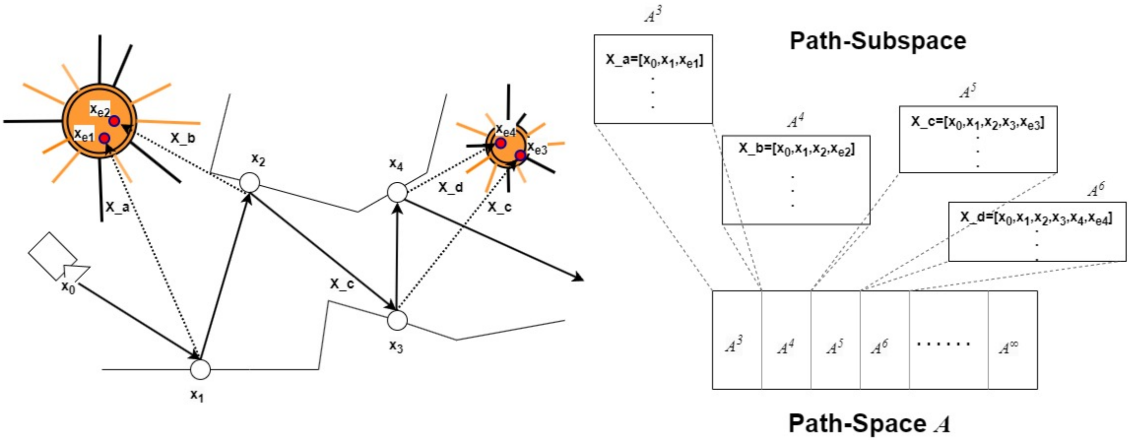

図 **3.7:** パスツリーとパス空間・部分パス空間の関係。

互いに独立であるから、その値は1となる。

$$m_i(X_i) = 1 \tag{3.1.21}$$

再サンプリング重みについて、目標関数 $\hat{p}$ を $\prod(f_s \cdot G \cdot V)L_e$ とすると、サンプル毎の不偏寄与重み W_i は単にサンプル毎の確率密度関数の評価値 $\frac{1}{p(X_i)P_j}$ を使えばよく、その値は

$$w_i = \Big(\prod_{j=1}^{D-1} \frac{f_s(\boldsymbol{x_{j+1}} \to \boldsymbol{x_j} \to \boldsymbol{x_{j-1}})G(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j})V(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j})}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}})P_j}\Big)L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}) \tag{3.1.22}$$

となる。

パスツリーから選択されたサンプル X_s に対する不偏寄与重み W_s は、式 TODO:より

$$W_s = \frac{1}{\hat{p}(X_s)} \sum_{i=1}^{D_{end}-1} w_i \tag{3.1.23}$$

となる。

NEE サンプリングと BSDF サンプリングの MIS による結合

NEE パストレーシングによるパスツリーの構築時、BSDF サンプリングによって発射したレイが光源に到達した場合、そのパスツリーには同じパス長を持つ異なるパス X_i^{NEE} と X_i^{BSDF} が存在することになる。

この2つのパスは同じドメイン $\mathcal{A}^{i+2}$ に属しているので、再サンプリング MIS 重みの適切な設計が必要になる。式 3.1.4 のようにバランスヒューリスティック

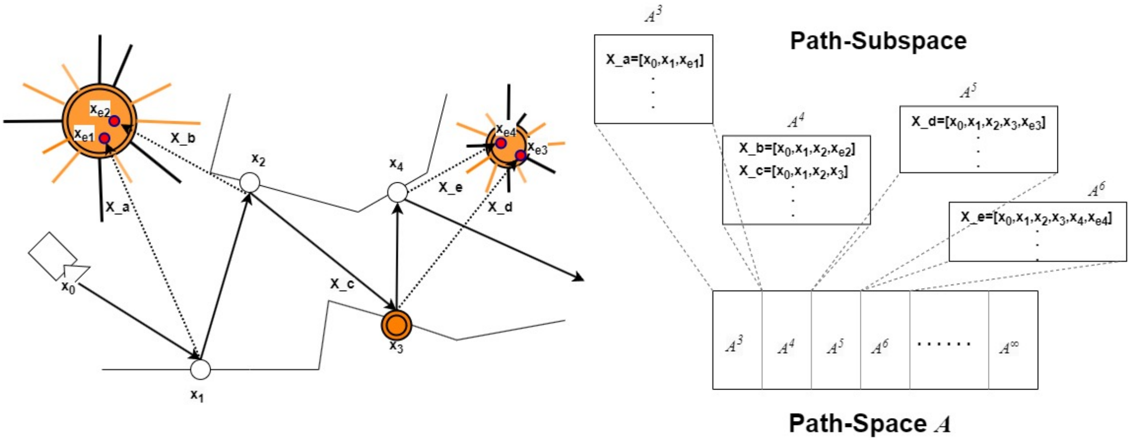

図 **3.8:** BSDF サンプリングによって光源に到達した場合のパスツリー。

スを用いればその重みは

$$
\begin{aligned}
m_i(X_i^{NEE}) &= \frac{p(X_i^{NEE})}{p(X_i^{NEE}) + p(X_i^{BSDF})} \\
m_i(X_i^{BSDF}) &= \frac{p(X_i^{BSDF})}{p(X_i^{NEE}) + p(X_i^{BSDF})}
\end{aligned}
\tag{3.1.24}
$$

となる。ここで $p(X_i) = \prod_{j=1}^{i+2} \frac{1}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}})}$ である。

時空間再サンプリング

初期候補サンプリングからのサンプル X_s を時空間再サンプリングする。
このとき、初期候補サンプルのドメイン Ω_x から現在ピクセルのドメイン Ω_y にシフトマッピング関数 T を用いてサンプルの変換をする必要がある。
時空間再サンプリングにおける再サンプリング重み w_{st} について、

$$
\begin{aligned}
w_{st} &= m_i(T(X_s))\hat{p}(T(X_s))|\frac{\nabla T}{\nabla X_s}|W_{x_s} \\
&= m_i(T(X_s))(\prod_{j=1}^{D} f_s \cdot G \cdot V\left|\frac{\nabla T}{\nabla x_j}\right|)W_{X_s} \\
&= m_i(T(X_s))(\prod_{j=1}^{D} f_s \cdot G \cdot V\left|\frac{\nabla T(\omega_j)}{\nabla \omega_j}\right|)W_{X_s}
\end{aligned}
\tag{3.1.25}
$$

GRIS ではシフトマッピング関数 T とサンプル x に対するヤコビアン $\left|\frac{\nabla T(x)}{\nabla x}\right|$ を再サンプリング重みの評価に必要とした。パスサンプルとヤコビアンについて

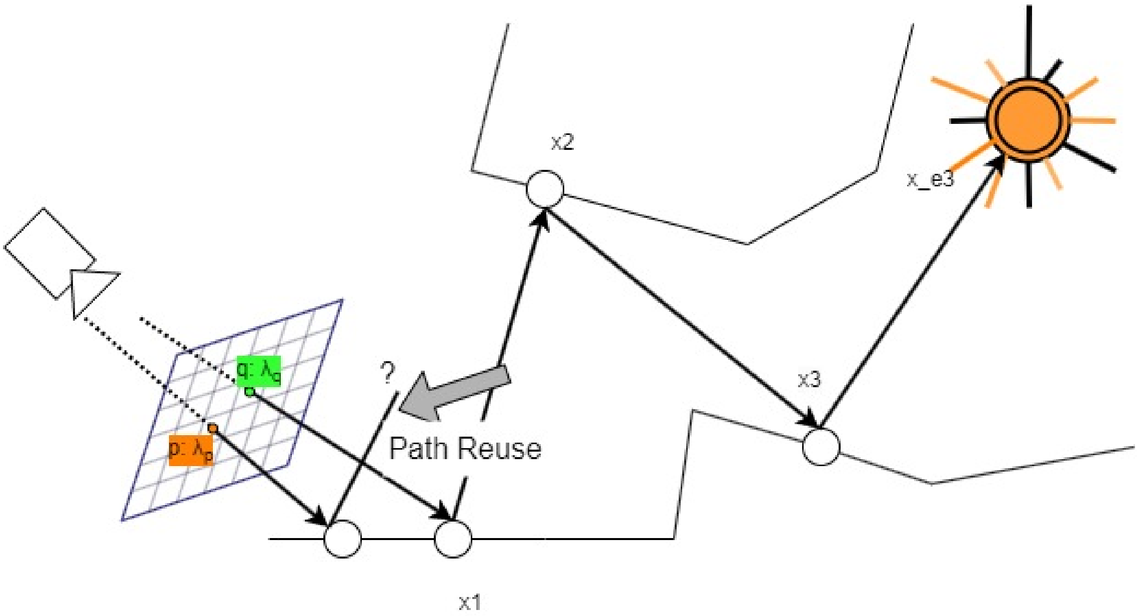

図 **3.9:** 異なるドメインからのパス再利用

の定式化に、シフトマッピング戦略と同様に勾配領域パストレーシング[2015]の定式化を用いることができる。
Kettunen ら [2015] はベースパス x、x について異なる測度への再パラメーター化を行ったパスを $\hat{x}$、オフセットパス $y = T(x)$ についても同様に $\hat{y}$ としたとき、そのヤコビアンを $|T'|$ を書くことにすると、

$$
\begin{aligned}
|T'| &= \left|\frac{\nabla y}{\nabla x}\right| \\
&= \left|\frac{\nabla y}{\nabla \hat{y}}\right|\left|\frac{\nabla \hat{y}}{\nabla \hat{x}}\right|\left|\frac{\nabla \hat{x}}{\nabla x}\right|
\end{aligned}
\tag{3.1.26}
$$

と、3 つの項に分解できることを示した。
このとき頂点コピー戦略における再接続では、パス

$$
\bar{\boldsymbol{x}} = [x_0, (x_1, \boldsymbol{l_1}), (\boldsymbol{x_2}, \boldsymbol{l_2}), ..., (\boldsymbol{x_{k-1}}, \boldsymbol{l_{k-1}}), (\boldsymbol{x_k}, \boldsymbol{l_k}), ..., (\boldsymbol{x_{D-1}}, \boldsymbol{l_{D-1}}), (\boldsymbol{x^t_D})]
$$

