Vol.2019-CG-173 No.3
2019/3/5

生成的ライトフィールドを使ったリアルタイム大域照明

森重伸也[†1]

概要： モバイルゲームからオープンワールドのハイエンドゲームまでスケーラブルに対応できるリアルタイム大域照明の手法について取り組んでいるものを紹介する.本手法は,動的な光源の変化,空の時間変化,天候変化に対応できる手法である．先行事例は,Ubisoft のオープンワールドのゲーム The Division で使われた Volumetric Real-Time GI と NVIDIA[a)] の Light Fields がある.The Division は品質が良いがデータの事前生成に 5 時間以上かかっており，一方で NVIDIA の手法はデータサイズの問題がある．実現方法は，ゲームの Level (Scene)から Light Fields をリアルタイム生成して，そこから幾何情報(G-Buffer)と可視情報(Visibility)を画面空間でサンプルして照明と陰影をリアルタイム計算する．その際，Light Probes に格納する幾何情報と可視情報の生成時間とデータサイズの問題に直面する．生成時間の問題は，View Independent Rasterization ベースの Multi-View Parallel Rendering を使って描画 1 フレームでシーンの幾何と可視を Point Cloud として生成する．データサイズの問題は，機械学習の生成モデルでデータから複雑な形状の確率密度関数を推定できる Generative Adversarial Networks (GANs)を使って，Light Fields を Level もしくは Streaming 単位で圧縮しておき，ランタイム時に生成する．

キーワード：大域照明，ライトフィールド，リアルタイムレンダリング，機械学習，デジタルゲーム

Real-Time Global Illumination using Generative Light Fields

SHINYA MORISHIGE[†1]

Abstract:

Keywords: Global Illumination, Light Fields, Real-Time Rendering, Machine Learning, Digital Game

1. はじめに

近年のリアルタイムゲームは，オープンワールド型でリアルタイムに変化する，生き生きした世界で，プレイヤーが能動的に遊ぶスタイルが人気を博している[1]．プレイヤーが何度でもゲームに参加したくなる要素として，朝から夜までのリアルタイム時間変化，快晴，雨，そして雪などリアルタイムの動的な環境変化がもたらすゲームのリプレイ性と美しいグラフィクスがあげられている．それらはゲームプレイの写真や動画を Web に公開して話題性に寄与する.そのような世界を実現するための重要な要素として，大域照明があり，下記の要件を満たすことが必要である．

- リアルタイムの時間と天候変化での照明と陰影
- 空気感の実現（Volumetric GI）
- オブジェクトの照明と陰影が環境に馴染む
- ゲーム制作において，高速な照明の iteration
- コンパクトなデータサイズ．室内と屋外が混合する 1 ㎞四方から数十 km 四方の広さを取り扱えること（自動配置、Streaming 対応）
- 描画 60fps / 30fps を実現できる GPU コスト

提案手法は，要件を満たすために，定点で光の情報を集める Light Field ベースのリアルタイム大域照明を用いる．

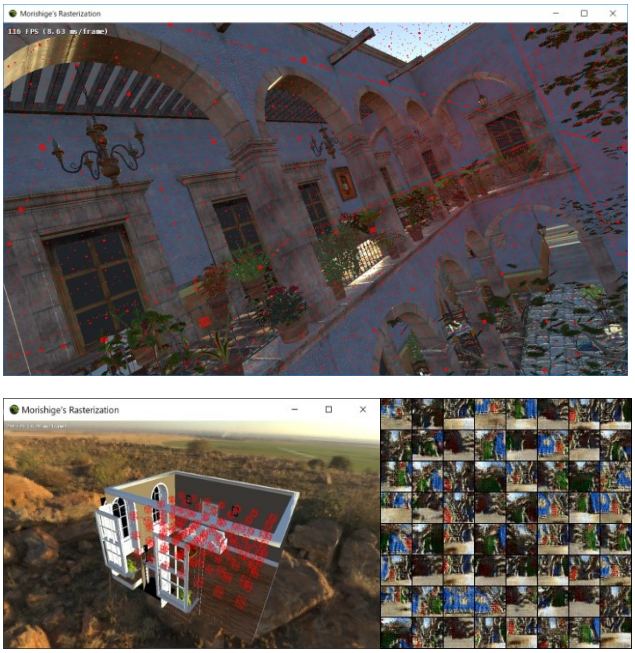

図 1 生成的ライトフィールドを使った大域照明(右下の画像は DCGAN で Light Field の Radiance を生成した結果)

