

3次元物体の形状・表面特性の獲得および表現手法

Sampling and Modeling of Shape and Reflectance of 3-D Objects

古川 亮[†], 川崎 洋^{††}, 中村 泰明[†], 正会員 大澤 裕^{††}Ryo Furukawa[†], Hiroshi Kawasaki^{††}, Yasuaki Nakamura[†] and Yutaka Ohsawa^{††}

Abstract In this paper, we developed a method for sampling and modeling reflectance of 3-D objects and performing real-time rendering of modeled objects. Our method can be used to render modeled objects from an arbitrary viewing direction with illuminations from point lights at any given place or with any illumination map. To achieve this goal, we developed a platform that can sample appearances of target objects with 4DOFs parameters of light and viewing directions. Light-dependent variations of sampled data were approximated using spherical harmonics functions as bases. To render a modeled object, we generated its appearance by synthesizing the effects of illuminations using spherical harmonics functions and by interpolating between sampled viewing directions. The appearance was then texture-mapped to the shape model. Using our method, we could represent reflectance properties of surfaces of 3-D objects that were parametrized by 4DOFs light and viewing directions. We also successfully rendered the modeled objects using programmable vertex/pixel shaders with real-time performance.

キーワード：実時間描画，球面調和関数，見えの標本化，視点依存テクスチャマッピング，イメージベースレンダリング，表面光線空間法

1. ま え が き

実在する立体物体を，任意の光源環境および視点位置において実時間描画することは，仮想博物館や，仮想現実感環境など，多くの応用が考えられる．

このための手法として，実時間，あるいはオフラインによるイメージベースレンダリングの手法が研究されている．これらの手法では，実在する物体の見えをさまざまな条件で標本化し，それらの標本集合から画像を合成する．任意の光源環境および視点位置においてこれを実現するには，光源方向に関して2次元，視点位置に関して2次元の合計4自由度のパラメータを変数として画像を標本化する必要がある．これは困難なことであるため，多くの研究では，標本化は光源方向・視点位置のパラメータ空間の一部に対してのみ行われている．古川らは3次元物体の光源方向，視点方向に関して完全な標本化を行う装置⁵⁾を作り，実在の物体を標本化し，それらの物体の描画を行った．た

だし，すべての処理をCPUで行っていたこともあり，描画フレームレートは5fps程度であった．

光源環境の変化によるテクスチャの変化をモデル化するための手法として，球面調和関数展開を利用する方法が提案されている¹¹⁾¹⁴⁾．この手法は，光源環境による各テクセル値の変化を球面調和関数の係数として表現し，各画素値にそれらの係数を保存することでテクスチャの変化をモデル化する．これらのテクスチャは，環境マップとして表された光源や，任意の点光源に対して実時間描画可能である．よって，球面調和関数の係数をコード化したテクスチャを，すべての視点について用意し，描画時にそれらを選択するようにすれば，任意の視点方向，光源環境での描画が可能になると考えられる．

複数の視点において標本化された見えの画像から，各視点からのテクスチャを作成する手法としては，形状モデルに対して，単一のテクスチャ座標を定義し，すべての視点においてこれを利用することが考えられる．この場合，各視点からのテクスチャ画像を作成するには，標本集合からそのテクスチャ座標についてのテクセル値を再標本化する必要がある．この方法は，描画時に通常のテクスチャマッピングのアルゴリズムを利用しやすい．その反面，視線に対して傾いた面や，オクルージョン等によって見えない面にも同じ解像度でテクスチャが割り当てられるため，利用するテクスチャ座標によっては，テクスチャ座標空間の利用効率が悪い，テクスチャの解像度が不均一になる，テク

2005年9月1日受付，2005年12月13日再受付，2006年2月14日採録
†広島市立大学 情報科学部

(〒731-3194 広島市安佐南区大塚東 3-4-1, TEL 082-830-1686)

††埼玉大学 工学部

(〒338-8570 埼玉県さいたま市桜区下大久保 255, TEL 048-858-9211)

† Faculty of information science, Hiroshima city university

(3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194, Japan)

†† Faculty of engineering, Saitama university

(255, Shimo-ohkubo, Sakura-ku, Saitama, Saitama 338-8570, Japan)

スチャの連続性が損なわれるといった問題が生じる。また、再標本化による画質の低下も起こる。

別の手法として、各視点に対して異なるテクスチャ座標を利用することが考えられる。このような方法の例として、射影テクスチャマッピングがある。射影テクスチャマッピングを利用すると、各視点における見えの画像を、テクスチャとしてそのまま利用可能である。このため、テクスチャの解像度は、各視点に適したものになり、オクルージョン等によるテクスチャ領域の無駄が生じない。また、再標本化が最小限ですむ、という利点もある。反面、テクスチャとして利用される見えの画像の周辺部に、利用されない領域が生じる問題や、形状モデルの各頂点のテクスチャ座標を描画時に計算する必要があるという問題がある。ただし、最近のグラフィックハードウェアでは、テクスチャ座標の計算を効率的に行うことが可能であり、計算コストの問題は解消されつつある。