をパス

$$
\bar{\boldsymbol{y}} = [y_0, (y_1, \boldsymbol{l^y_1}), (\boldsymbol{y_2}, \boldsymbol{l^y_2}), ..., (\boldsymbol{y_{k-1}}, \boldsymbol{l^y_{k-1}}), (\boldsymbol{x_k}, \boldsymbol{l_k}), ..., (\boldsymbol{x_{D-1}}, \boldsymbol{l_{D-1}}), (\boldsymbol{x^t_D})]
$$

に、現在ピクセルについての立体角測度に再パラメーター化する。
微小面積と微小立体角の関係は式 2.3.2 で示した通りである。このとき頂点コ

ピーシフト関数 $T_{reconnection}$ におけるヤコビアンは、

$$
\begin{aligned}
|T_{reconnection}`| &= \left|\frac{\nabla\omega_{k-1}^{y}}{\nabla\omega_{k-1}^{x}}\right| \\
&= \frac{|cos(\phi_k^y)|}{|cos(\phi_k^x)|}\frac{\|x_k - x_{k-1}\|^2}{\|x_K - y_{k-1}\|^2}
\end{aligned}
\tag{3.1.27}
$$

となる。ここで、ϕ_k^y と ϕ_k^x はベクトル $x^k - y^{k-1}$ と $x^k - x^{k-1}$ が x_k の法線 n_k とのなす角である。

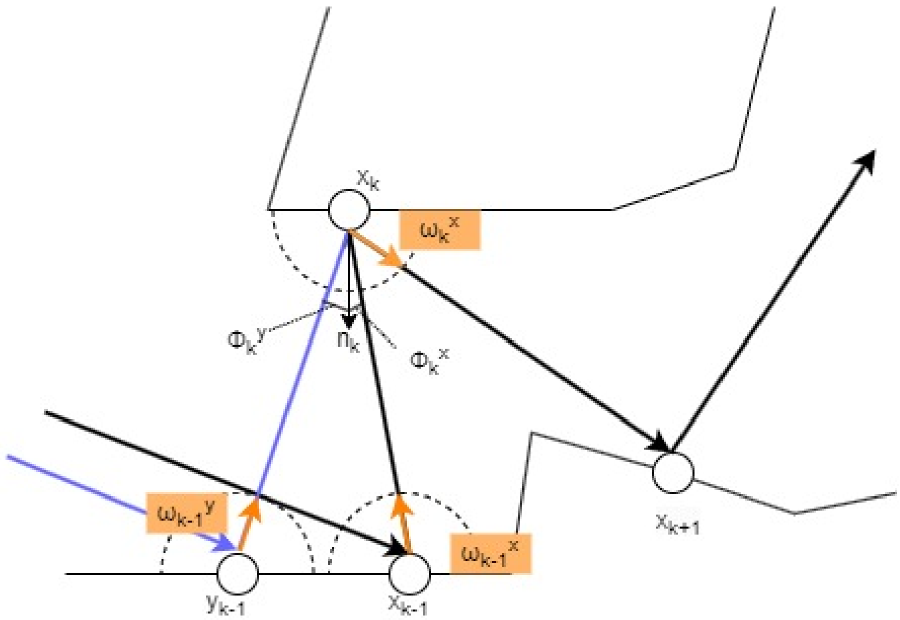

図 **3.10:** 頂点コピー戦略におけるヤコビアンの幾何的な意味。サンプル点は立体角におけるBSDFサンプリングで生成され、その確率密度はピクセル毎に異なるという事実を考慮したものである。

Lin ら [2022] は、シフトマッピングを Primary Sample Space [31, 2002] で行うことの有用性を示した。

PSS では、パスの積分値は以下のように表せる。

$$
\begin{aligned}
I &= \sum_{D=1}^{\infty}\int_{\mathcal{U}_D} F(\bar{\boldsymbol{u}})d\bar{\boldsymbol{u}} \\
with \quad F(\bar{\boldsymbol{u}}) &= \frac{\bar{f}(\mathcal{X}(\bar{\boldsymbol{u}}))}{p(\mathcal{X}(\bar{\boldsymbol{u}}))}
\end{aligned}
\tag{3.1.28}
$$

ここで、$\mathcal{U}_D$ は D 次元の単位超立方体で、$\bar{\boldsymbol{u}}$ はその中の点であり、これは長さ $D+1$ のパスを生成するのに適した乱数の列である。

また、$\mathcal{X}(\bar{\boldsymbol{u}})$ は拡張パスサンプル 3.1.17 を乱数から生成する関数であり、$\bar{f}$ はそのパスの寄与である。
p はそのパスを発生させる確率密度関数である。
ハイブリッドシフト戦略では乱数再利用戦略はこの PSS で行う。ハイブリッドシフト戦略は頂点接続を行う x_{k-1} と x_k で乱数の変更を行うため、そのヤコビアンは

$$\begin{aligned}\left|\frac{\nabla\bar{\boldsymbol{u}}^y}{\nabla\bar{\boldsymbol{u}}^x}\right| &= \left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{1}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{1}}}\right|\cdots\left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{k-1}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{k-1}}}\right|\left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{k}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{k}}}\right|\cdots\left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{D-1}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{D-1}}}\right| \\ &= \left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{k-1}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{k-1}}}\right|\left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{k}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{k}}}\right|\end{aligned} \tag{3.1.29}$$

となる。ここで、$\left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{D}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{D}}}\right|$ は、点 x_D がパスの終端であるため、その項は削除されることに注意する。
このとき Lin らは

$$\begin{aligned}\left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{k-1}}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{k-1}}}\right| &= \left|\frac{\nabla\bar{\boldsymbol{u}}^y_{\boldsymbol{k-1}}}{\nabla\omega^y_{k-1}}\right|\left|\frac{\nabla\omega^y_{k-1}}{\nabla\omega^x_{k-1}}\right|\left|\frac{\nabla\omega^x_{k-1}}{\nabla\bar{\boldsymbol{u}}^x_{\boldsymbol{k-1}}}\right| \\ &= \frac{p_{(\omega,l)^y_{k-1}}(\boldsymbol{y_k})}{p_{(\omega,l)^x_{k-1}}(\boldsymbol{x_k})}\left|\frac{\nabla\omega^y_{k-1}}{\nabla\omega^x_{k-1}}\right|\end{aligned} \tag{3.1.30}$$

となることを示した。
ここで、$p_{(\omega,l)^x_{k-1}}(\boldsymbol{x_k})$ は

$$p_{(\omega,l)^x_{k-1}}(\boldsymbol{x_k}) = p_{\omega^x_{k-1}}(\boldsymbol{x}_k)\cdot p_{l^x_{k-1}}(\boldsymbol{x}_k) \tag{3.1.31}$$

で表される確率密度関数の積であり、パス $\boldsymbol{x}$ 上の BSDF サンプリングローブ l と方向 ω の、立体角測度における確率密度関数を意味する。
注意点として、$\omega^x_k = \omega^y_k$ かつ $l^x_{k-1} = l^y_{k-1}$ であったとしてもセンサー頂点が異なるため、その確率密度関数は異なる。

GRIS における ReSTIR PT のためには、パスサンプルが保持すべきデータが複雑になる。Wyman ら [7, 2023] は ReSTIR PT のためのリザーバー実装として、以下の設計 5 を提案している。

ここで J はヤコビアン計算のためのキャッシュであり、その値は

$$J = p_{BSDF}(\boldsymbol{x_{k-1}}, w_{\overrightarrow{k-1,k}})\cdot\frac{|cos(\phi^x_k)|}{\|\boldsymbol{x_k}-\boldsymbol{x_{k-1}}\|^2}\cdot p_{BSDF}(\boldsymbol{x_k}, w_{\overrightarrow{k,k+1}}) \tag{3.1.32}$$

とする。

```
struct {
    struct {
        float3 dir // Incident Direction
        float3 radiance // Incident Radiance
        int id_triangle // Triangle ID on current vertex
        float2 barycentrics // Barycentric Coordinates on current vertex
        BSDFLobe l_prev // BSDF Lobe on previous vertex
        BSDFLobe l_current // BSDF Lobe on current vertex
    } RcVertex
    RcVertex rcv // Reconnection-able Vertex
    float rand_prev // Random seed for BSDF Sampling on previous vertex
    float rand_current // Random seed for BSDF Sampling on current vertex
    int current // Current Path vertex index K
    float J // Cache for jacobian
} Sample
class {
    Sample y
    float UCW
    float w_sum
    float c
    Function _init_:
        y = ∅
        UCW = 0.0f
        w_sum = 0.0f
        c = 0.0f
    Function Update(Sample T(s_i), Float w_i, Float T(p̂_i) , Float rand01):
        w_sum ← w_sum + w_i
        if rand01 < wi / w_sum then
            sample ← T(s_i)
            p̂ ← T(p̂_i)
        end
        M++
        return true
} Reservoir
```

Algorithm 5: ReSTIR PT における WRS の Reservoir

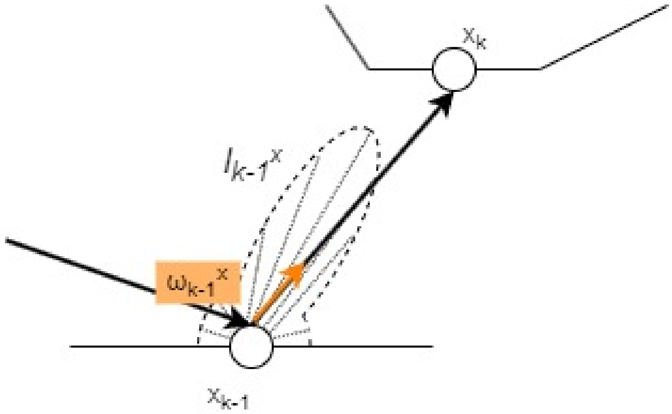

図 **3.11:** 点 x_{k-1} における BSDF ローブと x_k に向けてサンプリングされた方向 ω^x_{k-1}。$p_{(\omega, D^x_{k-1})}(\boldsymbol{x_k})$ はこの状況における重点的サンプリングの確率である。

