HDRI と機械学習を用いた物理ベースレンダリングの評価手法

Evaluation and improvement methods for Physically-Based Rendering using HDRI and machine learning

川島　基展† 菅野 昌人‡ 山口 翔平‡ 鈴木 雅幸‡

Motonobu KAWASHIMA† Masato KANNO‡ Shohei YAMAGUCHI‡ and Masayuki SUZUKI‡

†東京工科大学 †Tokyo University of Technology

‡株式会社バンダイナムコスタジオ ‡Bandai Namco Studios Inc.

E-mail: †kawashimamn@stf.teu.ac.jp, ‡{m-kanno, s6-yamaguchi, m6-suzuki}@bns-g.com

1. はじめに

ビデオゲームコンテンツや CG アニメーションコンテンツにおける背景やキャラクターには物理ベースレンダリングによる写実的な質感表現を行うことが有効であり,近年の主流となっている.しかし,アーティストが物理ベースレンダリングやライティングの理論を正しく理解せずにパラメータの設定を行ってしまうと,物理ベースレンダリングを用いても物理的に正しくない,不自然な映像表現となってしまう.また,現状のルックディベロップメント（以下ルックデブ）工程におけるマテリアル設定が適切であるかどうかの判断は属人的であり,オブジェクトの質感が自然に見えるかどうかはアーティストの感覚で評価されがちである.

そこで本研究では,HDRI による物理的に適正なライティング環境下で,機械学習を用いて物理ベースレンダリングによるマテリアルを客観的に評価し,不適切な設定箇所を検出することができる手法を提案する.これにより,ビデオゲームコンテンツ開発や CG アニメーションコンテンツ制作における普遍性のあるルックデブ環境の構築に寄与することを目的とする.

2.物理ベースレンダリングとルックデブ工程の課題

2.1 物理ベースレンダリングによるマテリアル表現

今日のビデオゲームコンテンツや CG アニメーションコンテンツの制作においては,キャラクターや背景モデルなどのグラフィックアセットに対して物理ベースレンダリング（Physically-Based Rendering, 以下 PBR）によるマテリアル表現を行うのが一般的である.PBR は,従来の Lambert 法や Blinn-Phong 法のような数学的に現実の質感を模倣する手法とは異なり,表面の粗さ（Roughness）や金属性（Metallic）のような物理的な属性パラメータを用いて現実世界の光の反射や散乱現象を表現するレンダリング手法である[1].この手法は,リアリスティックな映像表現にはもちろんのこと,近年ではデフォルメされたアニメーション作品において,ノンフォトリアリスティック表現を行う際にも重要視されている.Burley らは,Walt Disney Animation Studios でのノンフォトリアリスティックレンダリングを用いた長編アニメーション映画制作において,図 2-1 のような PBR に基づくシェーディングモデルを整備し,異なるシーン環境間においても一貫性のあるマテリアル表現を実現した [2] .また,Pixar Animation Studios や Sony Pictures / Imageworks などの主要なアニメーションスタジオにおける長編劇場映画作品においても,図 2-2 に示すように,PBR による普遍的なシェーディングを基盤としつつも独自の様式化されたマテリアル表現が行われている[2][3] .

図 2-1. Pixar Animation Studios における PBR によるマテリアル表現の例　©Walt Disney Pictures

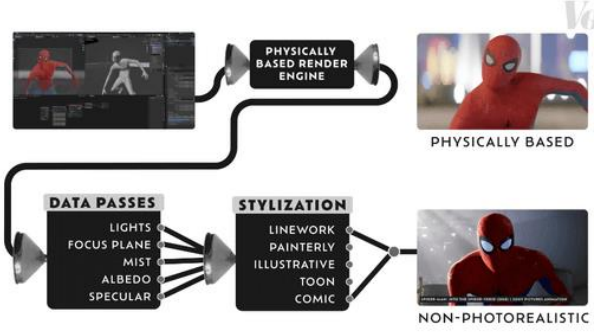

図 2-2. "Spider-Verse"における PBR によるシェーディングを発展させたマテリアル表現の例
©Sony Pictures/Imageworks