2. リアルタイム大域照明の手法

ゲームでの大域照明の要素は，CG 分野における要素の部分集合である．本研究で対象とする大域照明の要素は下記 4 点とする．

†1 個人発表，株式会社 Cygames
Cygames, Inc.

a) NVIDIA は、米国および/または他国の NVIDIA Corporation の商標および/または登録商標です。

1. Irradiance (Indirect Diffuse)
2. Specular Radiance (Indirect Specular)
3. Ambient Occlusion (Large Obscure)
4. Soft Shadows (Contact Shadows)

ゲームにおけるオープンワールドの世界は，小さな空間 Level の集合から構成される．リアルタイム大域照明の手法は，Level 内の幾何形状に依存する手法と依存しない手法に分けられる．

表 1　リアルタイム大域照明手法の分類

環境変化	幾何依存	幾何非依存
Dynamic	Screen Space Reflection	Voxel Based GI
Static	Light Map Texture	Irradiance Volume

ゲームでは,プレイヤー視点からの近景,中景,遠景ごとに手法を使い分けて品質と処理負荷とデータサイズのバランスをとっている．近景は 100 m 程度まで，中景は 1km 程度まで，遠景はそれ以上の距離とする．

表 2　大域照明の要素と適用範囲の分類

要素	近景	中景	遠景
Indirect Diffuse	Irradiance Volume Light Map LPV VXGI Light Fields PRT SSDO PFF[*1]	Irradiance Volume Radiance Probe VXGI Light Fields PRT LPV PFF	Irradiance Volume Height Field GI Distant VPL[*2] PFF
Indirect Specular	SSR VXGI Light Fields Localized IBL	Localized IBL Reflection Probe VXGI	Distant Reflection Probe
Ambient Occlusion	SSAO	Visibility Probe PRT	Height Field AO Light Fields
Soft Shadows	LPV SSDO	Visibility Probe PRT LPV	Light Fields

*1:Precomputed Form Factor(PFF)，有限要素法ベースの手法

*2:Distant VPL，The Division，Assassin's Creed など UbiSoft のタイトルで使われた遠景向け間接照明の手法[3]

3. 先行事例と課題

先行事例と課題は次の通りである．

3.1 先行事例

(1) **Light Fields**

NVIDIA が提案したリアルタイム大域照明の手法である[2]．少数の Light Probes を一定間隔で配置して実現する．従来の大域照明では，Irradiance が変化する場所に Light Probe を自動配置する方法に着目して，Light Probe の数を削減しながら品質を維持していた．一方，Light Leak / Shadow Leak の問題が発生していた．これを Light Field では，Visibility Probe を生成して，間接照明のレンダリング時に，Shading Pixel 単位で，可視性を，分散シャドウマップのチェビシェフ不等式を用いて判定してリークを軽減している．この手法はビルなど建物など狭いレベルであれば利用できるが，1 つの Light Probe 辺りのデータ量が大きく，オープンワールド型ではデータサイズの問題に直面する．

(2) **Real-Time Volumetric GI**

2016 年に発売されたゲーム The Division で編み出された技法である[3]. ゲームのレベルで Surfel と呼ぶ幾何情報(albedo/normal)を定点でオフライン作成しておき，in-game 時にそれらを使って Lighting して Light Probe を生成する．近景は Light Probes から on the fly で, Irradiance Volume を生成することで, 高品質な Real-Time Volumetric Lighting を実現している．遠景はレベル全体を頭上からキャプチャした大きな 2D テクスチャを簡易 VPL とみなして Irradiance を計算する．Light Probe は，Ambient Cube でデータを格納する．PRT を使った事前計算で 5 時間以上かかる．制作時のデータ生成 Iteration に時間がかかる．6km^2 のレベル，マンハッタンでデータ量が 1.07GB 程度，Surfel 数が 56,442,867 であり，より広域の世界を作りたい場合にデータサイズが問題になる．