このリザーバーを使用し、時空間方向に GRIS によるフルパス再サンプリングを行った結果は以下のようになる。

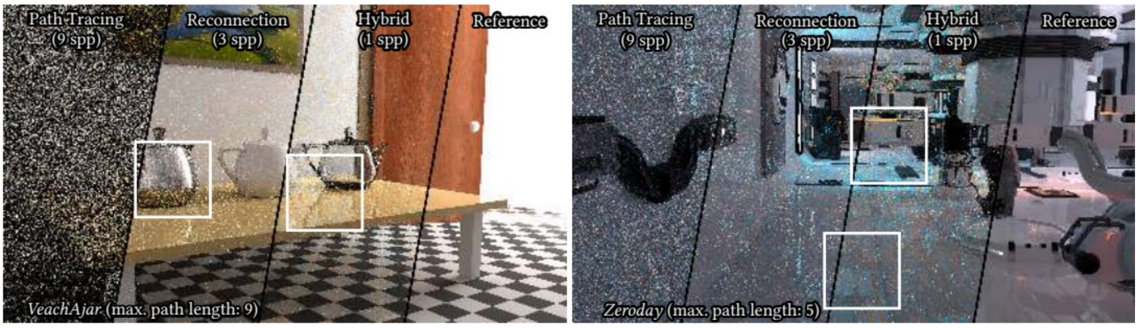

図 **3.12:** Veach Door と Zero Day における ReSTIR PT の結果 ([4] より引用)。左より、単方向パストレーシング、頂点コピー戦略のみの再サンプリング、ハイブリッドシフト戦略による再サンプリング、リファレンス画像。

3.2 スペクトラルレンダリングのサンプリング効率化

ここでは、本研究の基礎となっているスペクトラルレンダリングのサンプリング効率化に関する研究について述べる。

3.2.1 代表波長によるスペクトラルサンプリング

スペクトラルレンダリングが可能なモンテカルロ積分を利用したパストレーサーの最も単純な形式は、サンプリングされたパスごとに一つの波長を使用することであるが、パスにスペクトル依存性が無い場合、そのパスは全ての波長で有効であるため非常に無駄が多い。
スペクトルの次元を分離することで、より効率よくモンテカルロ積分を行うことができる。

$$\langle I \rangle = \frac{1}{N}\frac{1}{C}\sum_{i=1}^{N}\sum_{j=1}^{C}\frac{f(\boldsymbol{x_i}|\lambda_i^j)}{p(\boldsymbol{x_i}|\lambda_i^j)} \tag{3.2.1}$$

ここで、$\boldsymbol{x_i} \in \mathcal{A}$ はサンプリングされたパスであり、$\lambda \in \Lambda$ はサンプリング波長一個、f は寄与放射輝度推定関数、p は、パスと波長のペア $\boldsymbol{x_i}|\lambda_i^j$ をサンプリングする確率である。
N はパス次元におけるモンテカルロ積分のサンプル数を意味し、C はスペクトル次元におけるモンテカルロ積分のサンプル数を意味する。
Wilkie ら [8, 2014] は、パスの”代表波長”と”従属波長”を用いた多波長推定関数を提案し、これは Hero Wavelength Spectral Sampling(HWSS) と呼ばれる。
HWSS では、パスの経路構築には代表波長を用い、パスの寄与評価では従属波長を含む複数の波長を用いる。
従属波長はランダムサンプリングせずに、可視光領域 Λ を均等にカバーする

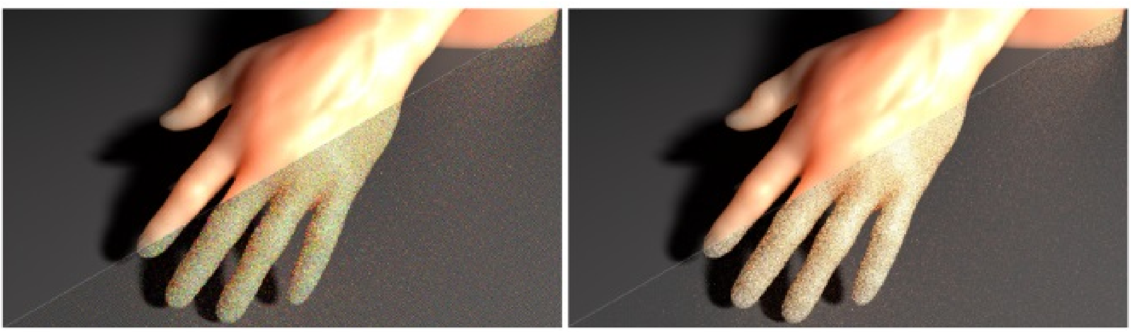

図 **3.13:** HWSS の効果 ([8] より引用)。Subsurface Scattering を引き起こす肌のシーンにて、左下が単波長によるレンダリング、右下が HWSS によるレンダリング。画像上側は 1024 サンプルによるリファレンス画像。

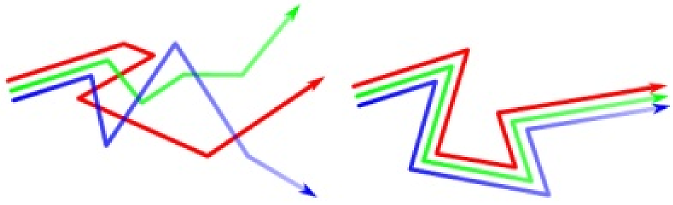

図 **3.14:** HWSS によるスペクトルパスのイメージ。[3] より引用。

ような回転関数 r_j を用いて生成する。

$$\begin{aligned}\lambda^j &= r_j(\lambda^0) \\ &= (\lambda^0 - \lambda_{min} + \frac{j}{C}(\lambda_{max} - \lambda_{min}))\boldsymbol{mod}(\lambda_{max} - \lambda_{min}) + \lambda_{min}\end{aligned} \tag{3.2.2}$$

ここで、λ^0 は代表波長、λ^j は従属波長、j は従属波長のインデックス、C は従属波長と代表波長の合計数、$\lambda_{min}, \lambda_{max}$ は可視光領域の最小波長、最大波長である。

これの具体的な例として、パス経路構築では代表波長による BSDF サンプリン

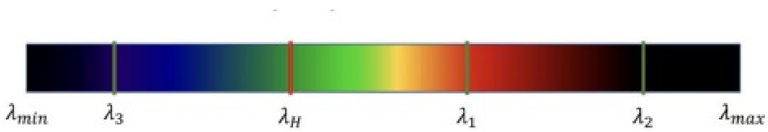

図 **3.15:** HWSS の回転関数の図示。代表波長と従属波長が等間隔となるように決定される。

グを行い、モンテカルロ積分によるパスの寄与評価では、”相乗り”している従属波長を含めたすべての波長のセットについて計算をするということである。

結合サンプル密度

あるパス $\boldsymbol{x} = [x_1, x_2, ..., x_D]$ について、複数の波長 $\boldsymbol{\lambda} = [\lambda^1, \lambda^2, ..., \lambda^C]$ を用いて寄与を計算する場合、パス上の各頂点 $x_i, (1 \leq i \leq D)$ を複数の波長で再利用する。x_i を生成したサンプリングが波長依存であるとき、それぞれの波長についてのサンプルペア $[x_i; \lambda_1^j], [x_i; \lambda_2^j], ..., [x_i; \lambda_C^i]$ に対して異なる確率になる。x_i は $\boldsymbol{\lambda}$ について固定されているため、$[x_i; \lambda_i^j], (1 \leq i \leq C)$ をサンプリングする確率は

$$p(x_i, \lambda_i^j) = \frac{\sum_{k=1}^{C} p(\lambda_i^k) p(x_i|\lambda_i^k)}{C} \tag{3.2.3}$$

となる。ここで、$p(\lambda)$ は波長 λ をサンプリングする確率、$p(x_i|\lambda)$ は波長 λ における頂点 x_i をサンプリングする確率である。
頂点 x_i においてスペクトル依存性が無い場合、この確率は次のように書ける。

$$p(x_i, \lambda_i^j) = p(\lambda_i^j)p(x_i|\lambda_i^j) \tag{3.2.4}$$

波長の MIS 重み

HWSS において、代表波長と従属波長についての MIS 重みを考える。この重みは複数のパスサンプリング技術を結合する際に必要となる。
あるサンプリング技術 s があるとき、HWSS による C 個の波長についての MIS 重みは次のように書ける。

$$\begin{aligned} w_s(x, \lambda) &:= \frac{p(x|r_s^{-1}(\lambda))p(r_s^{-1}(\lambda))}{\sum_{k=1}^{C} p(x|r_k^{-1}(\lambda))p(r_k^{-1}(\lambda))} \\ &= \frac{p(x|\lambda_i^0)p(\lambda_i^0)}{\sum_{k=1}^{C} p(x|\lambda_i^k)p(\lambda_i^k)} \end{aligned} \tag{3.2.5}$$

ここで、$r_s^{-1}(\lambda)$ は、サンプリングテクニック s において使用する λ を発生させた元の代表波長 $\lambda^{0,s}$ を返す。
また、モンテカルロ積分推定量は

$$\begin{aligned} \langle I \rangle &= \frac{1}{N}\sum_{i=1}^{N}\sum_{s=1}^{C} w_s(x_i^s, \lambda_i^s)\frac{f(x_i^s, \lambda_i^s)}{p(x_i^s|r_s^{-1}(\lambda_i^s))p(r_s^{-1}(\lambda_i^s))} \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{s=1}^{C} w_s(x_i^s, \lambda_i^s)\frac{f(x_i^s, \lambda_i^s)}{p(x_i^s|\lambda_i^0)p(\lambda_i^0)} \end{aligned} \tag{3.2.6}$$

となる。
式 3.2.5、 3.2.6 より、複数のサンプリング技術における HWSS による MIS 重みは次のように書ける。

$$\begin{aligned} \langle I \rangle &= \frac{1}{N}\sum_{i=1}^{N}\sum_{s=1}^{C} \frac{p(x_i|\lambda_i^0)p(\lambda_i^0)}{\sum_{k=1}^{C} p(x_i|\lambda_i^k)p(\lambda_i^k)}\frac{f(x_i, \lambda_i^s)}{p(x_i|\lambda_i^0)p(\lambda_i^0)} \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} \frac{f(x_i, \lambda_i^j)}{\sum_{k=1}^{C} p(x_i|\lambda_i^k)p(\lambda_i^k)} \end{aligned} \tag{3.2.7}$$