本論文では、球面調和関数の係数をコード化したテクスチャと、視点に依存した射影テクスチャマッピングによって、標本化された3次元物体を任意の光源方向、視点方向について実時間描画する手法を提案する。

2. 関連研究

これまで多くの3次元物体の表面特性を獲得しモデル化する研究が行われてきた。標本化されたデータは、表面光線空間や、空間BRDFとしてモデル化される³⁾⁵⁾⁸⁾¹⁰⁾¹²⁾¹⁷⁾。これらの研究において、実時間描画は研究の主要課題ではなかった。

材質の反射特性を数値モデル化し、実時間で描画するために、多く用いられている手法として、因子分解法と、球面調和関数展開による球面関数近似が挙げられる。因子分解法は、多次元の関数を、より少ない次元を持つ複数の関数の積和として表現する手法である。例えば、視線方向と光源方向の両方に依存するBRDFを、視線方向のみに依存する関数群と、光源方向のみに依存する関数群とで表現することができる。KautsらやMcCoolらは、因子分解法によってBRDFを複数の2次元テクスチャとして表現し、表現されたBRDFを素材として適用し、実時間描画を行った⁶⁾⁹⁾。Chenらは、光源を固定して視点を変化させながら物体を標本化し、視点の変化と、モデル表面のテクスチャ座標とを因子分解する手法を用いて、固定された光源方向に対し、任意の視点からのシーンを実時間描画する手法を提案している¹⁾。

材質の反射特性の光源方向による変化を、球面調和関数展開によって近似する手法は、Ramamoorthiらによって提案された¹¹⁾。この手法では、物体の周囲の光源を、球面調和関数で近似することで、環境マップによる光源などの大域的照明を効率的に扱うことが可能である。Sloanらは、球面調和関数の係数をテクスチャにする事前計算放射輝度伝達(PRT)と呼ばれる手法を提案し、任意の平行光源下で

の実時間描画を実現した¹⁴⁾。また、Malzbenderらは、多項式テクスチャマップ(PTM)によって光源に依存するテクスチャを表現し、実時間描画した⁷⁾。PTMは、球面調和関数を陽には使っていないが、方向パラメータの球面座標表示の多項式であるという点で、球面調和関数と共通しており、非常に近い手法であると考えられる。

上記の研究は、視点方向、光源方向のいずれかが固定であるか、あるいは表面の材質が均質かつ滑らかである、という仮定がされている。表面の材質が均質でない、あるいは微細な表面形状を持つ材質の描画を、視点方向、光源方向の両方を変化ながら行うことが可能な手法の一つとして、双方向テクスチャ関数(BTF)が挙げられる。BTFは、視点方向、光源方向、2次元テクスチャ座標をパラメータとした、6自由度変数の関数であり、テクスチャの光源方向、視線方向に依存した変化をモデル化可能である²⁾。Sattlerらは布のBTFを実際に測定し、それら標本集合から実時間描画を実現した¹³⁾。Suykensらは因子分解法の手法を拡張し、テクスチャへの格納手法を工夫することでBTFの実時間描画を実現した¹⁶⁾。Sloanらは、自己シャドウなどの効果とBTFを同時に実時間描画する手法を提案し、双スケール放射輝度伝達(BRT)と名づけた¹⁵⁾。これらの研究は、平面状の素材から標本化された比較的小さなサイズのBTFを、モデル全体に繰返しマッピングすることを目的とするものであり、立体形状をもった物体の見えを標本化し、再現しようとするものではない。もちろん、物体の表面全体にテクスチャ座標を定め、大きなサイズのBTFを用意すれば、任意の視点方向、光源方向に対しての物体の見えを表現することが可能である。しかし、3次元物体の表面全体にテクスチャ座標を定めることは、困難な問題である。また、獲得された画像集合からBTFを作成するには再標本化を要するため、画質の低下を招きかねないという問題もある。提案手法では、表面全体にテクスチャ座標を定める必要がなく、画像の再標本化も行わないため、画質の低下を避けることができる。

また、本手法は、球面調和関数展開を利用した手法であるため、SloanらのBRT¹⁵⁾と同様、大域的照明への対応が可能である。ただし、Sloanらの手法が、既存のBTFのモデル化および描画のみを対象としており、3次元形状物体の見えを獲得し再現することについて触れていないのに対し、本論文で提案する手法は、任意の視点方向および光源方向で、3次元形状を持った物体の見えを、材質変化や表面の凹凸を含めて再現することが可能である。

3. 標本化装置による見えの画像データベースの取得

物体の見えの標本化を行うために、多次元のパラメータを持つ画像標本集合を収集する必要がある。そのため、これまで多くの計測装置が提案されてきた。例えば、平面形状などに対象が限定されるBRDFやBTFの計測に関しては、カメラ、光源、対象物体をモータ等で制御することに

