

# ハードウェアを用いた実時間描画を実現するためのBTF合成手法 BTF Synthesis for Hardware Accelerated Real-Time Rendering

徐慶 大<sup>†</sup>  
Keidai Jyo

川崎 洋<sup>†</sup>  
Hiroshi Kawasaki

古川 亮<sup>‡</sup>  
Furukawa Ryo

大澤 裕<sup>†</sup>  
Yutaka Ohsawa

## 1. はじめに

本論文ではハードウェアを用いて、効率よくBTF(bidirectional texture function)を合成する手法を提案する。BTFとは物体表面テクスチャの反射特性を、光源及び視点の変化として記述したものである。任意の物体表面にBTFを適用するためには、BTFを合成する必要がある。これまでBTF合成は、テクスチャ解析に基づいた手法で行われてきた。一方、通常のテクスチャ合成では、近年パッチベースの手法が注目されている。そこで、本論文では、パッチベースによるBTFの合成手法を提案する。提案する手法は圧縮したBTFに直接適用可能なため、データサイズおよび計算量の削減に大きな効果がある。

## 2. BTF合成を用いたCGレンダリング

Danaら[1]によってBTFが提案され以来、BTFの研究がCG分野で盛んに行われてきた。BTFは、視点と光源の4次元のデータセットから成り、その容量が膨大であり演算コストも高い。そのため、実際にBTFを利用しCGを製作するためには、小さなBTFを合成し大きなテクスチャを生成する必要がある。そこで、BTFの合成に関する研究がこれまで行われてきた。

Liuら[2]はnon-parametricによるShape-from-shading技術を使用したBTF合成手法を提案している。しかし、演算コストが高く適用可能な素材に限られているため、実用は難しい。そこでTongら[3]はsurface texturesを用いたBTF合成を行った。この手法は幅広い素材に適用可能である。ところが提案手法はテクスチャの特徴解析が必要であり、アルゴリズムも複雑であった。

本論文ではシンプルかつ効率のよいBTF合成手法を提案する。これにより、光源だけでなく視点変化にも対応したBTFの実時間での描画が可能となる。提案する手法は、既存のパッチベースによるシンプルなテクスチャ合成アルゴリズムをBTF合成に拡張したものであり、演算コスト軽減のため圧縮したBTFデータに対し合成を行う手法となっている。以降ではBTFサンプリングと圧縮法、BTF合成アルゴリズムとそれを用いた実時間描画手法について述べる。

## 3. BTF合成手法

### 3.1 BTFサンプリングと圧縮

本論文で利用するBTFデータは[4]で提案された計測機器を用いるか、3DCGソフトにより、仮想カメラと仮想光源を用いてサンプリングすることで得られる。

一般にBTFのデータセットは膨大なサイズである。そこで、我々はPRTに基づいた球面調和関数(SH:spherical harmonics)を拡張することにより、BTFデータを圧縮する。これにより、1GBデータを10MB以下に圧縮することが出来る。

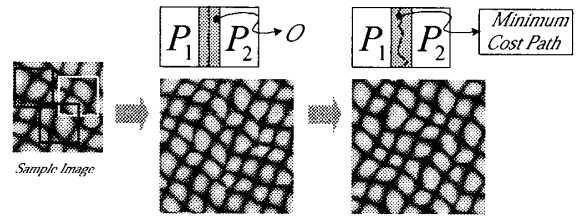


図1: Image Quilting

### 3.2 パッチベースによるBTF合成アルゴリズム

テクスチャ合成は大きくピクセルベースとパッチベースの手法に分けることができる。パッチベースの手法はピクセルベースの手法に対して、境界線の処理等が必要となるが、演算コストや画質において優れた手法と考えられている。これを踏まえ、本研究ではパッチベースによるテクスチャ合成手法を拡張し利用する。その際、RGB値ではなくSHにより圧縮したBTFデータの係数を合成することで、大幅な演算コストの削減を実現した。本論文ではEfrosら[5]のImage Quiltingによる単画像に対するテクスチャ合成手法を用いた。

#### 3.2.1 Image Quiltingによる合成手法

Image Quiltingとは図1のようにパッチ $P_1$ と $P_2$ のオーバーラップ領域 $O$ において最もエラーが小さいパスを検索し、繋ぎ合わせる技術である。以下にそのアルゴリズムを簡単に紹介する。