複数のサンプリング手法の結合

式 3.2.5 を用いることで、NEE サンプリングや BSDF サンプリングなどのサンプリング手法を結合することができる。
ある技術 t によるパスサンプリング確率を p_t とし、パス構築に用いる t の集合を $\mathcal{T}$ とする。このとき、技術 t_t における MIS 重みは

$$w_{t_t}(x_i, \lambda_i^{t_t}) := \frac{p_{t_t}(x_i|\lambda_i^0)p(\lambda_i^0)}{\sum_{t_u \in \mathcal{T}} \sum_{k=1}^{C} p_{t_u}(x_i|\lambda_i^k)p(\lambda_i^k)} \tag{3.2.8}$$

となる。

3.2.2 スペクトラルレンダリングにおける再サンプリング

HWSS を用いたスペクトラルパストレーシングにおいて、ReSTIR DI アルゴリズムを適用し、間接照明寄与を与えるパスと不偏な結合を行うことを考えた、*Wētā Digital* による成果 [9, 2022] をここで紹介する。なお、この研究は進行中のものであり、適切な実用シーンで十分な検証が行われていないプロトタイプであることに注意されたい。
我々の知る限り、スペクトルレンダリングにおける再サンプリングの適用については、この研究が初めてである。

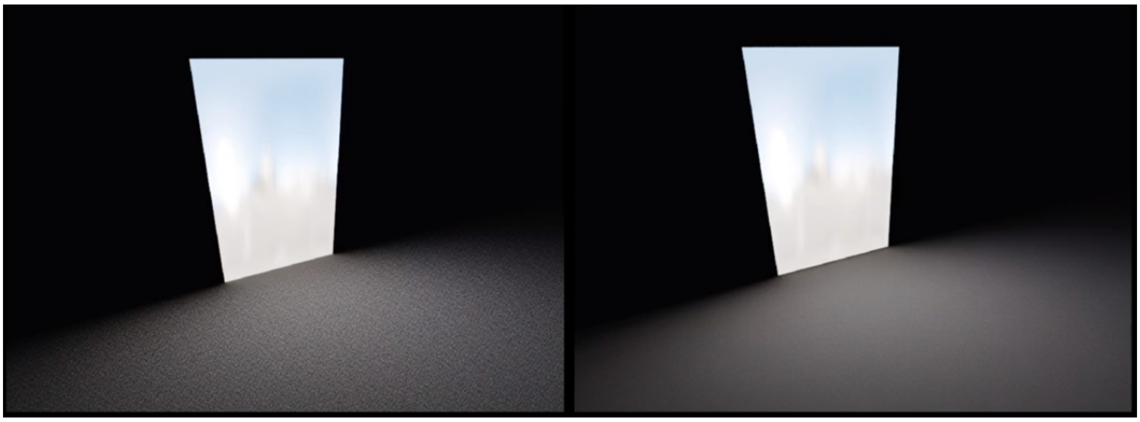

図 3.16: 左:スペクトラルパストレーシング、右:再サンプリングを結合したスペクトラルパストレーシング。[3] より引用。

スペクトラルレンダリングにおける直接照明 **NEE** 再サンプリング

ReSTIR DI と同じように、直接照明に寄与する NEE サンプルを再サンプリングの対象とする。

候補サンプルの生成: M 個の初期サンプル $\boldsymbol{x_e} = [x_{e1}, x_{e_2}, ..., x_{e_M}]$ を確率密度関数 p に従って生成する。初期サンプルの生成過程では HWSS による代表波長を $\lambda^0 \in \Lambda$ として固定する。すなわち、一つのパスにつき一つの代表波長を選択し、代表波長は可視光ドメイン Λ よりサンプリングされる。
このとき、各候補サンプル x_{e_i} に対応するリサンプリング重みは

$$
\begin{aligned}
w_i &= \frac{1}{M}\frac{\hat{p}(x_{e_i} \to x_1|\lambda^0)}{p(x_{e_i})} \\
with \quad \hat{p}(x_{e_i} \to x_1|\lambda^0) &= ToRGB(G(x_{e_i} \leftrightarrow x_1)L_e(x_{e_i} \to x_1|\lambda^0)
\end{aligned}
\tag{3.2.9}
$$

とする。
スペクトル領域への応用の検証を目的とするため、BSDF 項 f_s を目標関数 $\hat{p}$ に含めない。また、目標関数 $\hat{p}$ は再利用するサンプルが含む任意の代表波長 $\lambda_0 \in \Lambda$ のドメイン Λ 全域に対して評価出来る必要がある。

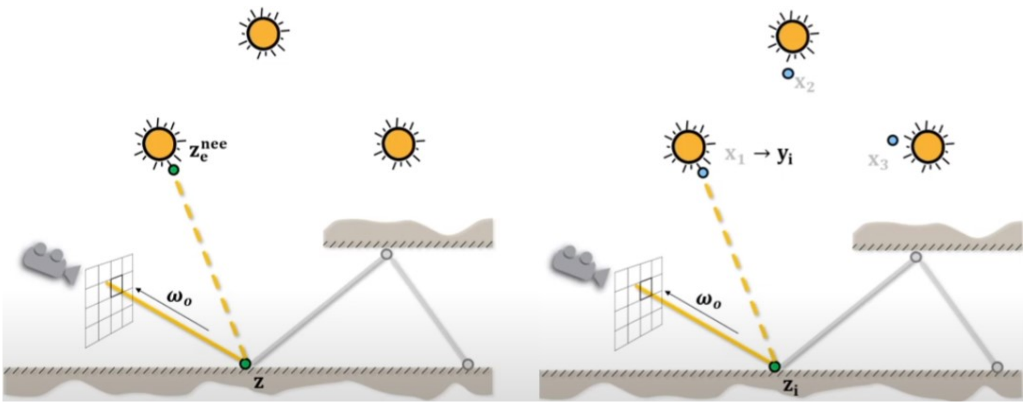

図 3.17: 初期サンプルの生成 ([9] より引用)。パスの構築については ReSTIR アルゴリズムと同等であるが、その再サンプリング重みは波長サンプリングを考慮しなければならない。

時間再利用: 時間方向再サンプリングにおいて、過去フレームからのサンプルを GRIS を用いたドメイン変換を行うことで、過去フレームのサンプルを現在フレームのサンプルに変換することができる。
照明計算が進行している現在のパスが保持する代表波長を λ_i^0、一次衝突頂点を

$x_{1,i}$とし、対応する過去フレームパスが保持する代表波長をλ_j^0、一次衝突頂点を$x_{1,j}$とする。このとき、現在フレームからの新鮮なサンプル$x_{e,i}$に対応するリサンプリング重みは

$$w_i = m_i(x_{e,i})\hat{p}(x_{e,i} \to x_{1,i}|\lambda_i^0)W_i \tag{3.2.10}$$

$$\text{with} \quad m_i(x_{e,i}) = \frac{\hat{p}(x_{e,i} \to x_{1,i}|\lambda_i^0)}{\hat{p}(x_{e,i} \to x_{1,i}|\lambda_i^0) + \hat{p}(x_{e,i} \to x_{1,j}|\lambda_j^0)\left|\frac{\nabla T_i^{-1}}{\nabla x_{1,i}|\lambda_i^0}\right|},$$

$$W_i = \frac{1}{\hat{p}(x_{e,i} \to x_{1,i}|\lambda_i^0)} \sum_{k=1}^{M} w_k$$

となる。新鮮なサンプルが他のサンプリング領域$x_{1,j}|\lambda_j^0$で評価できることに注意する。彼らの実験ではヤコビアンの値は$\left|J_{i\to j}\right| = 1$となったが、これはヒーロー波長の変化によるパスの確率密度の変化が無いためと推察する。
過去フレームから再利用するサンプル$x_{e,j}$に対応するリサンプリング重みは

$$w_j = m_j(x_{e,j})\hat{p}(x_{e,j} \to T_i(x_{1,j}|\lambda_j^0))W_j\left|\frac{\nabla T_i}{\nabla x_{1,j}|\lambda_j^0}\right| \tag{3.2.11}$$

$$\text{with} \quad m_j(x_{e,j}) = \frac{\hat{p}(x_{e,j} \to x_{1,j}|\lambda_j^0)}{\hat{p}(x_{e,j} \to x_{1,j}|\lambda_j^0) + \hat{p}(x_{e,i} \to T_i(x_{1,j}|\lambda_j^0))\left|\frac{\nabla T_i}{\nabla x_{1,j}|\lambda_j^0}\right|},$$

$$W_i = \frac{1}{\hat{p}(x_{e,j} \to x_{1,j}|\lambda_j^0)} \sum_{k=1}^{M} w_k$$

となる。

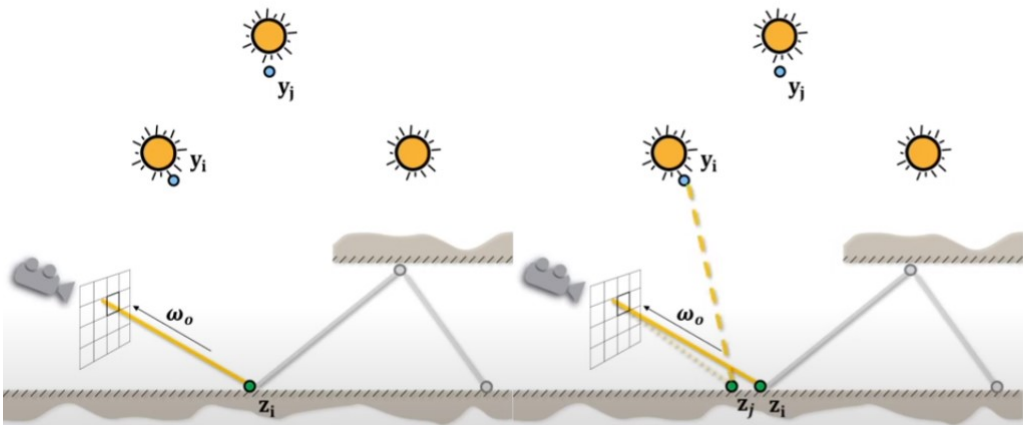

図 **3.18:** 異なるピクセルからのサンプルを表した図([9] より引用)。

空間再利用: 時間方向再サンプリングと同様に、現在フレームの近傍ピクセルからサンプルの再サンプリングを行う。彼らのプロトタイプでは、時間方向再サンプリング後に、近傍から3つのサンプルを現在のリザーバーに入力した。この際、サンプルされたサンプリングドメインに関する全ての情報、すなわち代表波長・法線・位置・時間などを保持することで空間再サンプリング後のリザーバーを次の進行に利用できるとした。