2.2 PBR を用いるルックデブ工程の課題

キャラクターや背景などの見た目の様式を決めるルックデブ工程においては,従来のマテリアル表現では作中のさまざまなシーンごとに異なる設定を行って調整する必要があり,シェーダ設計に際しては多くのパラメータを用意する必要があり,アーティストによる設定作業の手間が膨大であった.上述のように,PBR に基づくマテリアル表現を行うことで,異なるシーン間での質感の普遍性をもたらすことができる.そのため,PBR を活用することが近年のビデオゲームコンテンツや CG アニメーションコンテンツの制作におけるマテリアル表現の主流となっている.ただし,これはアーティストが PBR 理論に精通していることが前提であり,特定のシーンに合わせて理論に沿わないパラメータ設定を行ってしまうと,別のシーンでは不自然な見た目の映像表現になってしまう.商業コンテンツで扱うモデルデータの量は膨大であり,それらのほとんどの PBR マテリアルのパラメータをデザイナーがテクスチャマップとして作成する.そのため,ゲームエンジンや CG ソフトウェアに組み込んでからマテリアル設定を精査することは難しく,また,さまざまなシーンにおけるライティング環境下で不適切なマテリアル設定を検出する手間を割くのは困難である.

3. 機械学習を用いた PBR の評価手法

上述の課題を解決するために,本研究では機械学習による物理ベースレンダリングの評価手法を提案する.図 3-1 に提案手法による評価のプロセスを示す.

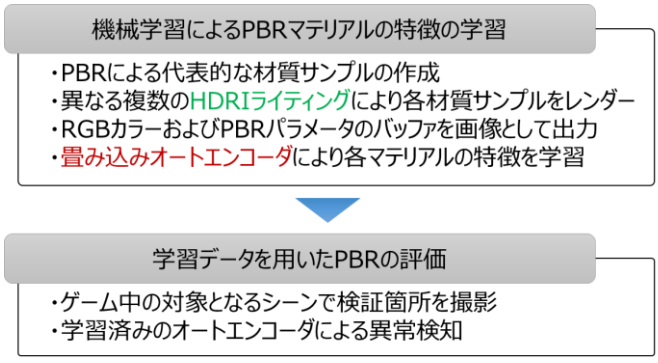

図 3-1. 提案手法による PBR マテリアルの特徴の学習および評価のプロセス

3.1 PBR の評価方法

PBR に基づくマテリアル設定が適切かどうかを評価するため,本研究では画像データからのエラー検知を得意とする畳み込みオートエンコーダによる事前学習を行う.学習データとしては,図 3.1 のように現実の計測値に基づいて PBR のパラメータ設定が適切に行われているマテリアルの RGB カラー画像を用いる.この際,ライティングについても現実に忠実でなければ適切な正解データの出力とはいえないため,本手法においては環境の異なる複数の HDRI によってライティングを行った画像を用いて学習を行うことが肝要である.

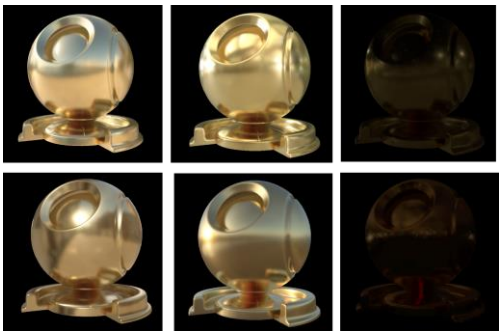

図 3.1. 学習データ 1：
異なる HDRI ライティングを施した
RGB カラー出力画像

また,レンダリングの結果だけでなく,マテリアルのパラメータ設定値からも評価できるよう,図 3.2 のように R チャンネルに輝度値,G チャンネルに Metallic, B チャンネルに Specular の輝度値のバッファ出力を混合した画像も併せて出力し,それぞれについての事前学習と評価を行う.

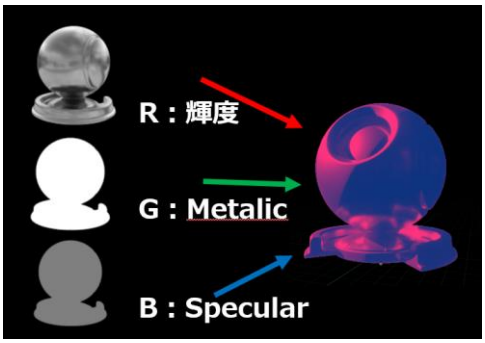

図 3.2. 学習データ 2：
輝度値,Metallic,Specular のバッファ出力の
混合画像の例

一般に,テクスチャマップ入力が伴う場合などはマテリアルのパラメータ値を直接検証することが難しくなるが,上記のようにレンダリングの際に出力する各バッファ要素を検証することにより,マテリアル設定の複雑さに依存しない検証が可能になる.