より、視点および光源に関する4次元データを比較的容易に計測することが可能であり、多くの装置が開発されてきた²⁾¹⁶⁾。

一方で、3次元形状物体の完全な表面反射特性の計測は、以下のような点から、BRDFやBTFの計測ほど容易ではない。

- 画像と3次元データの位置合わせのためにキャリブレーションが必要であり、カメラや対象物体を自由に動かすと、位置ずれなど誤差の要因となる。

- 3次元物体、特に遺物などの貴重品の場合、任意の姿勢に固定することが難しく、自由に動かすことが難しい。そこで、当初は、光源・視点方向の4自由度パラメータすべてを変数とするデータではなく、その一部のみを変数として標準化をする装置が主に利用されてきた。Debvecらは、光源を円弧状に配列したアームを回転させることで、対象物体の光源方向(2自由度)に関して標準化する装置を作成した³⁾。同様に、PTM⁷⁾においては、ドーム状に光源を配置することで光源方向(2自由度)に関するデータの獲得を実現している。Woodらは、ロボットアームでカメラを天球上で自由に操作し、視点(2自由度)に関するデータを獲得する装置を利用して標準化を行った¹⁷⁾。

我々は3次元物体の、完全な視点・光源方向(4自由度)に関するデータを標準化するため、図1に示されるようなデータ収集プラットフォームを開発した⁵⁾。図に示されるように、本装置には2個の円弧状の部品がついており、それぞれCCDカメラとハロゲンランプを複数個取り付けることが可能である。CCDカメラを取り付ける円弧をカメラアームと呼び、ハロゲンランプを取り付ける円弧を光源アームと呼ぶことにする。これらのカメラと光源はPCにより制御される。円弧の中心付近には、計測対象物体を載せるための回転テーブルが取り付けられている。回転テーブルと光源アームは、同一の垂直な軸を中心に回転する。これらの回転も、PCによって制御される。回転テーブルによる角度変化と撮影するカメラの選択によって、視点の方位角と仰角を、また、光源アームの角度変化と点灯する光源の選択によって、光源の方位角と仰角を変えることができる。これらのパラメータを変えながら画像を標準化することで、4自由度パラメータについて見えのデータを獲得できる。

対象物体の見えの標準化と同時に、光切断法を利用して物体の形状計測が行われる。本装置には、ラインレーザ投光機が取り付けられ、事前の校正によってレーザのなす平面(レーザ平面)の位置が計測されている。図2のように対象物体にレーザ光を照射し、カメラでその反射を測定することで、三角測量の原理を用いて照射された点の3次元位置を計測できる。レーザ平面は、回転テーブルの回転軸を含むように調整する。回転テーブルを回しながら光切断法を適用することで、物体全体の表面上の点を計測することができる。

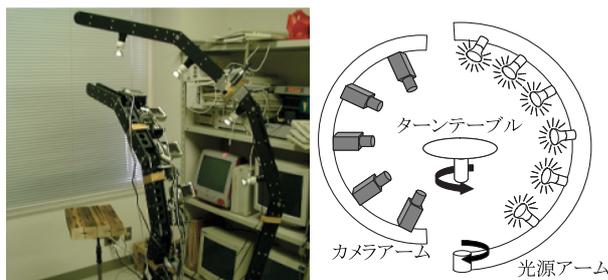


図1 画像標準化装置。
Image sampling platform.

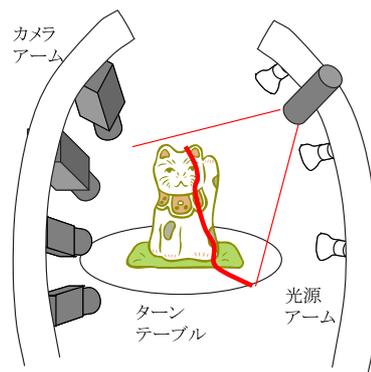


図2 光切断法による形状測定。
Shape measurement with light sectioning method.

計測された点の集合を接続することで、ポリゴンメッシュが構築される。構築されたポリゴンメッシュのポリゴン数は大きく、実時間描画には不向きであるので、メッシュを簡素化してポリゴンを減らす。

本装置の構成の利点として(1)複数のカメラおよび複数の光源について、機械的な回転制御なしに標準化を行えるため、一つのカメラと一つの光源を利用する装置に比べて効率的に標準化できること、(2)球状に光源やカメラを配置する装置に比べ、カメラ数や光源数を大幅に減らすことができ、装置そのものも簡素化できること、(3)回転テーブルと平面レーザを組み合わせることで、物体の形状測定を同時におこなうことができ、それにより事後のテクスチャと形状モデルの位置あわせの必要がなくなること、などを挙げることができる。

4. 見えのモデル化

4.1 光源による見えの変化の近似

物体の見えは、BTFと類似した方法で表現される。ただし、視点方向は標準化されたデータを再標準化せずにそのまま利用する。物体の各点での輝度は、

$$T(i_v, \mathbf{l}, s, t, c),$$

である。ここで、 i_v は視点の標準化位置を表す番号を、 \mathbf{l} は光源方向を、 (s, t) は画像上での座標を、 c は $\{0, 1, 2\}$ の値をとり、色の要素(R,G,B)を表す。