直接照明 NEE 再サンプリング寄与度の重み付け

再サンプリングされたサンプル x_e^{NEE} を使用し、最終的な NEE による放射輝度の寄与度 fW を評価することを考える。ここで、$f = f_s \cdot G \cdot Le$ であり、W は x_e^{NEE} の UCW である。
注意すべき点として、f は代表波長 λ^0 に依存し、それを評価した値は代表波長と従属波長の個数 C の大きさをもつベクトル $\langle I \rangle \in \mathbb{R}^C$ である。
3.2.5 より、あるパス $\boldsymbol{x} = [x_0, x_1]$ から NEE サンプリングしたパス $\boldsymbol{x^{NEE}} = [x_0, x_1, x_e^{NEE}]$ における HWSS の波長毎 MIS 重みは

$$w_{NEE}(\boldsymbol{x^{NEE}}, \lambda^k) = \frac{p(\boldsymbol{x}|\lambda^0)p(x_e^{NEE}|\lambda^0)p(\lambda^0)}{\sum_{k=1}^{C} p(\boldsymbol{x}|\lambda^k)p(x_e^{NEE}|\lambda^k)p(\lambda^k)} \tag{3.2.12}$$

となるが、$p(x_e^{NEE}|\lambda^k)p(\lambda^k)$ は、リサンプリングされた頂点 x_e^{NEE} に対する確率であり、その確率密度 p は難解であるから計算し評価することができない。
しかしその不偏寄与重み $W_{x_e^{NEE}}^{\lambda^k}$ は評価できるためこの p を近似することができ、

$$\begin{aligned} p(x_e^{NEE}|\lambda^k)p(\lambda^k) = p_{NEE}(x_1 \to x_e^{NEE}, \lambda^k) &\simeq \frac{1}{W_{x_e^{NEE}}^{\lambda^0} \frac{\hat{p}(x_e^{NEE} \to x_1|\lambda^0)}{\hat{p}(x_e^{NEE} \to x_1|\lambda^k)}} \\ &\simeq \frac{\hat{p}(x_e^{NEE} \to x_1|\lambda^k)}{W_{x_e^{NEE}}^{\lambda^0} \hat{p}(x_e^{NEE} \to x_1|\lambda^0)} \end{aligned} \tag{3.2.13}$$

$$. \qquad \simeq \frac{\hat{p}(x_e^{NEE} \to x_1|\lambda^k)}{\sum_{q=1}^{M} w_q} \tag{3.2.14}$$

となる。この近似値は UCW の性質から、偏りのない p の代替として機能する。
式 3.2.14 の分母は RIS 進行ごとに更新され、その値は単に RIS 重みの総和である。目標関数 $\hat{p} = 0$ となるような波長 $\lambda^{invalid}$ について、その場合 x_e^{NEE} はサンプリングできない。

直接照明 **NEE** 再サンプリング寄与と単方向パスからの間接照明寄与の結合

式 3.2.8 より、BSDF サンプリングによるサンプル x_e^{BSDF} と再サンプリングされた NEE サンプル x_e^{NEE} における MIS 重みは

$$
\begin{aligned}
w_t(\boldsymbol{x^t}, \lambda_t^i) &:= \frac{p_t(\boldsymbol{x^t}|\lambda_t^0)p(\lambda_t^0)}{\sum_{j\in[BSDF,NEE]}\sum_{k=1}^{C} p_j(\boldsymbol{x^t}|\lambda_t^k)p(\lambda_t^k)} \\
with \quad & p_{NEE}(\boldsymbol{X}|\lambda) = p_{BSDF}(x_0, \vec{w_{01}}, \lambda)p_{NEE}(x_1 \to x_e, \lambda) \\
& p_{BSDF}(\boldsymbol{X}|\lambda) = p_{BSDF}(x_0, \vec{w_{01}}, \lambda)p_{BSDF}(x_1, \vec{w_{12}}, \lambda)
\end{aligned}
\tag{3.2.15}
$$

となる。ここで、t は BSDF サンプリングと NEE サンプリングを選択するインデックスである。
また、$p_{NEE}(x_1 \to x_e, \lambda) = p(x_e^{NEE}|\lambda)$ であることに注意すると、3.2.15 の p_{NEE} は、式 3.2.14 で代替することができる。
報告によると、リザーバーに 0 でないサンプルが少ない RIS 進行初期の段階では、式 3.2.15 の NEE ウェイトは 0 に近い値をとるため、推定量の分散が大きくなる可能性がある。

第4章

提案手法

4.1　スペクトラルパス再サンプリング

4.2　初期候補サンプルの生成

この節では再サンプリングの初期候補となる、波長データを含むパスを生成について述べる。

4.2.1　パスツリーの構築

RGB ReSTIR PTと同様に、NEEパストレーシングを用いてパスツリーの構築を行う。また、一回のNEEパストレーシングにおいて、代表波長$\lambda^0 = \lambda \in \Lambda$として固定する。
このとき、パス空間$\mathcal{A}$をパス長ごとに分割したパス部分空間$\mathcal{A}^{\mathcal{D}-\infty}, \mathcal{A} = \cup_{D=2}^{\infty} A^{D-1}$についての積分だからその推定量は、

$$\begin{aligned}\langle I \rangle &= \langle \int_{\mathcal{A}} f(x) dx \rangle \\ &= \sum_{D=2}^{\infty} \Big(\prod_{j=1}^{D-1} \frac{f_s(\boldsymbol{x_{j+1}} \to \boldsymbol{x_j} \to \boldsymbol{x_{j-1}}, \lambda) G(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j}) V(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j})}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}}, \lambda) P_j} \Big) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}}, \lambda)\end{aligned} \tag{4.2.1}$$

となり、最大で$D_{end} - 1$本のパスツリーを生成する。
このパスツリーの全てのパスをRISの入力とし、初期候補サンプル$\boldsymbol{x} = [X_1, X_2, ..., X_{D_{end}-1}]$とする。
また、それぞれのサンプルは式3.1.17を単に代表波長について拡張した以下の定式化を用いて定義される。

$$\bar{\boldsymbol{x}} = [\lambda, x_0, (x_1, \boldsymbol{l_1}), (\boldsymbol{x_2}, \boldsymbol{l_2}), (\boldsymbol{x_3}, \boldsymbol{l_3}), ..., (\boldsymbol{x_{D-1}}, \boldsymbol{l_{D-1}}), (\boldsymbol{x^t_D})] \tag{4.2.2}$$

4.2.2　再サンプリング重みの定式化

サンプル毎の再サンプリング重み、不偏寄与重みGRISの定式化を波長λについて拡張する。
一回の初期候補サンプル生成で波長がλに固定されていることを考えると、その再サンプリングMIS重みは式3.1.24をそのまま使うことができる。

$$\begin{aligned} m_i(X_i^{NEE}) &= \frac{p(X_i^{NEE})}{p(X_i^{NEE}) + p(X_i^{BSDF})} \\ m_i(X_i^{BSDF}) &= \frac{p(X_i^{BSDF})}{p(X_i^{NEE}) + p(X_i^{BSDF})} \end{aligned} \tag{4.2.3}$$

ここで$p(X_i) = \prod_{j=1}^{i+2} \frac{1}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}}, \lambda) P_j}$となる。
このとき各サンプル毎の再サンプリング重みは式3.1.22より、代表波長$\lambda = \lambda^0$についてパスの構築を行っていることに注意して

$$w_i = \Big(\prod_{j=1}^{D-1} \frac{f_s(\boldsymbol{x_{j+1}} \to \boldsymbol{x_j} \to \boldsymbol{x_{j-1}}, \lambda^0) G(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j}) V(\boldsymbol{x_{j+1}} \leftrightarrow \boldsymbol{x_j})}{p(\boldsymbol{x_j} \to \boldsymbol{x_{j+1}}, \lambda^0) P_j}\Big) L_e(\boldsymbol{x_D} \to \boldsymbol{x_{D-1}} | \lambda^0) \tag{4.2.4}$$

となる。また、代表波長λ^0に対する不偏寄与重みは式3.1.23より、

$$W_s^{\lambda^0} = \frac{1}{\hat{p}(X_s | \lambda^0)} \sum_{i=1}^{D_{end}-1} w_i \tag{4.2.5}$$

となる。

4.2.3　初期候補サンプル生成におけるスペクトル的側面

NEEパストレーシング時に波長を固定することで、パス空間についての積分のみの問題となることから、初期候補サンプルのプロセスはRGBレンダリングにおけるパス再サンプリングのプロセスと完全に同等になる。しかし、HWSSのような多波長サンプリングによるパスの構築を行った場合、パスの確率は波長毎に異なることから、サンプルされたパスの不偏寄与重みが波長に依存することに注意する必要がある。
これは半透明物体の分光現象などがわかりやすい例である。

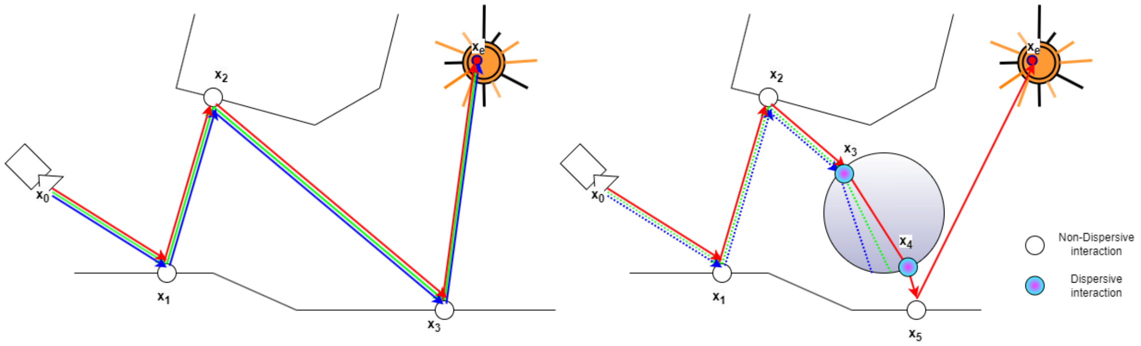

図 4.1: 分光現象によるパス確率の変化の例。左図では、センサー・光源間に波長依存の頂点が存在しないが、右では屈折物体が存在するため青と緑の従属波長はその確率が0になる。