3.2 実用における課題

大域照明のデータ生成時間とデータサイズがある．

(1) データ生成時間

表 3 単位 frame で生成できるプローブ数の比較

方法	生成個数
Cube Map Rasterization	1 probes から数個程度
Voxelization	32x32x32 voxels 程度
Point Cloud+	1024x1024 probes 程度

Point Cloud は，VIR との組み合わせ

(2) データサイズ

1 つのプローブについて、6 面，128x128 texels のキューブマップを使い，Light Field (w, h, depth) = 1024 x 8 x 1024 probes の場合についてデータサイズを比較する．

表 4　1 つのプローブ・データサイズ(Bytes)の比較

表現	Indirect Specular	Indirect Diffuse
Raw(FP16)	589,824	24,576
SH	768	118
CNN1	19,488,768	0(Specular と共有)
CNN2	19,488,768	0(Specular と共有)
DCGAN	14,443,932	0(Specular と共有)

Raw は，Indirect Diffuse は，32x32 texels を使う．SH は，Indirect Specular は，L7SH で 64 基底を使う．Indirect Diffuse は，L2SH で 9 基底を使う．CNN1 は，低解像度のデータ 32x32 texels から復元する. CNN2 は, Light Field 空間の位置と時刻から画像 128x128 texels を復元する．CNN と DCGAN は，ネットワークの重みサイズも含む．DCGAN は，学習データの生成分布に fitting した一様分布 100 次元空間ベクトル$\mathbf{z} \in [-1,1]^{100}$で 1 つのプローブを表現する．その値をプローブ位置や時刻に対応づける．

表 5　データサイズ(MBytes) 1024 x 8 x 1024 probes の比較

表現	Indirect Specular	Indirect Diffuse
Raw(FP16)	4,718,592	196,608
SH	6,144	944
CNN1	196,627	0(Specular と共有)
CNN2	18.5	0(Specular と共有)
DCGAN	13.7	0(Specular と共有)

CNN2 と DCGAN は入力パラメタから画像を生成するので学習済みニューラルネットワークの重みサイズのみ.

4. 機械学習の活用

数十 km^2 の面積を持つオープンワールドでリアルタイム大域照明を実現する際，照明データサイズの問題に直面する．オープンワールドの状態，例えば，現在位置，時刻，天候，気候などから Light Field（Light Probes 集合）を生成できるとデータの最適化として便利である．近年，機械学習で最尤推定から派生した陰的密度（Implicit Density）に基づく手法 GAN を使った画像生成がある．GAN は，生成器と識別器を使って，学習データからその確率密度関数を獲得する．CNN を使った DCGAN が良く使われる[4]． 本手法では，DCGAN を大域照明におけるデータサイズの圧縮として活用する．DCGAN 生成器の学習で，Light Field の照明情報生成の確率密度関数を獲得する．

5. 生成的ライトフィールド

本手法は，ゲーム制作での実用を考慮して，リアルタイム大域照明を Point Cloud を使った間接照明とデータ最適化の 2 段階で実現する．

5.1 問題の定式化

Light Field Lはレベルに一定間隔で定点配置した N 個の Light Probes の集合から構成される．Light Probes L_kは，周囲の照明情報，可視情報，幾何情報を保持する．

$$\mathrm{L} = \{L_k\}, k = 1,2,\dots,N$$

画面空間 S の texel に対応する三次元空間の Shading Point $\mathrm{p} = (\mathrm{x},\mathrm{y},\mathrm{z}) \in R^3$について Indirect Diffuse，Indirect Specular を L から計算したい．L はゲームレベルの状態，時間や天候の変化に応じてリアルタイム更新したい.