評価に際しては,対象となるビデオゲームコンテンツの中のシーンにて,カメラ視点からの RGB 画像および混合画像を出力する.その後,学習済みの畳み込みオートエンコーダを用いて異常検知を行い,どちらかの画像の復号ができない場合にはマテリアル設定に不備があるものと判定する.

4. 実用検証

4.1 実用検証の概要

前章で述べた手法の実用性を検証するため,本研究ではUnreal Engine 5とOpenCV, Pythonを用いてビデオゲームコンテンツの開発工程に即して利用可能なツールを実装した.図 4.1 に評価ツールの構成を示す.

図 4.1. 実用検証のための評価ツールの構成

4.2 畳み込みオートエンコーダの設計

コンテンツからの出力画像を評価し,マテリアル設定の異常を検出するために,図 4.2 に示すような畳み込みオートエンコーダをTensorflowとKerasを用いて実装した.畳み込みオートエンコーダでは,前章で述べた正解データとしての RGB 画像およびパラメータの混合画像を入力し,前半のエンコーダ層では次第に画像サイズを小さく圧縮していき,後半のデコーダで復元していく過程で特徴の学習を行う.学習の結果,デコーダで復元できない画像は学習データの特徴を持たない異常データとして検出できる.本研究では,学習および評価に用いる入力画像のサイズを 128×128px とし,2ブロック構成のエンコーダ層で 32×32px まで圧縮のうえ,デコーダ層でふたたび 128×128px まで復元する.

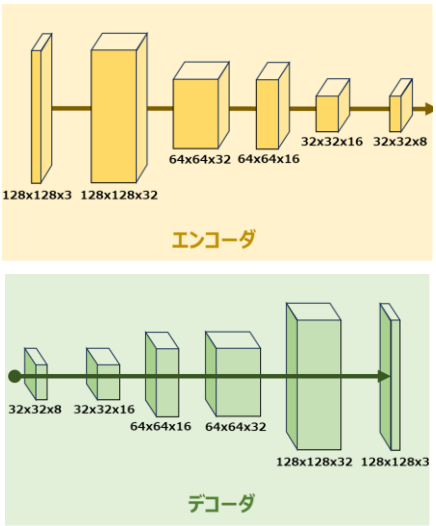

図 4.2. 畳み込みオートエンコーダの設計

4.3 学習データの生成

事前の学習では Unreal Engine 上に主要な材質を適用したオブジェクトを用意し,オートエンコーダの仕様にしたがい 128×128px のサイズで画像を出力した.マテリアルの計測値については,出版物など根拠のある情報ソースに基づいて Open API として公開している[5]および Epic Games 社が公開している計測値[6]を指標とした.また,現実に忠実なライティング環境として図 4.3 に示すような 6 種類の異なる環境の HDRI によるライティングを行い,それぞれの材質のレンダリングを行った.また,パラメータの混合画像についてはポストプロセスシェーダによって実装のうえ出力した.対象以外の要素を学習させないよう,いずれの画像についてもシェーダ処理によって対象以外を除外した.

図 4.3. 実用検証に用いた HDRI 画像

生成した学習用データを図 4.4 に示す.学習用データは,金,銅,プラスチックなどの主要な材質サンプルにつき,上記の 6 種類の HDRI によるライティングを行い,各サンプルの回転画像を 13 枚ずつ,計 78 枚ずつ出力した.畳み込みエンコーダではそれぞれの画像に対して垂直・水平方向の画像反転,回転操作を加えたオーグメンテーションを行い,データ数を 4 倍まで増やした.