1.  $P_1$ 上のオーバーラップ領域 $O$ と類似性の高い領域をサンプル画像内から探索し、それを $P_2$ とする。
2.  $P_1$ と $P_2$ のオーバーラップ領域 $O$ において、パスに基づいたエネルギー $E$ を求める。
3. エネルギー $E$ が最小となるパスを求め、パッチを繋ぎ合わせる。
4. 上記の手続きを合成が完成するまで繰り返す。

#### 3.2.2 Image QuiltingのBTFへの拡張

BTFデータは視点と光源の4次元の広がりを持つ。そのため、2次元情報に基づくImage Quiltingの手法を視点と光源方向の4次元に拡張する必要がある。以下ではBTFにおけるパッチの選択法および最小パスの探索法について述べる。

BTFはデータサイズが巨大なため、類似度の高いパッチを全データを用いて選択することは難しい。そこで、本研究ではBTF合成の際、境界の不連続の主な原因は低周波数成分にあると考え、低次のSH係数のみを用いることで演算コストを軽減する。また、真上からみたテクスチャが合成の際重要であるため、これを考慮した重み $\sigma(i)$ を与える。その結果、パッチの類似度 $d$ と許容類

<sup>†</sup>埼玉大学院理工学研究科

<sup>‡</sup>広島市立大学情報科学部

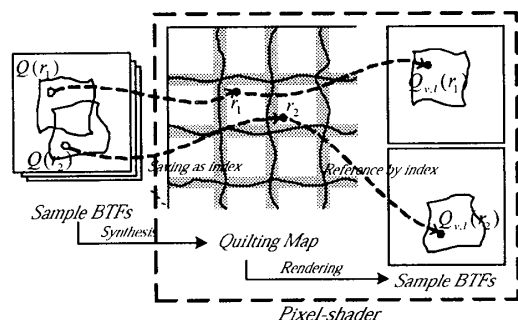


図2: Rendering by Quilting Map

似度  $d_{max}$  は以下のように定義される。

$$d = \left\{ \frac{1}{N} \sum_{i=1}^V v_i \sum_{j=1}^L \sigma(j) \sum_{k=1}^N (C_k^1 - C_k^2)^2 \right\}^{\frac{1}{2}} \quad (1)$$

$$d_{max} = \left\{ \frac{1}{N} \sum_{i=1}^V v_i \sum_{j=1}^L \sigma(j) \sum_{k=1}^N (pC_k^1)^2 \right\}^{\frac{1}{2}} \quad (2)$$

ここで、 $V$  は視点方向のデータ数、 $L$  は光源方向のデータ数、 $N$  はオーバーラップ領域内のピクセル数である。 $C_k^1$  と  $C_k^2$  は  $P_1$  と  $P_2$  のオーバーラップ領域  $O$  における  $k$  番目のピクセル値を表し、 $p$  は許容する類似度を決定する係数であり、ユーザにより指定される。

次にパッチ同士を滑らかにつなぐパスを探索する。この時、パッチ選択の時と同様、真上からテクスチャを見た時の影響が最大となるように重み  $\rho(i)$  を定義する。これにより、斜めから見た時、テクスチャがつぶれてしまい、パスの探索に悪影響をおよぼすことを避けることができる。この際、パスのエネルギー  $E$  は、以下に定義される  $e_p(\rho(i))$  を用いて次のように計算される。

$$e_p(\rho(i)) = \sum_{i=1}^V \rho(i) \sum_{j=1}^L \alpha_j (O_p^1 - O_p^2)^2 \quad (3)$$

$$E_{k,l} = e_{k,l} + \min(E_{k-1,l-1}, E_{k-1,l}, E_{k-1,l+1}) \quad (4)$$

ここで、 $O_p^1$  と  $O_p^2$  は  $P_1$  と  $P_2$  のオーバーラップ領域のピクセル値を表し、 $k$  と  $l$  はピクセル座標を指す。横方向の最小コストパスも同様の手法で求めることができる。

