

視点依存の微小面を用いた複雑な表面形状を持つ物体の表示手法

山崎俊太郎[†] 佐川 立昌[†] 川崎 洋[†] 池内 克史[†]
坂内 正夫[†]

Rendering Intricately-shaped Objects using View-dependent Microfacets

Shuntaro YAMAZAKI[†], Ryusuke SAGAWA[†], Hiroshi KAWASAKI[†], Katsushi IKEUCHI[†], and Masao SAKAUCHI[†]

あらまし 仮想空間内でコンピュータグラフィックスを用いて実物体を表示するには、対象物の幾何情報とテクスチャ情報を用いて描画する方法が一般的である。しかし物体が複雑な細部構造を持つ場合、幾何モデルを正確に計測することが出来ないため、現実感を失うことなくこれらの物体を描画することが難しい。そこで描画視点に依存して視線と面が常に直交するように微小面を層状に発生させ、さらに各面に視点依存テクスチャマッピングを行うことで、より現実感の高い描画を行う手法を提案する。これらの処理はプログラム可能グラフィックスハードウェアを使って通常の PC で効率的に実装することが可能である。本手法は視点依存の詳細度制御が可能であり、複雑な物体や風景を対話的な速度で描画できることを示す。

キーワード 視点依存微小面, 視点依存テクスチャマッピング, 射影テクスチャマッピング, 詳細度制御

1. はじめに

実世界の物体や風景を計測し、コンピュータグラフィックスとして再現することは、仮想現実感、拡張現実感といった現実感モデルを構築する上で不可欠な処理である。しかし、対象物が複雑な細部構造を持つ場合、それらを正確に計測、表現することは困難であり、また描画処理に要するコストも高くなる。

コンピュータグラフィックスで実物体を表示する際に最も広く使われる手法は、物体表面を境界面表現し、表面の属性をテクスチャで与えて表示する手法 (Model-Based Rendering) である。この方法では、物体表面の幾何形状を精度良く計測するために、レーザーレンジファインダや、パターン光投影法等が用いられる。しかし幾何形状の取得は必ずしも容易ではなく、例えば毛髪や樹木のような細かく入り組んだ形状を持つ物体の場合、計測精度が不足するため、得られる形状は不正確、不完全である。また計測精度が撮影対象の表面特性に影響されやすく、ノイズを多く含む場合がある。計測物体の表面特性の影響を比較的受けにくい計測手法として、シルエット投影法 [4], [6] があるが、形状が複雑な

場合にはシルエット抽出が困難であり、また多くの場合得られる結果は不正確である。計測によって得られた不正確な幾何形状を信号処理によって平滑化 [10] する研究もあるが、処理に必要なパラメータの選択は一般に困難である。また、仮に複雑な細部構造の幾何形状を完全に取得、もしくは復元できたとしても、それを表現するために必要なデータ量は膨大であり、表示を目的とした場合にこれらを全て保持することは現実的ではない。

一方で、3次元復元を行わずに、カメラで撮影して得られる2次元画像から直接、結果画像を合成する表示手法 (Image-Based Rendering) も提案されている [7]。この手法を用いると、入力画像を増やすことにより、原理的にはあらゆる対象物体を描画することが可能であるという利点がある。しかし、完全に幾何形状を持たない手法では、しばしば膨大な量のデータが必要になり、描画できる視点位置や対象物の大きさも限定される。また、計測した物体を他の環境下で描画したり、入力画像中になく視点から表示することは難しく、仮想空間における利用は著しく限定される。

Image-Based Rendering の欠点を補う方法として大まかな幾何形状を利用することにより表示精度を向上させる方法も提案されている。この手法では物体

[†] 東京大学, 東京都目黒区駒場 4-6-1

The University of Tokyo, 4-6-1 Komaba Meguro-ku, Tokyo, 153-8505 JAPAN

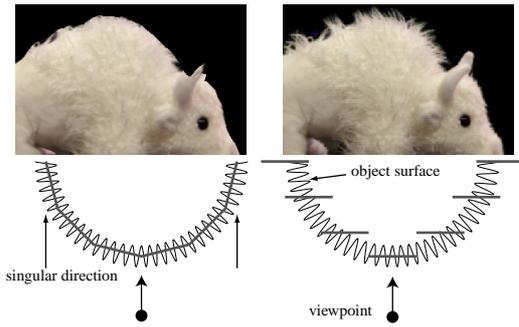


図1 幾何形状を微小面近似した場合の結果の比較. 左図: 固定微小面で近似する場合. 右図: 視点依存微小面で近似する場合.