本論文では、物体の見えの輝度の、光源方向に依存する変化を、球面調和関数展開で表現し、視点方向に依存する

変化は、標本化された離散データをそのまま保持する。視点方向に関する補間手法としては、視点依存テクスチャマッピングの手法を利用する。

輝度の光源に依存した変化の近似は、各視点に対して別個に行われる。手順を以下に示す。最初に、画素位置および視点を固定したときの画素値の集合を、標本化された画像から抽出する。これらの値は、見えの光源依存変化を表す球面関数とみなすことができる。次に、これらの値を、物体座標系で表した光源方向をパラメータとして、キューブテクスチャマップに描画する（描画は、各光源方向を頂点とする球状の多角形を描画することで行う）スムーズシェーディングを利用することで、離散的な光源方向の標本位置の間を補間することができる。

球面関数を球面調和関数展開した式の係数は、球面積分によって得られる。球面積分は、キューブテクスチャマップで表された球面関数の画素値と、同様に表された基底関数の画素値と各画素に対応する立体角との積の、すべての画素についての和として近似できる。 T の光源依存変化を近似する関数は、計算された係数を使って以下のように表される。

$$\sum_{m=1}^M c_m(i_v, s, t, c) Y_m(\mathbf{l})$$

ここで Y_m は m 番目の球面調和関数基底を、 M は利用する基底の数を、 \mathbf{l} は光源方向を、 i_v は視点方向の番号を、 (s, t) は標本化された画像の座標を c は色の要素を表す。この近似は、すべての視点方向、画素位置、色要素について行われる。

P 次までの球面調和関数基底を利用する場合には、 $M = P^2$ となる。高次の球面調和関数は、高い空間周波数に対応するので、 P 次までの関数で近似することにより、光源依存変化は空間周波数において低域に制限される。鏡面反射が強くない、光沢のある物体では、通常 $P = 5$ 程度でよいことが、Sloan らによって報告されている¹⁴⁾。 P を不適切に小さくした場合、見えの光源依存変化の高周波成分が再現されなくなり、特に自己シャドーや、鏡面反射がぼけてしまう。本論文の実験では、 $P = 6, M = 36$ の条件でモデリングおよび描画を行った。

計算された球面調和関数基底の係数はベクトルとして表現され、PRT¹⁴⁾と同様に、主成分分析(PCA)によってさらに圧縮される。PCAに必要な計算量を減らすため、係数ベクトルの一部のみを標本化してPCAによって分析し、得られた主成分方向を基底ベクトルとしてすべてのベクトルを近似する。

PCAによって得られた、各係数ベクトルを近似する重みベクトルは、パラメータ (i_v, s, t) によって指定される。 c の色、 m の基底に対する k 番目の主成分ベクトルを $b_{k,m,c}$ ($1 \leq k \leq K$) とし、その重みベクトルを $W_k(i_v, s, t)$ とする。ここで K は、近似に使用する主成分の数である。視点

i_v 、画素位置 (s, t) に対する見えの光源依存変化は、以下の関数で近似される:

$$T(i_v, \mathbf{l}, s, t, c) \approx \sum_{m=1}^M \sum_{k=1}^K W_k(i_v, s, t) b_{k,m,c} Y_m(\mathbf{l}) \quad (1)$$

m に関する和を先に取り、 $L(\mathbf{l}, k, c) \equiv \sum_{m=1}^M b_{k,m,c} Y_m(\mathbf{l})$ とすると、

$$T(i_v, \mathbf{l}, s, t, c) \approx \sum_{k=1}^K W_k(i_v, s, t) L(\mathbf{l}, k, c) \quad (2)$$

となり、 $T(i_v, \mathbf{l}, s, t, c)$ を K 個の項の和によって表現できる。

4.2 視点による見えの変化の近似

1式の見えの近似は、連続な光源方向パラメータ \mathbf{l} を持つが、視点方向については離散的なパラメータに対してのみ定義されている。関数(1)の三個の視点方向の見えの重みつき平均をとることで、任意の視点方向について見えを近似する。これは、視点依存テクスチャマッピング(VDTM)⁴⁾と類似した手法である。

任意の視点方向について、視点方向の補間に用いる視点番号と、その重みを効率的に計算するために、これらの値を参照テーブルにコード化する。これらの参照テーブルをテクスチャとすることで、これらの情報をピクセルシェーダから参照することができる。これによって、各画素を描画する時に、対応する視点番号と重みを得ることができる。

標本化された視点方向の集合の各要素は、単位球面上の点と対応させることができる。これらの視点方向集合に対応する点を頂点とするメッシュで、単位球面を分割し、視点標本メッシュと呼ぶ。これにより、任意の視点方向 \mathbf{v} から、視点標本メッシュにおいて \mathbf{v} を囲む一つの三角形への写像が決まる。 \mathbf{v} が、三角形メッシュの辺上の点に写像される場合、 \mathbf{v} は隣り合う三角形の一つに写像されるとする。この写像によって、 \mathbf{v} に対応する三角形を近傍三角形と呼び、 $\Delta(\mathbf{v})$ と記述する。視点標本メッシュの各三角形の頂点は、それぞれの三角形に関する局所番号 $0, 1, 2$ で指定される。 $\Delta(\mathbf{v})$ の三個の頂点を、 \mathbf{v} の近傍視点標本と呼び、各頂点の局所番号の順に、 $V(\mathbf{v}; 0), V(\mathbf{v}; 1), V(\mathbf{v}; 2)$ と記述する。視点標本 λ の視点番号を $\iota(\lambda)$ と記述するとし、 $I(\mathbf{v}; n) \equiv \iota(V(\mathbf{v}; n))$ と定義する。これを \mathbf{v} から視点番号への写像と見なし、近傍視点番号関数と呼ぶ。同様に、 \mathbf{v} から、視点補間のための重みへの写像を、頂点の局所番号の順に、 $H(\mathbf{v}; 0), H(\mathbf{v}; 1), H(\mathbf{v}; 2)$ で表し、近傍視点重み関数と呼ぶ。