5.2 第一段階 Point Cloud を使った間接照明

間接照明計算で使う Light Probes が保持する直接照明，可視性，幾何情報のリアルタイム生成に Point Cloud を活用する．

Point Cloud 生成は，View Independent Rasterization(VIR)[5]を活用する．理由は，レベルの光源の追加，時間変化，天候変化といった動的な照明環境の変化に対応するためである．制作ワークフローでレベル変更と照明結果確認の iteration を高速化するためでもある．

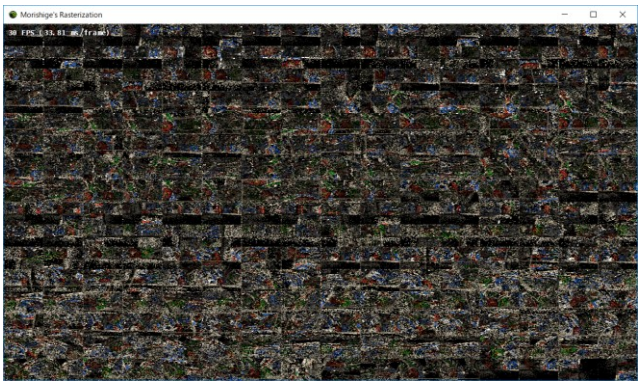

図 2　VIR で Light Probe をリアルタイム並列更新（Sponza）

5.3 第二段階 DCGAN を使ったデータ最適化

ゲームのレベルが完成してきたら，製品リリースに向けてデータサイズの最適化を行う必要が出てくる．そのような目的のために，本手法では，DCGAN でデータを圧縮する．言い換えると，DCGAN の生成器 g が Light Field のデータ生成分布を獲得することで実現する．

$$L_q = g(z)$$

$$z \sim p_z(z), z \in [-1,1]^n$$

L_qは，生成する Light Probe (Radiance, Normal, Visibility)

gは，DCGAN の生成器で畳み込みニューラルネットワーク

DCGAN の構成は，入力潜在変数 z の 100 次元ベクトル，それを生成器 g は，128x128x3(RGB/RGBA)の画像として出力する．識別機 d は，g の生成結果とノイズを加えた結果の判別を[0,1]で行う．

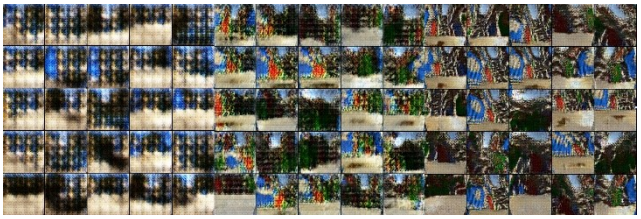

図 3　Sponza の Light Field，DCGAN 学習過程，左から epochs=40，100，290 での生成結果.

5.3.1 学習データ生成

1. Light Field の AABB と Probe 配置間隔を決める
2. Probe の位置からシーンをキャプチャする
3. DCGAN に 2 の結果を入力して学習する(Probe 位置)

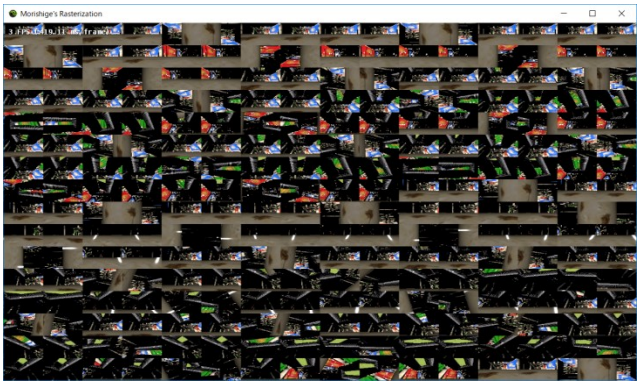

図 4　学習データ生成の例（Radiance, Sponza）

5.3.2 最適化(Optional)

Light Field の 2x2x2 probes 以上の subset Field 単位で支配的な Probe を選択する．その Probe のみを Indirect Specular (Reflection Probe) として扱い、レンダリングで使う．

5.3.3 レンダリング

1. シーンの Point Cloud を生成する
2. ワールド空間位置と時間など環境パラメタを DCGAN の入力変数 z として正規化する．生成器 g に入力することで，Light Field を復元する．
3. Light Field を Relighting する．（固定環境なら省略）
4. 画面空間（Screen Space）の pixel ごとに、Light Field 部分集合をサンプルして，Indirect Specular と Indirect Diffuse を計算する．ワールド空間の Shading Point に対して，近傍 8 個の Light Probes を参照する．