Fig.1 Comparison of the images rendered using microfacets. Left: fixed facets. Right: view-dependent microfacets.

の概形を表す幾何形状を利用し, 細部はテクスチャ画像の合成により表示する. これに分類される手法として View-dependent texture mapping 法 [3], Unstructured Lumigraph 法 [1], Opacity-Hull 法 [5] などが挙げられ, いずれも視点非依存 (固定) の近似表面形状に対して視点依存テクスチャマッピングを行うことで描画を行う. しかしこれらの方法では, 近似した表面と実際の表面のいずれの方向と視線のなす角が大きくなる描画視点において, 形状の近似誤差が描画結果に大きく影響し, 現実感が著しく失われる問題がある (図 1, 左図).

我々は, 視線と常に垂直になるように視点に依存して生成される微小平面の集合を用いて幾何形状を近似し, 視点依存のテクスチャマップを行うことで描画を行う手法を提案する (図 1, 右図). 提案手法は, 画像を用いた表示手法の特徴として, 任意の対象に適用可能であると同時に, 利用できる幾何形状を最大限に活用することにより, 入力画像中に含まれていない視点での描画を精度よく行うことができる. また幾何形状の近似を多重解像度化することが容易であり, 視点依存の詳細度制御が可能である.

2. 視点依存微小面によるモデル生成と描画

2.1 計測データの取得

本手法では, センサーによる計測やステレオ視などによって得られた物体の粗い幾何形状と, 複数視点から撮影されたカラー画像を入力とする. 一般に, 幾何形状が正確であるほど, またカラー画像の撮影位置の数が多ければ最終的な描画結果の画質は向上する. 幾何形状とカラー画像の撮影位置の相対関係, およびカメラ

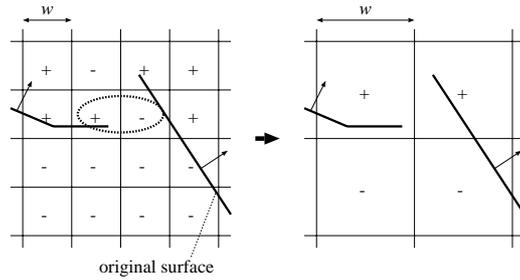


図2 最適なサンプリング幅の決定.
Fig.2 Determination of optimal sampling width.

パラメータは既知とする.

2.2 幾何形状のボクセル近似

まず, 与えられた幾何形状をボリュウム空間で再サンプリングし, テクスチャを利用した描画処理に適したデータ構造に変換する. そのために, 対象物体を完全に含むボリュウム空間を考え, 各ボクセルにおいて入力幾何形状と交差判定を行って二値ボリュウムを生成する. 二値化処理では通常, 物体表面が交差するボクセルに 1 を, 交差しないボクセルには 0 を割り当てるが, 入力形状が不完全な場合には, 表面に位置するはずのボクセルが形状と交差せずに, 穴が発生する可能性がある. そこで本手法ではこれを次のような 2 段階処理で行うことで, より頑健に表面位置を決定する.

- (1) 表面形状を符号付距離ボリュウムに変換する.
- (2) 距離の絶対値が一定値以下のボクセルを表面と交差するボクセルとしてボリュウムを二値化する.

符号付距離ボリュウムとは, 各ボクセルに対しボクセルの中心から最も近い面への距離を, 最近点が面の裏側の場合には負の符号を, 表側の場合には正の符号をつけて割り当てたボリュウムである. ただし入力幾何形状が多量のノイズを含む場合にはこの計算は安定しないため, consensus surface 法 [11] を用いて符号付距離の計算を行う.

また再サンプリングするボリュウムの一辺の分割数 L は次のように決定する.

- (i) L を十分に大きい値 L_{max} に設定する.
- (ii) 表面形状を符号付距離変換して $L \times L \times L$ ボクセルからなるボリュウムを生成する.
- (iii) 得られたボリュウム中のボクセルで, 隣接するボクセルの符号付距離との差の絶対値が αw よりも大きいもの数 N_i を調べる. ここで w はボクセル幅, α は許容度をあらわす係数 (≥ 1) である.

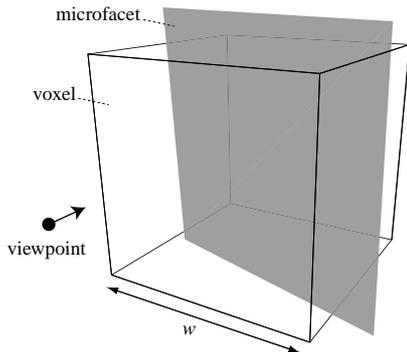


図3 物体表面と交差するボクセル内に微小面を発生させる。

Fig.3 A microfacet is a slice which intersects the voxel and is perpendicular to the viewing direction.