近傍視点重み関数 $H(\mathbf{v}; n)$ の定義には、いくつかの候補が考えられる。視点の補間に線形補間を用いる場合、 $H(\mathbf{v}; n)$ は、視点方向 \mathbf{v} と $\Delta(\mathbf{v})$ の交点の、 $\Delta(\mathbf{v})$ に対する重心座標として定義できる（ここで、 Δabc 上の点 p の Δabc に対する重心座標が、頂点の順序を a, b, c として (α, β, γ) であるとは、 $\Delta pbc, \Delta pca, \Delta pab$ の面積比が、 $\alpha : \beta : \gamma$ であることを意味する。） $\Delta(\mathbf{v})$ の頂点の順序を局所番号順として、

v の重心座標が $(B(v;0), B(v;1), B(v;2))$ で表される場合、 $H(v;n) \equiv B(v;n)$ とすればよい。本論文では、この定義を用いた。

$I(v;n)$ と $H(v;n)$ はキューブテクスチャマップとして表現される。これらのテクスチャは、視点標本メッシュを適切な頂点色で描画することで得られる。

5. プログラブルシェーダによる実時間描画

計測された物体の実時間描画について述べる。まず、見えのデータをビデオハードウェアにテクスチャとして読込む。

テクスチャとしてコード化されるデータは、 $W_k(i_v, s, t)$, $I(v;n)$, $H(v;n)$, $L(l, k, c)$ である。 $W_k(i_v, s, t)$ は通常の2次元テクスチャとして、 $L(l, k, c)$ は k 個のキューブテクスチャマップとして、 $I(v;n)$ と $H(v;n)$ は視線方向 v についてのキューブテクスチャマップとして表現する。

プログラブルシェーダのアルゴリズムは、頂点シェーダとピクセルシェーダとして記述される。本描画アルゴリズムの頂点シェーダは、世界座標系で表された光源および視点方向を、物体座標系での表現 v, l に変換する。また、形状モデルの3角メッシュの頂点位置を物体座標系からカメラ座標系に変換する。

ある視点での見えを表すテクスチャを形状モデルにマッピングするためのテクスチャ座標は、射影によって計算される。具体的には、表面上の各点を、その視点での標本化に使われたカメラの内部および外部パラメータで変換、射影し、得られた2次元画素位置をテクスチャ座標とする。

ビデオハードウェアのプログラブルシェーダを利用すると、テクスチャ座標の計算を効率的に行うことができる。ピクセルシェーダと頂点シェーダの計算負荷のバランスを考えると、頂点シェーダで射影の計算を行うことがより望ましいと考えられるが、その実現には困難な点がある。射影テクスチャマッピングの計算には近傍視点番号関数 $I(v_0;n)$ の参照が必要になるが、これは、テクスチャにコード化されている。多くのビデオハードウェアの頂点シェーダは、テクスチャにアクセスできないため、これを参照できない。このようなハードウェアでは、頂点シェーダで射影計算を行うことはできない。ただし現在では、頂点シェーダによる、テクスチャアクセスは、標準的な機能になりつつあるため、この問題は解決しつつあると言える。

ピクセルシェーダを利用する場合には、各視点に対するカメラパラメータもテクスチャにコード化して格納する。頂点シェーダは、その3次元位置を補間してピクセルシェーダに渡し、ピクセルシェーダは、受け取った3次元位置と、テクスチャから読み取ったカメラパラメータから射影計算を行えばよい。本論文では、ピクセルシェーダによる射影計算を実装した。

もっと簡便な計算方法として、見えの再現を行う視点方向として、対象物全体にわたって固定した視点方向 v_0 を利用する手法が考えられる。この場合、近傍視点番号 $I(v_0;n)$ 、

および近傍視点重み $H(v_0;n)$ が、対象物全体で同一となる。射影計算は、3個の近傍視点についてのみ行われるので、これらの視点のカメラパラメータのみをビデオハードウェアのレジスタを通して、頂点シェーダに渡す。この方法は、対象物のすべての点において視線方向が一定であることを仮定している。この仮定は、視点が物体から離れている場合にはよい近似になる。視点が物体に近づいた場合、形状は透視投影されるのに対し、テクスチャは中心位置での視線方向にもとづいて生成されるので、周辺部で見えの誤差が生じる。本論文では、ピクセルシェーダで射影計算を行う実装に加え、この手法による実装も行った。我々の実験では、この仮定による誤差は目立ちにくく、気づきにくい程度であった。