6. 実験

屋内と屋外のレベルで，本手法での結果を示す．

6.1 レベル

表 6 レベル一覧

レベル	内容	Light Field サイズ
Cornell Box	屋内	(2,2,2), 8 probes
Sponza	屋内と屋外	(18,7,11), 1,386 probes
Fireplace Room	屋内	(7,3,5), 105 probes
San-Miguel	屋内と屋外	(86,18,33), 51,084 probes

6.2 実験環境

表で示す動作環境で実験した．

表 7 実験環境

機材	内容
CPU	Intel Core i7 4771K
GPU	NVIDIA RTX2060 GDDR6 6GB
Memory	16 GB GDDR3
OS	Windows 10 build 1809
Framework	Falcor 3.2
PyTorch	version 0.4

6.3 実験結果

第一段階と第二段階について実験した結果を示す．

6.3.1 第一段階 Point Cloud を使ったリアルタイム間接照明

Cornell Box の結果を示す. Light Probes の数は 8 個である．リアルタイムで白色の点光源を移動して Color Bleeding が変化しているのが確認できた．

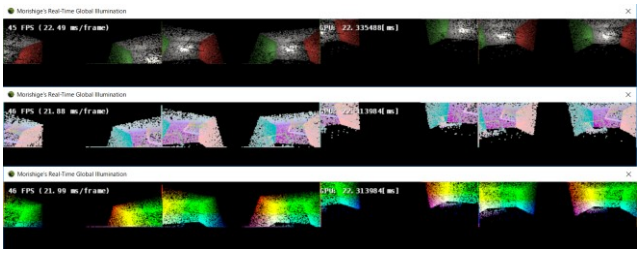

図 5 VIR を使った Point Cloud 生成(上段から Radiance, Normal，Position)

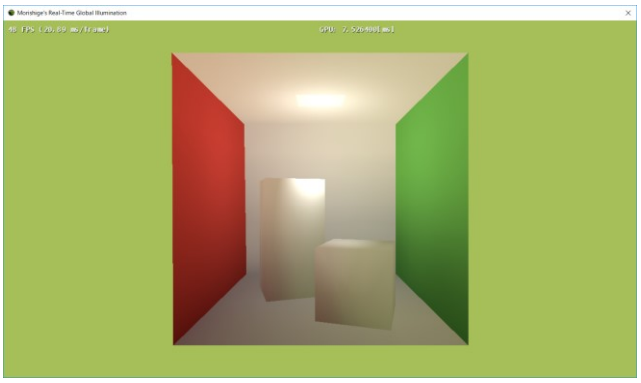

図 6 Point Cloud リアルタイム生成を使った間接照明

図 7 点光源の移動

図 8 点光源の移動

図 9 Irradiance のみ表示

6.3.2 第二段階 DCGAN を使ったデータ最適化

Sponza での結果を示す。DCGAN の学習時間は，表の実験環境で 2 時間程度だった．Indirect Diffuse (Irradiance)のみで

あれば，30 分程度でもそれなりの画像が得られる．実験結果には，Ambient Occlusion と Soft Shadows は含めていない．Indirect Diffuse と Indirect Specular のみ．