(iv) 閾値 $t \in [0, 1]$ を与え, 条件式

$$\frac{N_i}{L^3} \leq t \quad (1)$$

を満たしていなければ L を $\frac{1}{2}$ 倍して処理 (ii) に戻る。満たしていれば終了。

ここで求められたボクセル分割数を L_{opt} とする。式 (1) 左辺は, 表面形状の不連続性のために信用できない値を持つボクセルの割合と考えることが出来る。実験では $L_{max} = 256$ とした。 α, t は入力 of 幾何形状の精度に応じて変更すべきであり, 実験で用いたデータでは $\alpha = 1.5$, $t = 0.01$ を使用した。

図2に処理の一例を示す。左図に示す楕円で囲まれた2つのボクセルの周辺で入力 of 表面形状に不連続性があり, 隣接する符号付距離の差が αw より大きくなる。右図のようにボクセルの解像度を下げることによりこのような不連続性を避けることが出来る。

2.3 微小面の生成

生成された二値ボリューム中で, 表面と交差するボクセルの位置に微小面を発生させて描画を行う。視点からボクセル中心に向かう方向を視線方向とするととき, 微小面を視線方向と垂直になるように, 視線に対して一定の間隔で生成する (図3)。即ち, 微小面の発生間隔を l とするとき, 視点から nl (n は整数) の位置に, 表面ボクセルと交差する場合に限り微小面を生成する。微小面は, 1つのボクセルに対して少なくとも1つ以上生成される。発生間隔 l は最終的な描画結果の画質に大きく影響するため, 第3.1節で詳しく考察する。任意の方向から見たときに微小面がボクセル全体を被覆するように, ボクセルの幅を w とする時, 微小面は一辺の



図4 入力画像 (左図), 対応する距離画像 (中央図), および距離に応じたテクスチャの切り抜きの例 (右図)。

Fig.4 Left: One of the color images. Center: Range image from the same viewpoint. Right: Clipped result of the range image.

長さが $\sqrt{3}w$ の正方形とする。

2.4 テクスチャ画像

微小面は入力 of カメラ画像を射影テクスチャマッピング (Projective texture mapping) [9] して描画される。カメラが複数ある場合には, 視線方向とカメラの撮影方向のなす角 ϕ が小さいカメラから順に, カメラ画像中に微小面の占める領域が完全に含まれるものを選択し, 画像を投影する。この処理はボクセル毎に行い, ボクセルに対する視線としては, 視点からボクセル中心に向かうベクトルを用いる。

選択されるカメラは視点の変更とともに描画時に動的に決定されるため, 特にカメラの撮影間隔が疎な場合, 視点移動の際にカメラ画像が切り替わることによる画像の変化が大きく, これによって描画結果の現実感が損なわれる。そこで先の ϕ を用いてカメラ画像を近いほうから順に複数枚選択し, それらを角度に応じて内挿し, 補間によって得られた画像をテクスチャとして投影することで, カメラ画像が切り替わる際の描画を滑らかに行うことができる。

2.5 距離情報を用いたテクスチャの切り抜き

微小面に対するテクスチャ画像の選択は, 視線方向とカメラの撮影方向から決定される。そのため, 複数の微小面で同一のテクスチャ画像が選択される可能性があり, これによりテクスチャ上の一つの画素が複数回描画されてしまう可能性がある。テクスチャの各画素は対象の一つの点から得られた情報であるから, これは描画時に同一の物体が複数現れる現象 (double imaging) として観測され, 描画結果の現実感を損なう。

そこで, カラー画像中の各画素に対してあらかじめ描画対象までの距離を持たせておき, 各画素の距離と描画対象の微小面の距離を比較して, それらが一定の閾値内に含まれる場合のみ表示することでこの問題を

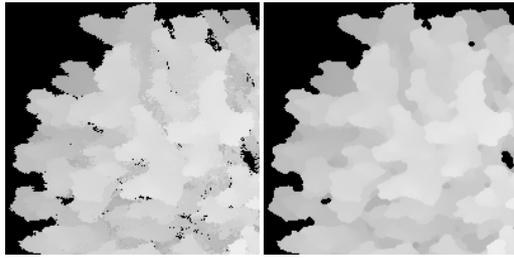


図5 モルフォロジーフィルタを使った距離画像の平滑化.
Fig. 5 The holes in a range image can be removed by applying a morphological filter. Left: Original range image. Right: Filtered range image.