### 3.3 インデクスによるリアルタイムレンダリング

さらに、本論文では任意サイズのテクスチャの実時間描画を可能とするため、図2のように合成結果をサンプルBTFデータのピクセル座標を指すインデクスとして保存する。 $r$  をインデクスとし、サンプルBTFデータにおけるピクセル値を  $Q(r)$  で表す。ここでインデクス  $r$  をピクセル値として保存した画像を Quilting Map (QM) と呼ぶ。これにより、合成結果を一枚のQMとして保存することができる。QMから  $r$  を読み出し、ピクセルシェーダの機能を用いてBTFデータの  $Q_{v,l}(r)$  を得る。ここで  $v$  と  $l$  はBTFデータの視点と光源を表す。実際にはピクセルシェーダがSH係数を用いて  $v$  と  $l$  に対応する実際のRGB値  $Q_{v,l}(r)$  をオンラインで計算している。また、各頂点ごとに計算が行われるため、点光源による実時間描画も可能である。

## 4. 実験

図3、4は64x64pixelのBTFデータから32x32pixelのパッチサイズを用いて、256x256pixelに合成したものを描画した結果である。パッチ間の境界線が滑らかに合成できていることがわかる。図3の(a)(b)は点光源を用

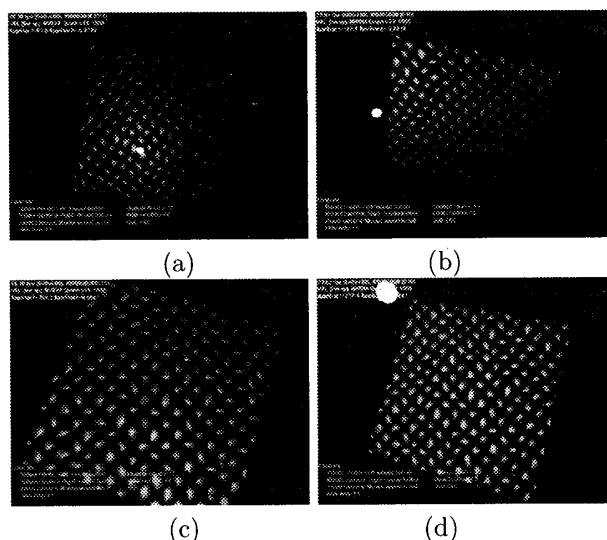


図3: Rendering by synthesized BTF

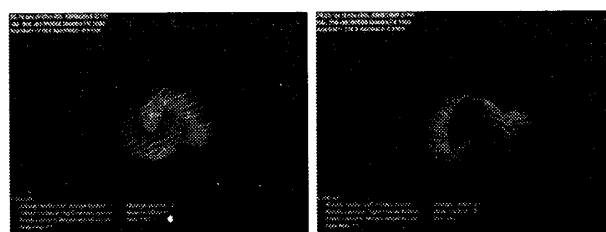


図4: Complicated shaped object

いた例で、(c)(d)は視点変化による描画結果である。図4は複雑な形状のオブジェクトを用いて描画した結果である。いずれも違和感なく描画できている。

## 5. まとめ

球面調和関数 (SH) により圧縮したBTFデータを Image Quilting を拡張した手法により、パッチベースで合成する手法を提案した。また、合成結果をQM (Quilting Map) というインデクスデータとして保存することでデータサイズの大幅な削減とシェーダによるリアルタイム処理を実現することができた。本手法を用いれば、玉虫色のような複雑な反射特性を持った素材や、細かい凹凸のある素材を、現実感豊かにリアルタイムに視点や光源を動かしながら、描画することが出来る。

## 参考文献

- [1] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. In *IEEE CVPR*, pp. 151-157, 1997.
- [2] X. Liu, Y. Yu, and H.-Y. Shum. Synthesizing bidirectional texture functions for real-world surfaces. In *SIGGRAPH 2001*, pp. 97-106.
- [3] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H.-Y. Shum. Synthesis of bidirectional texture functions on arbitrary surfaces. In *SIGGRAPH 2002*, pp. 665-672.
- [4] R. Furukawa, H. Kawasaki, R. Kawatsu, D. Miyazaki, K. Ikeuchi, and M. Sakauchi. Appearance based object modeling using texture database: Acquisition, compression and redering. In *Rendering Techniques 2002*, pp. 257-265.
- [5] A.A. Efros and W.T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH 2001*, pp. 341-346.