4.3 時空間再サンプリング

4.3.1 シフトマッピング戦略

RGB ReSTIR PT [4] では頂点コピーによる再接続戦略と乱数再利用による再生戦略のハイブリッドシフト戦略によって時空間のパス再サンプリングを行った。これをスペクトラルパストレーシングに応用するには、パス構築に関わる波長依存の光学現象を考慮してシフトマッピング戦略を再設計する必要がある。
そのようなシフトマッピングとそのヤコビアンについて Petitjean らによるスペクトル勾配領域パストレーシング [10] において定式化が行われている。
Petitijean らは屈折による分光を引き起こすパス頂点についてハーフベクトルコピー戦略を適用してパスの再構築を行い、スペクトル非依存な頂点にてパスの再接続を行う戦略を提案している。

我々はこのハーフベクトルコピー戦略 [10] と Lin らによるハイブリッドシフト戦略 [4] を組み合わせた拡張ハイブリッドシフト戦略を提案する。
拡張ハイブリッドシフト戦略では、屈折による分光を引き起こす頂点について乱数再生戦略を適用し波長非依存の頂点で再接続を行う。それ以外の頂点については Lin らのハイブリッドシフト戦略を適用する。すなわち、パスの各頂点ごとに、距離条件と粗さ条件を確認し、有効であれば頂点コピーを行う。そうでなければ乱数コピーを行い、頂点コピーを遅延する。そして、頂点コピー後は光源にたどり着くまで分光条件を確認する。分光が発生すれば乱数コピーを行い、粗さ条件と分光条件によって次の頂点コピーを行うかを決定する。これをパスが光源にたどり着くまで繰り返す。
各条件について、式 4.2.2 の拡張パス

$$\bar{\boldsymbol{x}} = [\lambda, x_0, (x_1, \boldsymbol{l_1}), (\boldsymbol{x_2}, \boldsymbol{l_2}), (\boldsymbol{x_3}, \boldsymbol{l_3}), ..., (\boldsymbol{x_{D-1}}, \boldsymbol{l_{D-1}}), (\boldsymbol{x^t_D})] \tag{4.3.1}$$

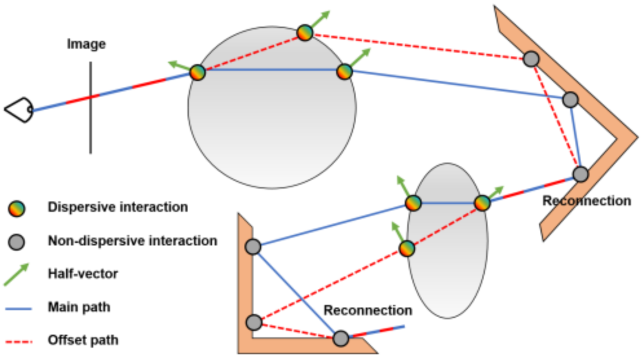

図 4.2: ハーフベクトルコピー戦略によるスペクトルシフトの概念。図は [10] より引用。

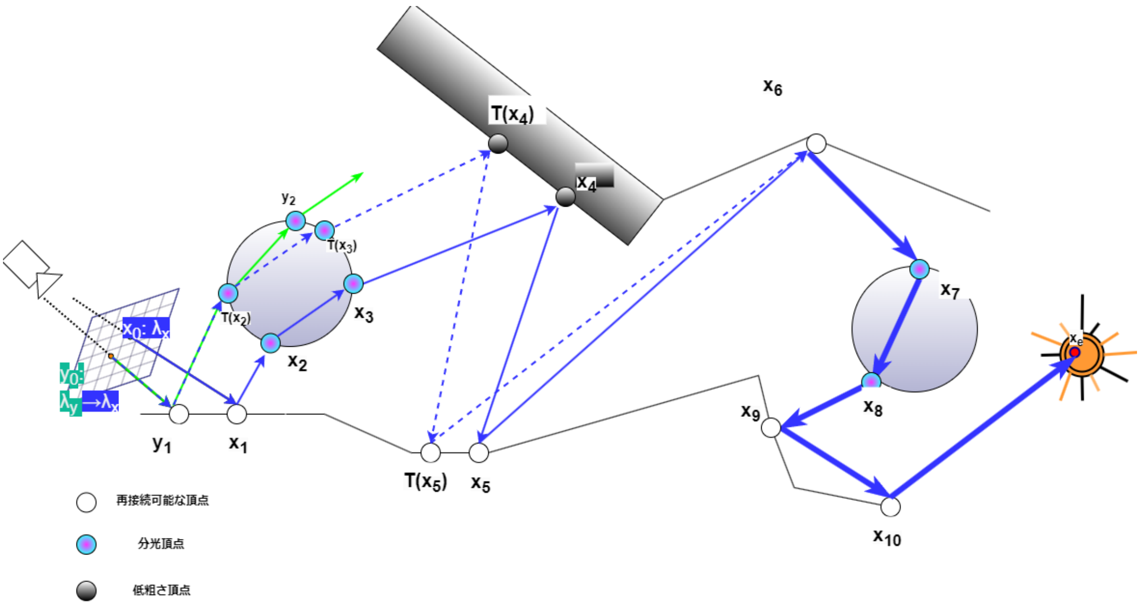

図 4.3: 拡張ハイブリッドシフト戦略の概念図。

において、ここで距離条件とは

$$min(\|\boldsymbol{x_k} - \boldsymbol{x_{k-1}}\|, \|\boldsymbol{x_k} - \boldsymbol{y_{k-1}}\|) \geq d_{min} \tag{4.3.2}$$

をみたすことである。
粗さ条件とは、

$$min(\alpha_{x_{k-1}}(l_{k-1}), \alpha_{y_{k-1}}(l'_{k-1}), \alpha_{x_k}(l_k)) \geq \alpha_{min} \tag{4.3.3}$$

を満たすことである。
また、分光条件とは

$$Dispersive(l_{k-1}) + Dispersive(l_k) + Dispersive(l'_{k-1}) = 0 \tag{4.3.4}$$

を満たすことである。ここで *Dispersive* は、屈折による分光を引き起こす頂点について1、それ以外の頂点について0を返す関数である。
$\boldsymbol{l}$ はBSDFローブ、$t \in \mathcal{T}$ は光源サンプリングテクニックのタグ、λ はパスごとに固定された波長である。

4.3.2 時空間パス再サンプリングにおける再サンプリング重みの定式化

あるピクセル p において、異なるピクセル q から波長 λ_q で再サンプリングされたパス X_q を用いて、波長 λ_p におけるパス X_p に対するパス再サンプリングを行うことを考える。
このとき再サンプリング重み w_{pq} は、式TODO:より

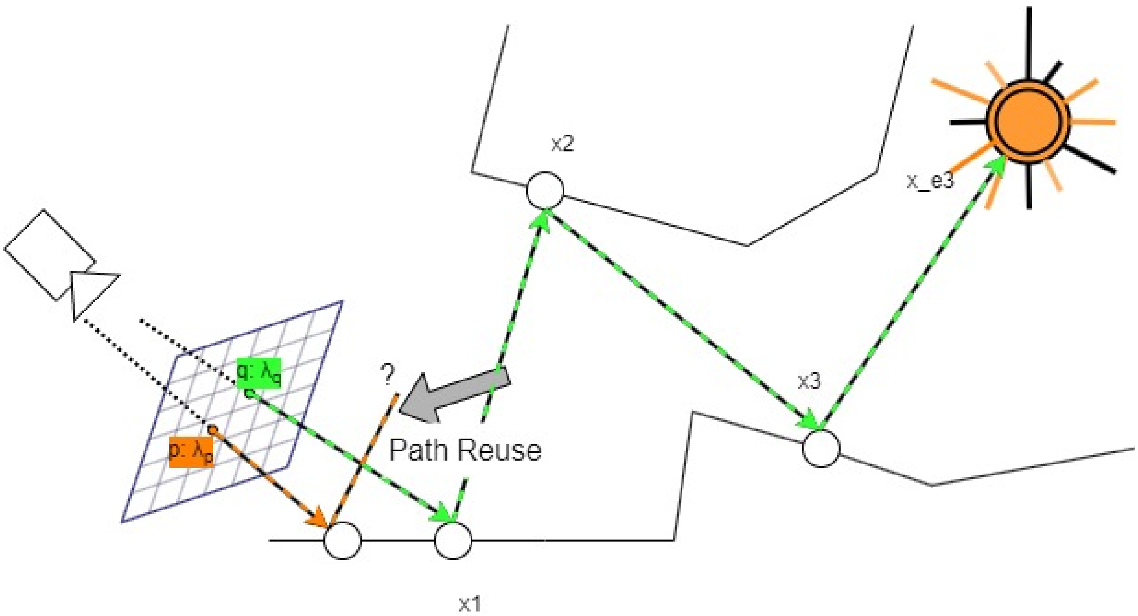

図 4.4: 時空間スペクトラルパス再サンプリングの簡単なイメージ図。積分の対象となるドメインがパス空間の側面、スペクトル的側面双方で異なることに注目されたい。

$$w_{pq} = m_p(T_p(X_q|\lambda_q))\hat{p}(T_p(X_q|\lambda_q))W_p\left|\frac{T_p}{X_q|\lambda_q}\right| \tag{4.3.5}$$

となる。ここで、T_p はパス $X_q|\lambda_q$ をパス $X_p|\lambda_p$ に変換するシフトマッピング関数である。すなわち $T(X_q|\lambda_q) = X_p|\lambda_p$。

また、リサンプリング重み m_p は Lin らが提案した式 2.4.5 に従い、再サンプリングされた各パスの UCW は初期候補サンプルのウェイトから求めることができる。

ヤコビアンの定式化

各シフトマップ戦略におけるヤコビアンは以下のように定式化される。

再接続戦略（頂点コピー）: 頂点コピーは波長に依存しない頂点で行われるので、[27] や [4] と同様の定式化を行うことができる。
式 3.1.27 より、

$$\begin{aligned} |T_{reconnection}`| &= \left|\frac{\nabla\omega^{y}_{k-1}}{\nabla\omega^{x}_{k-1}}\right| \\ &= \frac{|cos(\phi^{y}_{k})|}{|cos(\phi^{x}_{k})|}\frac{\|x_k - x_{k-1}\|^2}{\|x_K - y_{k-1}\|^2} \end{aligned} \tag{4.3.6}$$