解決する. 図4にその一例を示す. カメラ画像(左図)の各画素に, 色情報の他に対象物体までの距離を記録しておく(中央図), 描画時にその値を微小面の位置と比較し, 距離が一致する領域だけを切り抜いて表示する(右図).

微小面の発生間隔を l , 微小面上の点のカメラ方向の奥行きを D , 画像に割り当てられている奥行きを d とする時, テクスチャの切り抜きは描画面上の画素単位で次の関係式に従って行う.

$$\begin{cases} |d - D| < \frac{l}{2} & \rightarrow \text{画素を微小面上に描画} \\ \text{それ以外} & \rightarrow \text{画素を破棄} \end{cases} \quad (2)$$

画像に割り当てられる距離情報は, 入力として与えられる幾何形状を画像に射影することで生成することが可能であり, 表示の前処理としてあらかじめ行っておく. ただし入力の形状は必ずしも完全ではなく, 射影した結果の距離画像には多くのノイズが含まれている可能性がある. 特に距離画像に穴がある場合, 描画結果にも同様の穴が観測されて現実感を損なう. そこで前処理として距離画像にモルフォロジーフィルタをかけて, 細かい穴は除去しておく(図5).

2.6 実装

視点依存の微小面を用いた描画処理は, ポリゴンに射影テクスチャマッピングを行うことで実現でき, 式(2)で示したテクスチャ処理をプログラム可能なテクスチャ合成ハードウェアを用いて効率的に実装することで, 高速に表示することが可能である. ここでは nVidia 社製の GeForce3, またはそれ以降のグラフィクスカードと OpenGL グラフィクスライブラリで利用可能なテクスチャ合成機能である Texture shader (GL_NV_texture_shader) 拡張, 及び Register combiners (GL_NV_register_combiners) 拡張を利用した実装

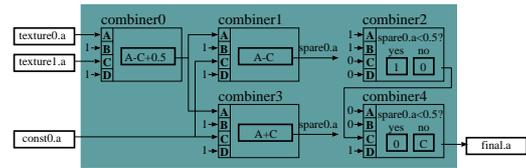


図6 テクスチャの切り抜き処理を行うための Register combiners の設定.
Fig. 6 Register combiners setup for texture clipping by substituting alpha value.

を示す.

各微小面は1つの正方形ポリゴンで表現され, それらは視点から遠いほうから順に, 射影テクスチャマップと α 混合を有効にして描画される. 各ポリゴンに対して第2.4節で述べたように複数のカメラ画像を選択し, マルチテクスチャマッピングを行う. 各画像のマッピングには, それぞれ2つのテクスチャユニット(ユニット0, ユニット1とする)を利用する. まずユニット0から, 撮影によって得られたカラー画素情報 C_{RGB} を RGB チャンネルに, 幾何形状を射影して得られた距離 d を α チャンネルに割り当てた, RGBz 画像を読み込む. ユニット1にはテクスチャ画像を割り当てず, テクスチャ座標として微小面の4つの頂点のカメラ方向への距離 D を割り当て, Texture shader の Pass through 関数 (GL_PASS_THROUGH_NV) を使いそれらの座標値を線形補間したものをテクスチャとして利用する.

テクスチャ合成ハードウェアの内部では, ユニット0の α チャンネルから d を, ユニット1の α チャンネルから D を読み込み, 式(2)の比較処理に従って, 描画される画素の不透明度 (α 値) を 1.0, 破棄される画素の不透明度を 0 に変更し, その不透明度で C_{RGB} をポリゴン上に描画する. この処理を行うテクスチャ演算は, 図6のように Register combiners を設定することにより実現可能である. なお式(2)の比較処理に必要な値 $\frac{l}{2}$ はあらかじめ定数レジスタの α チャンネルに設定している.

第2.4節で述べたカメラ画像の混合を行うためには, さらに別のテクスチャユニットの組(ユニット2, ユニット3等)を利用して別の RGBz 画像に関しても同様の処理を行い, Register combiners の最終段階で両者を内挿補間する. その際の補間係数 ϕ もあらかじめ定数レジスタに設定しておく.