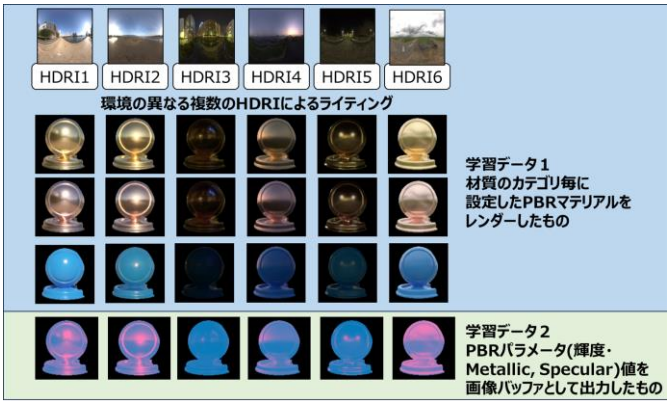

図 4.4. 生成した学習データ（一部抜粋）

4.4 PBR の評価

学習データに基づく PBR の評価を行うモジュールは,ビデオゲームコンテンツの開発中に用いる機能テスト関数を Unreal Engine のプラグインとして実装した.検証を行う対象には材質の分類を示すタグを付与しておき,機能テスト実行中にユーザが検査対象を撮影することで評価画像を生成する.評価画像はテスト終了時にオートエンコーダに転送し,推論による評価を行えるようにした.なお,評価対象のコンテンツは図 4.5 のように[7]の一部を修正したものをベースとし,PBR のマテリアルパラメータを設定するときにデザイナーが起こしがちな次の a～c のミスを想定し,正しいマテリアル設定ができているものと混合した.

図 4.5. 評価対象のコンテンツ

a. 図 4.6 の作例では, 金属製品の Base Color 要素に関する色調整を Emissive Color で行ってしまい,陰影感に乏しく,全体的に明るくなってしまっている. 今回の評価実験では, 金属の分類で検査したところ,パラメータの混合画像に異常が検知できた.

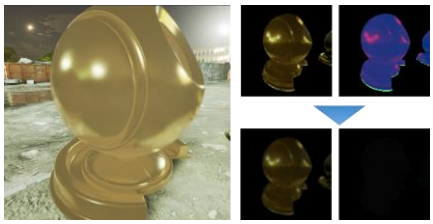

図 4.6. Emissive Color に起因する問題の検出例

b. 図 4.7 の作例では,樹脂製品の PBR 属性がじゅうぶん考慮されていない.色彩やコントラストの調整のために Metallic 値を調整してしまっており,表面色が暗く, 過剰な映り込みが生じている.今回の評価実験では,樹脂の分類で検査したところ, パラメータの混合画像に異常が検知できた.

図 4.7. Metallic 値に起因する問題の検出例

c. 表面の散乱性を表現するためには Roughness 値を調整するのが適切であるが,図 4.8 の作例では,ハイライトを得るために誤って Specular 値を高く設定してしまっている.ここでは金属の分類で検査したところ,RGB 画像,パラメータ画像の両方に異常が検知できた.

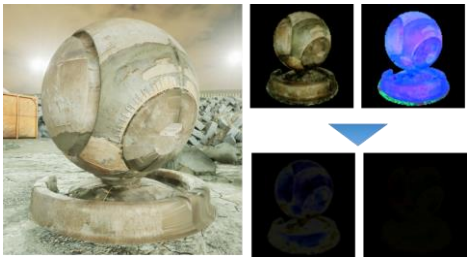

図 4.8. Specular 値に起因する問題の検出例

検証では,現時点では学習で用いたものと同一のモデル以外では異常が検知できなかった,また,RGB 画像よりもパラメータ画像による異常検知に依存しており,学習精度の課題は残るものの,デザイナーの感覚的な評価に依らずにマテリアル設定の不備を検出できることを確認した.

5. 考察と展望

本研究では,HDRI による物理的に適正なライティング環境下で,機械学習を用いて物理ベースレンダリングによるマテリアルを客観的に評価し,不適切な設定箇所を検出できる手法を提案した.本研究ではビデオゲームコンテンツ開発に即した実装を行ったが,学習・評価モジュールはゲームエンジンとは独立しており,CG アニメーション制作にも活用可能である.現時点ではマテリアル設定の異常を検出するまでに留まっているが,今後はゲームエンジンと学習環境との連携を強め,利便を高められるよう拡張していきたい.

文　献

[1] Robert L. Cook, Kenneth E. Torrance, A Reflection Model for Computer Graphics, ACM, 1982
[2] Burley, Physically-Based Shading at Disney, ACM Siggraph Course, 2012
[3] Smits, Reflection Model Design for WALL-E and Up, ACM Siggraph Course, 2012
[4] Vega, How "Spider-Verse" forced animation to evolve,https://www.vox.com/videos/23349660/spider-verse-cgi-animation-pixar-photorealism, 2022
[5] PHYSICALLYBASED, https://physicallybased.info
[6] Epic Games Inc., 物理ベースのマテリアル https://dev.epicgames.com/documentation/ja-jp/unreal-engine/physically-based-materials-in-unreal-engine?application_version=5.3
[7] Scans Factory, [SCANS] Construction Site: Loader, https://www.unrealengine.com/marketplace/ja/product/scans-construction-site