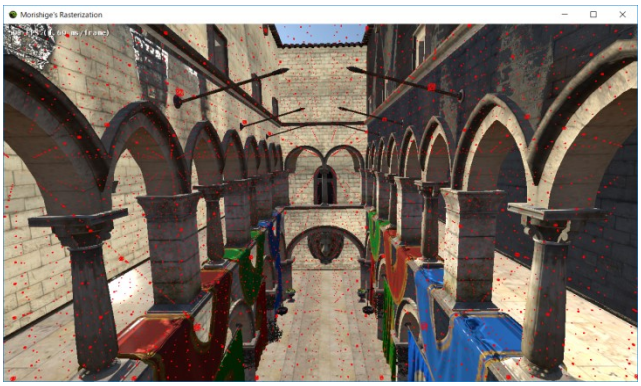

図 10　プローブの生成

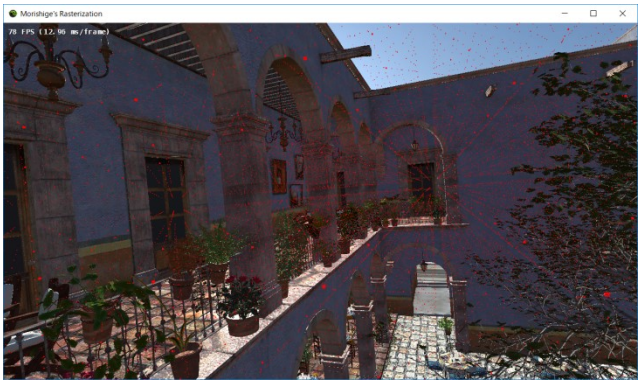

図 11　プローブの生成

朝，昼，夕方，夜について Light Field を生成して，GAN で圧縮して復元した結果である．

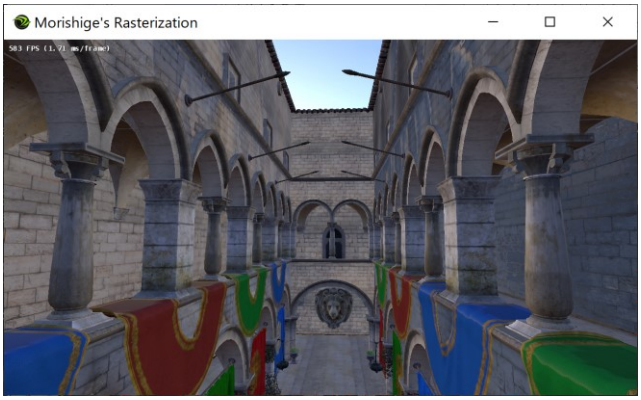

図 12　朝

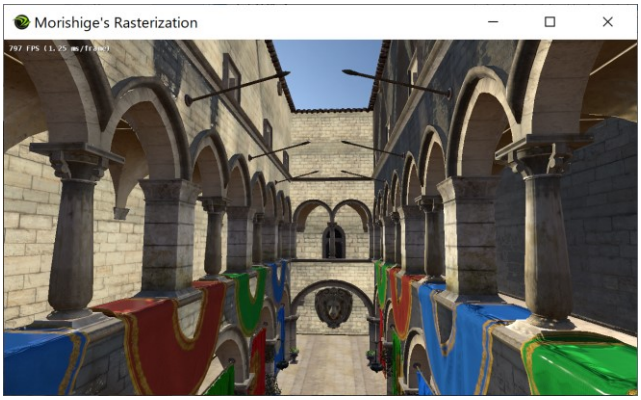

図 13　昼

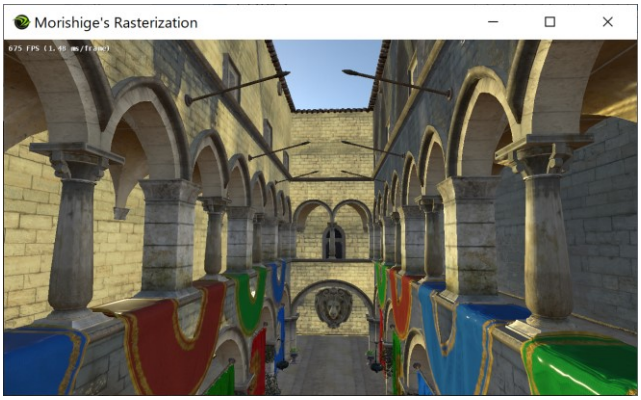

図 14　夕方

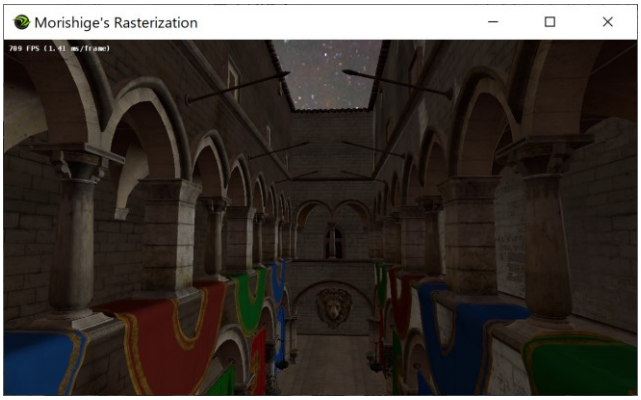

図 15　夜

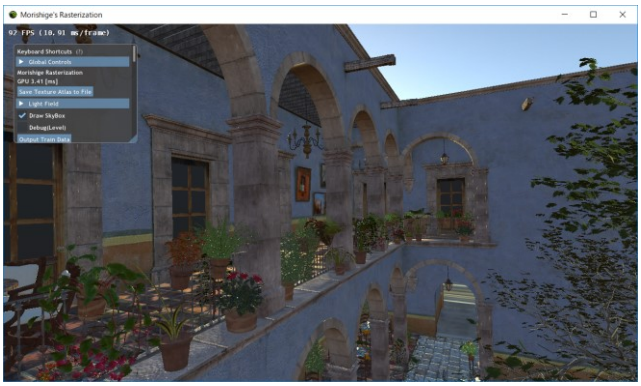

図 16　昼

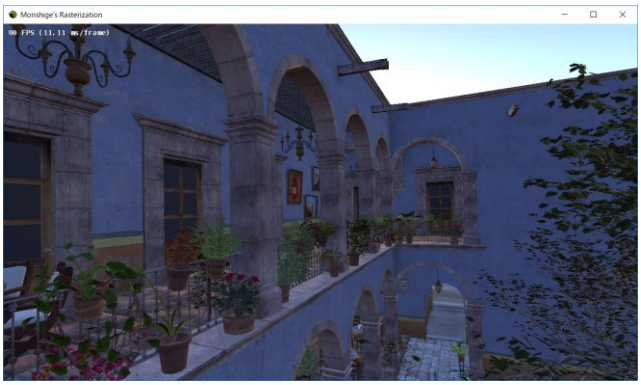

図 17　朝

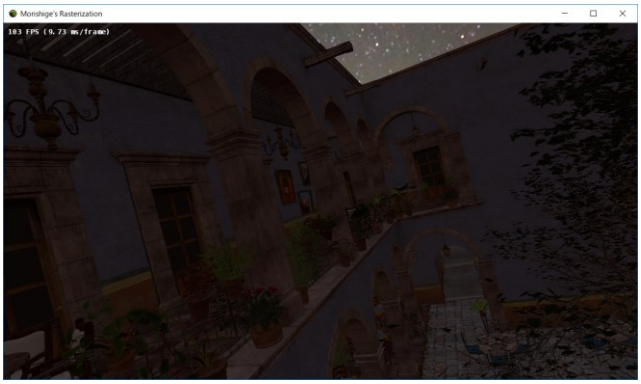

図 18　夜

7. おわりに

DCGAN で Light Field について，Specular Radiance Probe（ゲーム分野では Reflection Probe）と Irradiance Volume を生成できる可能性を少し感じることができた．DCGAN の生成器ニューラルネットワークが獲得した Light Field の確率密度関数から，所望のデータを狙って生成できるように潜在変数をうまく取り扱えるようにしたい．その他，実用のデータ，より広く複雑なレベルで良い結果が生成できるかどうかは引き続き検証しなければならない．

研究としては，次は Light Field 上でリアルタイムの光伝播を実現したい．Volumetric Lighting と Light Field 上での粗い Ray Tracing / Photon Mapping に取り組む．

製品とその制作ワークフローで実際に使えるようにするには，ゲームコンソールでデータを生成するために使う CNN （畳み込みニューラルネットワーク）を実用的な処理速度で実装する必要がある．

要望や意見があれば，morishige_shinya@cygames.co.jp まで，お寄せいただきたい．

参考文献

[1] “A Certain Slant of Light: Past, Present and Future Challenges of Global Illumination in Games”. http://openproblems.realtimerendering.com/s2017/index.html
[2] “Real-Time Global Illumination using Precomputed Light Field Probes”. https://research.nvidia.com/publication/real-time-global-illumination-using-precomputed-light-field-probes.
[3] “Global Illumination in Tom Clancy’s The Division”. http://mrakobes.com/Nikolay.Stefanov.GDC.2016.pdf
[4] “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. https://arxiv.org/abs/1511.06434
[5] “Real-Time View Independent Rasterization for Multi-View Rendering”. https://www.csc2.ncsu.edu/faculty/healey/download/eg.17.pdf