となる。
乱数再生戦略(乱数コピー): 乱数コピーは分光条件を満たす頂点でも行われるがそのヤコビアンは 1 となり、Lin らの式 3.1.30 と同様に頂点再接続時のヤコビアンをサンプリング確率の比で表すことができる。

$$\begin{aligned} \left|\frac{\nabla\bar{\boldsymbol{u}}^{\boldsymbol{y}}_{\boldsymbol{k-1}}}{\nabla\bar{\boldsymbol{u}}^{\boldsymbol{x}}_{\boldsymbol{k-1}}}\right| &= \left|\frac{\nabla\bar{\boldsymbol{u}}^{\boldsymbol{y}}_{\boldsymbol{k-1}}}{\nabla\omega^{y}_{k-1}}\right|\left|\frac{\nabla\omega^{y}_{k-1}}{\nabla\omega^{x}_{k-1}}\right|\left|\frac{\nabla\omega^{x}_{k-1}}{\nabla\bar{\boldsymbol{u}}^{\boldsymbol{x}}_{\boldsymbol{k-1}}}\right| \\ &= \frac{p_{(\omega,l)^{y}_{k-1}}(\boldsymbol{y_k})}{p_{(\omega,l)^{x}_{k-1}}(\boldsymbol{x_k})}\left|\frac{\nabla\omega^{y}_{k-1}}{\nabla\omega^{x}_{k-1}}\right| \end{aligned} \tag{4.3.7}$$

ここで、$p_{(\omega,l)^{x}_{k-1}}(\boldsymbol{x_k})$ は

$$p_{(\omega,l)^{x}_{k-1}}(\boldsymbol{x_k}) = p_{\omega^{x}_{k-1}}(\boldsymbol{x}_k)\cdot p_{l^{x}_{k-1}}(\boldsymbol{x}_k) \tag{4.3.8}$$

で表される確率密度関数の積であり、パス $\boldsymbol{x}$ 上の BSDF サンプリングローブ l と方向 ω の、立体角測度における確率密度関数を意味する。

最終的なヤコビアンの値は、

$$|T`| = \frac{p_{(\omega,l)^{y}_{k-1}}(\boldsymbol{y_k})}{p_{(\omega,l)^{x}_{k-1}}(\boldsymbol{x_k})}\frac{|cos(\phi^{y}_{k})|}{|cos(\phi^{x}_{k})|}\frac{\|x_k - x_{k-1}\|^2}{\|x_K - y_{k-1}\|^2}\frac{p_{(\omega,l)^{y}_{k}}(\boldsymbol{y_{k+1}})}{p_{(\omega,l)^{x}_{k}}(\boldsymbol{x_{k+1}})} \tag{4.3.9}$$

となり、これを頂点再接続時のシフトマッピング関数の補正値として使用する。

第5章

評価

5.1　評価手法

評価に使用したハードウェアを以下に示す。

表 **5.1:** 評価に使用したハードウェア

GPU	CPU	Memory	OS	GAPI
AMD Radeon 7900 XTX	AMD Ryzen9 3900X	64GB	Windows11 23H2	DirectX12

また、実験はNVIDIA Falcor フレームワーク [32] をスペクトラルレンダリングのために拡張したもの [33] を使用した。

実験は一パスごとに単一の波長を割り当て、標準的なNEE パストレーサーおよび再サンプリングパストレーサーを用いて1サンプル レンダリングの結果とGT のRMSE、実行時間を比較する。

5.2　評価結果

図5.1 に評価結果を示す。左から標準パストレーサー、再サンプリングパストレーサー、GT である。サンプル数の平方に比例してノイズが減少することを考慮すると、1サンプルあたりおおよそ1.9倍から4.1倍の性能向上が見られ、また複雑な経路をとるシーンほどそのノイズ削減効果が高いことがわかる。

5.3 考察

1パス1波長の再サンプリングパストレーサーによるスペクトラルレンダリングの実装と評価を行った。
再サンプリングによって分散低減の効果が確認されたが、多波長サンプリングでは先行研究の定式化が機能しなくなる興味深い現象が確認された。HWSSによる多波長サンプリングを行った場合、直接照明のみの再サンプリングでは推定器の一致性が確認されたが、Reservoirの再サンプリング数が十分でない(1〜8程度)では図5.2に示すように顕著な色の偏りが確認された。これは*Wētā Digital*による、従属波長の波長MIS重みは代表波長の不偏寄与重みによって決定される確率変数となり、その分散が代表波長のMIS重みより激しいという報告[9, 2022]と一致する。
しかしながらフルパス空間において多波長サンプリングを行った場合、図5.3に示すように従属波長における積分推定値に誤りが生じることが確認された。これにより本来の結果よりも彩度が失われた結果になり、これはフルパス空間におけるパス再サンプリングでは、多波長サンプリング時の波長MIS重みは先行研究の定式化では誤った結果に収束することを意味する。

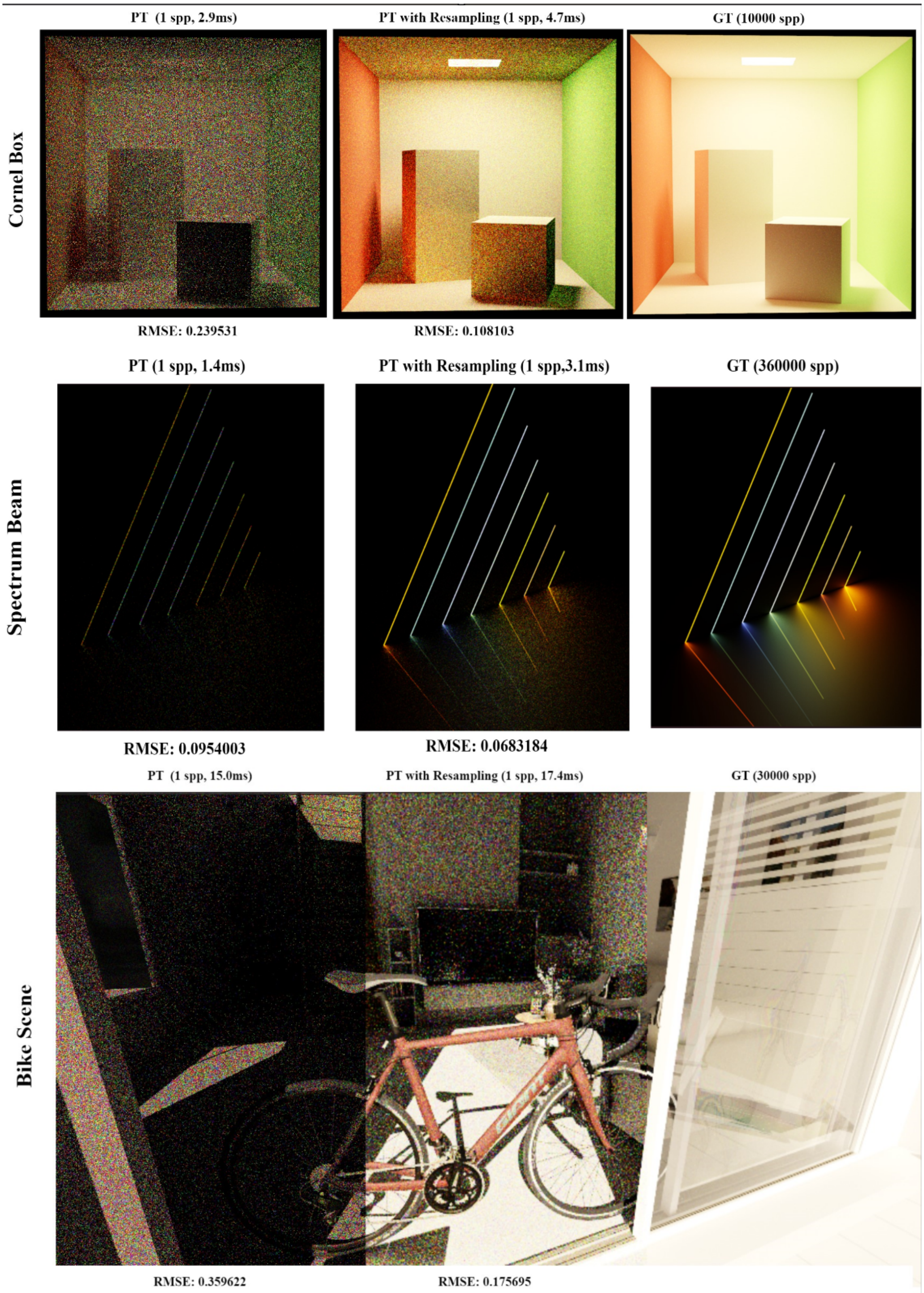

図 5.1: 評価結果。左から標準パストレーサー、再サンプリングパストレーサー、GT。

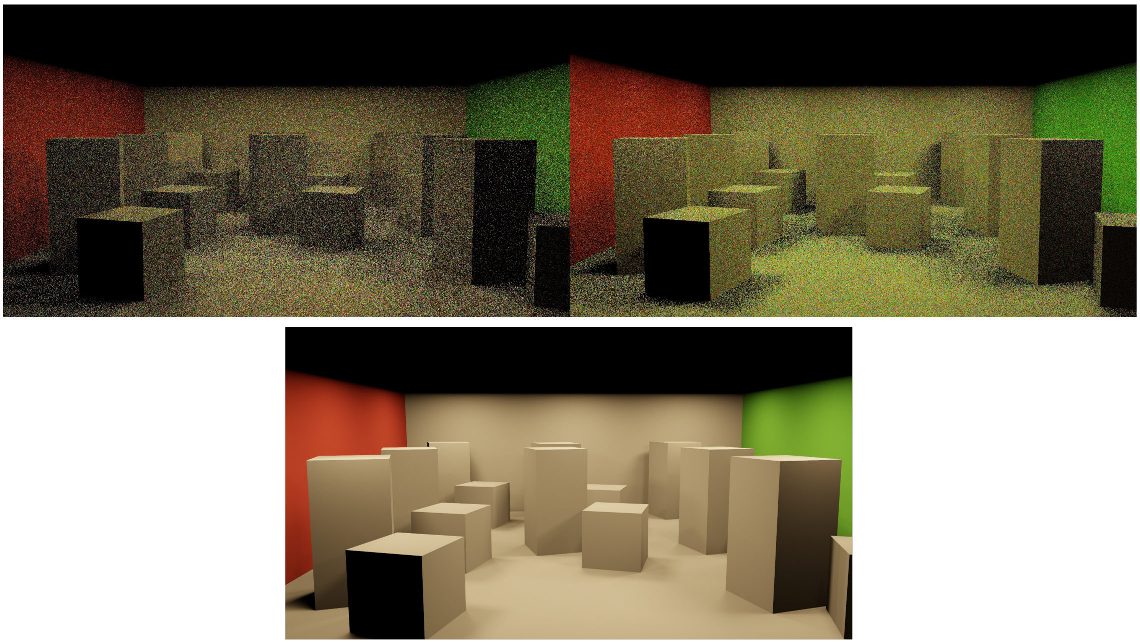

図 5.2: 多波長サンプリングにおける、低い再サンプリング数における色分散。左上が1sppの標準パストレーサーによる直接照明、右上が1spp、リザーバーサイズ8の再サンプリングパストレーサーによる直接照明、下がGT。