ピクセルシェーダの処理を以下に述べる。 PIX はピクセル値を、 IV は近傍視点の番号を、 WV は近傍視点重み関数の値を格納し、 WL は一時変数である。ステップ2と3は、 $I(v;n)$ と $H(v;n)$ の参照がアプリケーション (CPU) によって行われる場合は省略される。

Step 1 $PIX[c] \leftarrow 0, c = 0, 1, 2$

Step 2 $IV[n] \leftarrow I(v;n), n = 0, 1, 2.$

Step 3 $WV[n] \leftarrow H(v;n), n = 0, 1, 2$

Step 4 $k = 0, 1, \dots, K-1$ について以下を繰り返す

Step 4.1 $WL[k] \leftarrow \sum_{n=0}^2 WV[n] * W_k(IV[n], s, t)$

Step 4.2 $PIX[c] \leftarrow PIX[c] + WL[k] * L(l, k, c), c = 0, 1, 2$

Step 5 $PIX[c]$ を画素値として出力

本実装では、基底球面関数 $L(l, k, c) \equiv \sum_{m=1}^M b_{k,m,c} Y_m(l)$ は、テクスチャとして格納される。これにより、平行光源ばかりでなく、点光源による照明にも、効率を犠牲にすることなく対応できる。環境マップで表された放射照度によって光源を表す場合には、環境マップの内容を基底球面関数で表現し、その係数を Step 4.2 において $L(l, k, c)$ の代わりに利用するように実装を変更すればよい。

6. 実験

本論文で提案した手法の有効性を確認するため、古代の銅鏡を計測し実時間描画を行った。図3が、銅鏡を撮影した画像である。標本化間隔は、経度方向は、視点、光源共に 18° 間隔の20標本とした。緯度方向に関しては、カメラ5台を上半球に配置し標本化を行ったため、視点の標本化間隔は $15 \sim 20^\circ$ である。ライトは全天球をカバーするように8つのライトを設置したため、緯度方向の光源の標本化間隔は 20° である。この設定により、我々は、 $20 \times 20 \times 5 \times 5 = 10000$ 枚の画像を標本化した。画像の解像度は 720×480 である。計測には約3時間を要した。

同時に3次元形状を、3節で説明した光切断法で計測した。計測した3次元データは、実時間描画可能なようにポリゴンメッシュを簡素化した。本実験においては5000ポリゴン程度にした。図4が、簡素化後のポリゴンメッシュ



図 3 標本化された銅鏡.
Sampled ancient mirror.

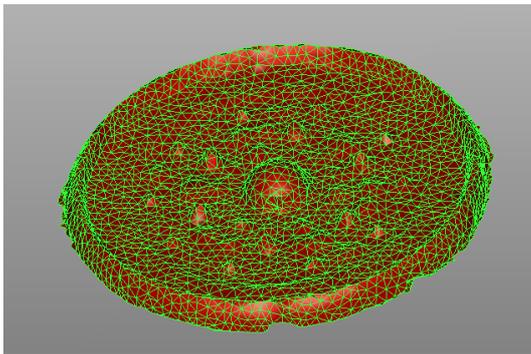


図 4 獲得されたポリゴンメッシュ.
Obtained polygon mesh.

を表したものである。

上記の標本化パラメータに従って計測した画像集合から、光源がカメラの陰になり正しく照明されなかったものなど、入力データとして利用できない画像を除いたものを、以降の解析に利用した。その結果、利用可能な視点方向と光源方向の標本数はそれぞれ、76 と 100 であったため、合計 $76 \times 100 = 7600$ 画像を実際の解析に用いた。

選ばれた画像集合から、視点および画素位置ごとに、光源方向に関する輝度の変化を球面調和関数で近似した。球面関数の描画のためのキューブマップの解像度は、各面において 32×32 とした。利用した球面調和関数基底の数 M は $6^2 = 36$ であり、すべてのデータの係数を計算するために要した時間は、CPU が Intel 社 Pentium4(2.0GHz) 主記憶装置容量 2.0GB の PC で、約 11 時間であった。

PCA の計算においては、計算量削減およびメモリの効率的利用を考慮し、全体の 1% のデータを抽出し、それらの主成分ベクトルを計算した。主成分分析に要した時間は、上記のシステムで約 2 分程度であった。PCA により得られる、光源に関する重み集合 W^* は、画像として表現される。図 5 が、重みを表現した画像である。それぞれの画像は、第 2 主成分および第 4 主成分を表している。ただし、それぞれの重み画像は、画像のダイナミックレンジの有効利用のため、最大の絶対値で正規化されており、実際の重みをそのまま表してはいない。実際の描画において利用した主成分の数 K は 12 とした。



(第 2 成分) (第 4 成分)
図 5 PCA によって計算された重み.
Weights obtained by PCA.

我々は、提案する手法による実時間描画を確認するため、描画アルゴリズムの実装を行った。ビデオハードウェアとしては、256MB のテクスチャメモリーを搭載する、nVidia 社の GeForceFX 5950 Ultra を用いた。テクスチャ座標の計算については、5 節で述べた 2 種類の方法を実装した。一つは、射影計算をピクセルシェーダで画素ごとに行う手法で、もう一つは、テクスチャマッピングされる見えの計算に使用する視線方向を、すべての点で同一であると仮定する方法である。我々は以後、前者を画素毎視点評価手法、後者を物体毎視点評価手法と呼ぶ。物体毎視点評価手法は、見えの生成に関する仮定のため誤差を含むが、テクスチャを生成するための近傍視点の番号と重みが形状モデル全体で一定であるため、描画効率は高い。画素毎視点評価手法においては、全データをビデオハードウェアのテクスチャメモリー上に載せることができなかつたため、実装に際しては、各視点の見えの画像サイズを、縦横それぞれ半分の 320×240 として実装した。

図 6 に、画素毎視点評価手法による描画結果を示す。白い球が光源を示している。点光源が左方向、正面、遠方、近接、と異なった位置にある時に、それぞれ正しく対象物体が描画されることが分かる。

図 7 は、物体毎視点評価法により異なった視点位置から描画した例である。テクスチャは視点位置によって生成されるため、銅鏡の凹凸の見えが視点によって変化する様子が正しく描画されていることが分かる。この凹凸は、簡素化されたポリゴンメッシュ(図 4)では、形状として表現されておらず、見えの画像からテクスチャとして再現されたものである。

描画速度は、対象物体の占める領域の大きさに依存する。物体が大きな面積を占める場合、ピクセルシェーダのプログラムがそれだけ多く実行されるので、描画に時間がかかる。図 8 に例を示す。この例においては、ウィンドウのクライアント領域の大きさは 640×480 である。図の例において計測された描画速度は、画素毎視点評価手法では 23fps、物体毎視点評価手法では 60fps であった。また、それぞれの手法で描画された図を見比べると、陰影などに若干の差は認められるものの、物体毎視点評価手法でも充分な品質で描画できていることがわかり、計算を簡略化したメリットの大きいことが分かる。



(左側の点光源による照明.)



(手前の点光源による照明.)



(遠くの点光源による照明.)



(近くの点光源による照明.)

図 6 異なる光源方向での描画結果 .

Rendered images with different light positions.

本論文で提案した見えに基づくモデルは、実時間描画のための BTF のモデルとしても利用できる。何種類かの BTF を標準化し、球面に貼り付けて描画した例を図 9 に示す。それぞれの画像において、BTF が視点および光源に依存して正しく描画されていることがわかる。

7. む す び

本論文では、3次元形状物体の表面特性を獲得し、実時間描画に適したモデルとして表現する手法を提案した。これを実現するため、我々は、実物体の見えを、4自由度の視点



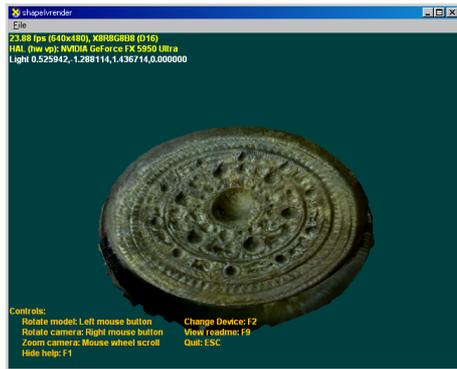
図 7 異なる視点からの描画結果 .

Rendered images with different viewpoints.

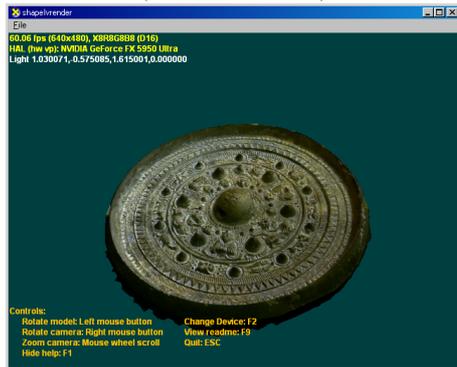
および光源方向のパラメータを用いてすべて標準化し、各視点における、2次元の光源パラメータ空間を球面調和関数基底によって表現した。また、視点方向に関しては、標準化した視点間を補間して、投影テクスチャマッピングにより描画する手法を提案した。提案した手法を、ピクセルシェーダ機能を持ったビデオハードウェアを用いて実装し、実際に計測した3次元物体を実時間で描画可能であることを確認した。