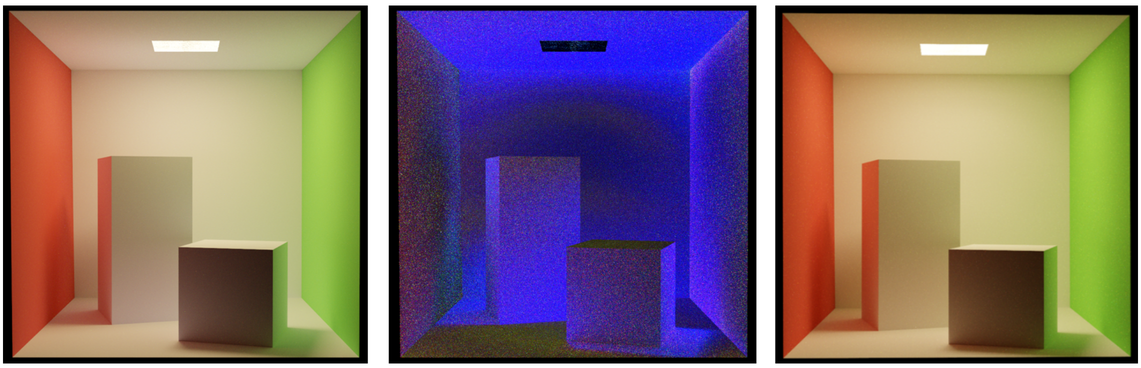

図 5.3: 多波長サンプリングにおける従属波長における積分推定値の誤り。左が再サンプリングパストレーサーによる結果、右が標準パストレーサー、中央が10倍差分の絶対値。

第6章

結論

リアルタイムの物理ベースレンダリングを目的として開発されたReSTIRアルゴリズムが、オフラインレンダリングの世界、特にスペクトルレンダリングにおいても有効であることを示した。
本手法はスペクトラルパストレーシングにおける積分推定の分散を有効に低減し、また複雑な実シーンでは特にその効果が発揮された。また、パストレーシングと比較してその処理時間は1.5倍から2倍ほどに収まっており、処理時間への影響も限定的である。
第5で議論したように、本手法はまだ多くの課題を含んでいる。最も重要なのは多波長サンプリング時における従属波長のMIS重みの不偏推定であり、これはHWSSのような多波長マルチサンプリングの強力な分散低減効果を得る上で重要な要素である。またほかの興味深い方向として、[34, 2023]のような再サンプル間の相関の低減や条件付き再サンプリング[35, 2023]は、再サンプリング時初期に現れる色分散を大幅に低減させる可能性がある。
スペクトラルレンダリングにおける勾配領域パストレーシングもRGBレンダリングと比較すると、まだ未知の領域であり多様体探索のような手法によって波長領域のよりよいシフトマッピング戦略が開発される可能性がある。
ReSTIRアルゴリズムは現在、リアルタイムパストレーシング実現のために非常に盛んに研究されており、そしてGRISによってその成果はオフラインレンダリングに十分に応用可能であることは本研究で示されたため、その領域の成果をさらに適用することで、より高速なスペクトルレンダリングが実現される可能性がある。

参考文献

[1] Andrew Stockman, Lindsay T Sharpe, et al. Colour and vision research laboratories, 2021. `http://www.cvrl.org`.

[2] Wenzel Jakob, Sbastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org.

[3] Andrea Weidlich, Alex Forsythe, Scott Dyer, Thomas Mansencal, Johannes Hanika, Alexander Wilkie, Luke Emrose, and Anders Langlands. Spectral imaging in production. In *ACM SIGGRAPH 2021 Courses*, August 2021.

[4] Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. Generalized resampled importance sampling: Foundations of restir. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 41.4*, July 2022.

[5] Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. Restir gi: Path resampling for real-time path tracing. In *Computer Graphics Forum (High Performance Graphics 2021)*, 2021.

[6] Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 39.4*, July 2020.

[7] Chris Wyman, Markus Kettunen, Daqi Lin, Benedikt Bitterli, Cem Yuksel, Wojciech Jarosz, Pawel Kozlowski, and Giovanni De Francesco. A gentle introduction to restir: Path reuse in real-time. In *ACM SIGGRAPH 2023 Courses*, August 2023.

[8] A. Wilkie, S. Nawaz, M. Droske, A. Weidlich, , and J. Hanika. Hero wavelength spectral sampling. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)33.4*, July 2014.

[9] Andrea Weidlich, Chloe LeGendre, Carlos Aliaga, Christophe Hery, Jean-Marie Aubry, JiÅ Vorba, Daniele Siragusano, and Richard Kirk. Practical aspects of spec-

tral data in digital content production. In *ACM SIGGRAPH 2022 Courses*, August 2022.

[10] Victor Petitjean, Pablo Bauszat, and Elmar Eisemann. Spectral gradient sampling for path tracing. In *Computer Graphics Forum (Proceedings of EGSR)*, 2018.

[11] CD PROJEKT RED. Cyberpunk 2077. `https://www.cyberpunk.net/`, 2023 (visited on 13/01/2024).

[12] NVIDIA. Cyberpunk 2077: Technology preview of new ray tracing overdrive mode out now. `https://www.nvidia.com/en-us/geforce/news/cyberpunk-2077-ray-tracing-overdrive-update-launches-april-11/`, 2023 (visited on 13/01/2024).

[13] Remedy Entertainment. Alan wake 2. `https://www.alanwake.com`, 2023 (visited on 13/01/2024).

[14] Marc Droske, Johannes Hanika, Jiri Vorba, Andrea Weidlich, and Manuele Sabbadin. Path tracing in production: The path of water. In *ACM SIGGRAPH 2023 Courses*, August 2023.

[15] 国際度量衡局 (BIPM). 国際単位系（SI）第 9 版（2019）日本語版, 2019.

[16] James T Kajiya. The rendering equation. In *ACM Transactions on Graphics (SIGGRAPH 1986)*, 1986.

[17] Paul S Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In *ACM Transactions on Graphics (SIGGRAPH 1990)*, 1990.

[18] E.Veach and L.Guibas. Bidirectional estimators for light transport. In *Eurographics Rendering Workshop*, 1994.

[19] H Jensen. Global illumination using photon maps. In *Eurographics Rendering Workshop*, 1996.

[20] T.Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping: A probabilistic approach. In *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*, 2008.

[21] E.Veach and L.Guibas. Metropolis light transport. In *ACM Transactions on Graphics (SIGGRAPH 1997)*, 1997.

[22] Shlomi Steinberg, Ravi Ramamoorthi, Benedikt Bitterli, Eugene d'Eon, Ling-Qi Yan, and Matt Pharr. A generalized ray formulation for wave-optics rendering. 2023.

[23] Justin F. Talbot, David Cline, and Parris Egbert. Importrace resampling for global illumination. In *Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering). Eurographics Association*, June 2005.

[24] M. T. Chao. A general purpose unequal probability sampling plan. In *BIOMETRIKA 69.3*, 1982.

[25] Jakub Boksansky, Paula Jukarainen, and Chris Wyman. Rendering many lights with grid-based reservoirs. In et al.'s Adam Marrs, editor, *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX*, 2021.

[26] Binh-Son Hua, Adrien Gruson, Victor Petitjean, Matthias Zwicker, Derek Nowrouzezahrai, Elmar Eisemann, and Toshiya Hachisuka. A survey of gradient-domain rendering. In *Computer Graphics Forum. Vol. 38. 2. Wiley Online Library*, 2019.

[27] Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frdo Durand, and Timo Aila. Gradient domain metropolis light transport. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, July 2013.

[28] Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. Specular manifold sampling for rendering high frequency caustics and glints. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, July 2020.

[29] Wenzel Jakob and Steve Marschner. Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)31.4*, July 2012.

[30] Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frdo Durand, and Matthias Zwicker. Gradient-domain path tracing. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, July 2015.

[31] C.Kelemen, L.Szlrmay-Kalos, A.Gyorgy, and F.Csonka. A simple and robust mutation strategy for the metropolis light transport algorithm. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)21.4*, July 2002.

[32] Simon Kallweit, Petrik Clarberg, Craig Kolb, Tom'aš Davidovič, Kai-Hwa Yao, Theresa Foley, Yong He, Lifan Wu, Lucy Chen, Tomas Akenine-Möller, Chris Wyman, Cyril Crassin, and Nir Benty. The Falcor rendering framework, 8 2022. `https://github.com/NVIDIAGameWorks/Falcor`.

[33] Shlomi Steinberg, Ravi Ramamoorthi, Benedikt Bitterli, Eugene d'Eon, Ling-Qi Yan, and Matt Pharr. Real-time physical light transport (plt) framework, 8 2022. `https://github.com/ssteinberg/PLTFalcor`.

[34] Rohan Sawhney, Daqi Lin, Markus Kettunen, Benedikt Bitterli, Ravi Ramamoorthi, Chris Wyman, and Matt Pharr. Decorrelating restir samplers via mcmc mutations. *ACM Transactions on Graphics (ToG)*, p. to appear, January 2024.

[35] Markus Kettunen, Daqi Lin, Ravi Ramamoorthi, Thomas Bashford-Rogers, and Chris Wyman. Conditional resampled importance sampling and restir. In *SIGGRAPH Asia (Conference Track)*, December 2023.