〔 文 献 〕

- 1) W.-C. Chen, J.-Y. Bouguet, M. H. Chu and R. Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. In J. Hughes, *SIGGRAPH 2002 Conference Proceedings*, Annual Conference Series, pp. 447–456, (2002).
- 2) K. Dana, B. V. Ginneken, S. Nayar and J. Koenderink. Reflectance and texture of real-world surfaces, *ACM Transactions on Graphics*, pp.1–34, (1999).
- 3) P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin and M. Sagar. Acquiring the reflectance field of a human face. In K. Akeley, *Siggraph 2000, Computer Graphics Proceedings*, pp. 145–156, (2000).
- 4) P. E. Debevec, G. Borshukov and Y. Yu. Efficient view-dependent image-based rendering with projective texture-mapping. In *9th Eurographics Rendering Workshop*, pp. 105–116, (1998).
- 5) R. Furukawa, H. Kawasaki, R. Kawatsu, D. Miyazaki, K. Ikeuchi and M. Sakauchi. Appearance based object modeling using texture database: Acquisition, compression and rendering. In *Rendering Techniques 2002 (Eurographics Workshop Proceedings)*, pp. 257–265 & 333, (2002).
- 6) J. Kautz and M. D. McCool. Interactive rendering with arbitrary brdfs using separable approximations. In *Rendering Techniques (Eurographics Workshop Proceedings)*, pp. 247–260, (1999).
- 7) T. Malzbender, D. Gelb and H. Wolters. Polynomial texture maps, *Proceedings of SIGGRAPH 2001*, pp.519–528, (2001).
- 8) W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler and L. McMillan. Image-based 3d photography using opacity hulls, *acm Transactions on Graphics*, pp.427–437, (2002).



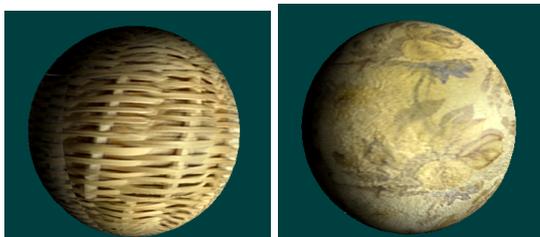
(画素毎視点評価手法)



(物体毎視点評価手法)

図 8 描画速度の測定例: ウィンドウのクライアント領域は 640×480 , 描画速度は画素毎視点評価手法で約 23 fps, 物体毎視点評価手法で約 60 fps.

Examples for measuring the rendering speed: the client areas of the windows were 640×480 , the frame rates were approximately 23 fps for the per-pixel-view implementation and 60 fps for the per-object-view implementation.



(木を編んだ素材)

(カーペット)

図 9 測定された BTF のモデル化と描画.

Modeling and rendering of measured BTFs.

- 9) M. D. McCool, J. Ang and A. Ahmad, Homomorphic factorization of brdfs for high-performance rendering, *Proceedings of SIGGRAPH 2001*, pp.171–178, (2001).
- 10) K. Nishino, Y. Sato and K. Ikeuchi. Eigen-texture method: Appearance compression based on 3D model. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, pp. 618–624, (1999).
- 11) R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 497–500, (2001).
- 12) R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In E. Fiume, *SIGGRAPH 2001, Computer Graphics Proceedings*, pp. 117–128, (2001).
- 13) M. Sattler, R. Sarlette and R. Klein. Efficient and realistic visualization of cloth. In *EGSR '03: Proceedings of the 14th Eurographics Symposium on Rendering*, pp. 167–177, (2003).
- 14) P.-P. Sloan, J. Kautz and J. Snyder, Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments, *ACM Transactions on Graphics (TOG)*, pp.527–536, (2002).

- 15) P.-P. Sloan, X. Liu, H.-Y. Shum and J. Snyder, Bi-scale radiance transfer, *ACM Trans. Graph.*, vol.22, no.3, pp.370–375, (2003).
- 16) F. Suykens, K. v Berge, A. Lagae and P. Dutre. Interactive rendering with bidirectional texture functions. In *Proceedings of Eurographics 2003*, pp. 463–472, (2003).
- 17) D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin and W. Stuetzle. Surface light fields for 3D photography. In K. Akeley, *Siggraph 2000, Computer Graphics Proceedings*, pp. 287–296, (2000).

ふるかわ りょう
古川 亮 1993年, 京都大学工学部卒業. 1996年, 奈良先端科学技術大学院大学博士後期課程修了. 現在, 広島市立大学情報科学部助手. コンピュータ・ビジョン, コンピュータ・グラフィックに関する研究に従事. 博士(工学).

かわさき ひろし
川崎 洋 1994年, 京都大学工学部卒業. 2003年, 東京大学大学院工学系研究科博士後期課程了. 現在, 埼玉大学工学部情報システム工学科助教授. 主として広域空間のモデリング, 3次元ビジョン, テクスチャ解析に関する研究に従事. 電子情報通信学会, 情報処理学会, IEEE, 各会員. 博士(工学).

なかむら やすあき
中村 泰明 1979年, 大阪大学大学院基礎工学研究科物理系専攻前期課程修了. 同年, 三菱電機株式会社入社. 中央研究所勤務. 画像処理, パターン認識, 空間データ構造, バーチャルリアリティ技術などの研究開発に従事. 1994年, 広島市立大学情報科学部教授, 現在に至る. 工学博士.

おおさわ ゆたか
大澤 裕 1976年, 信州大学工学部卒業. 1978年, 信州大学大学院工学系研究科修士了. 東京大学生産技術研究所助手, 埼玉大学工学部助手, 助教授などを経て, 1998年より, 埼玉大学工学部情報システム工学科教授. 地理情報システム, 時空間情報システムの研究に従事. 信学会論文賞受賞. 情報処理学会, 日本 GIS 学界, ACM, 各会員. 工学博士. 正会員.