3. 誤差の評価に基づくパラメータ設定

3.1 微小面近似による誤差の補正

近似幾何形状に入力画像を投影する表示手法では一

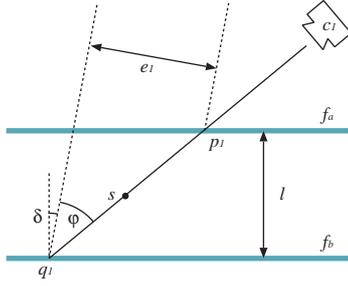


図7 カメラ画像が2つの異なる微小面に射影されるときに観測される不連続性。
Fig. 7 Visual discontinuity observed when a single camera image is projected onto two separate microfacets.

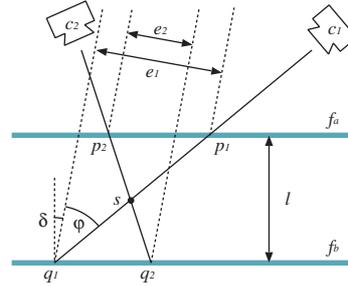


図8 複数のカメラ画像が2つの微小面に射影されるときに観測される不連続性の比較。
Fig. 8 Comparison of visual discontinuities observed when two divergent camera images are projected onto two microfacets.

般に、描画視点が入力のカメラ位置にある場合は結果画像は入力画像と一致する。しかし幾何形状の精度が入力画像の撮影密度に対して不十分である場合、視点がカメラ位置から外れるに従って表示上の誤差が生じる。これは入力データの計測精度、密度の不足に起因する根本的な問題であり [2]、ここでは触れないものとする。

これとは別に、提案手法では不連続な微小面集合で幾何形状を近似しているため、結果画像に不連続性が観測される可能性がある。また視点依存で幾何形状が変化するため、その誤差は視点によって変化する。そこで、微小面集合で幾何形状を近似することに起因する描画画像の不連続性を評価し、その誤差が視点によらず一定範囲内に収まるようにパラメータ設定を行う方法について考察する。

図7のように、隣り合う2つの微小面 f_a, f_b に対してカメラ c_1 が1つ選択されている状況を考える。問題を簡単にするため、微小面のサイズが視点までの距離と比べて十分に小さいものとし、従って平行投影を仮定する。微小面の間隔を l 、視線方向とカメラ方向のなす角を ϕ 、視線方向と微小面の鉛直方向のなす角を δ とし、 $\phi, \delta \in (-\pi/2, \pi/2)$ を仮定する。

まず、表面と交差する各ボクセルに少なくとも一つの微小面を発生させるためには、カメラ方向の微小面間隔がボクセル幅 w 以下、即ち

$$\frac{l}{\cos(\phi + \delta)} \leq w \quad (3)$$

である必要がある。

次に、 c_1 で撮影された画像中の画素で、 f_a, f_b の間の中間の位置 s に対応する点を選び、それが微小面に射影される点をそれぞれ p_1, q_1 とする。このとき描画視

点で観測される $\overline{p_1q_1}$ の大きさを e_1 とすると

$$e_1 = l \frac{\sin\phi}{\cos(\phi + \delta)} \quad (4)$$

となる。この値は c_1 の画像中のひとつの点が複数の微小面にまたがってテクスチャマップされる際に生じる不連続性を表し、描画の際に観測されるテクスチャのずれに相当する。

式(4)より、入力画像を計測するカメラがひとつの場合には、任意の視点で e_1 を最小にするのは $\cos(\phi + \delta) = 1$ のとき、即ち $\delta = -\phi$ のときである。ただし実際には入力のカメラがひとつの場合には誤差は非常に大きくなるため、現実的な精度で表示を行うためには複数のカメラ画像を入力とする必要がある。この場合、結果画像の合成に利用される入力画像集合は描画視点ごとに変化するの、個々の画像に対する誤差が最小になるよりも、選択されているカメラ画像間で、この不連続性が均等に分布するほうが望ましい。

そこで図8のように、描画視点の変化に伴い選択されるカメラが c_1 から c_2 に変化する瞬間を考える。記号の意味は図7と同様とし、 c_2 の画像中で s が f_a, f_b に射影される点をそれぞれ p_2, q_2 とする。カメラ選択は視線方向とカメラ方向の比較によって決定されるとすると、視線方向とカメラ方向のなす角は c_1, c_2 共に ϕ であり、従ってカメラ c_2 の画像に関して観測されるテクスチャの不連続性 $e_2 = \overline{p_2q_2}$ は

$$e_2 = l \frac{\sin\phi}{\cos(\phi - \delta)} \quad (5)$$

である。このときカメラ間のテクスチャの不連続性の差の絶対値は

$$|e_1 - e_2| = \left| l \frac{\sin\phi}{\cos(\phi + \delta)} - l \frac{\sin\phi}{\cos(\phi - \delta)} \right|$$

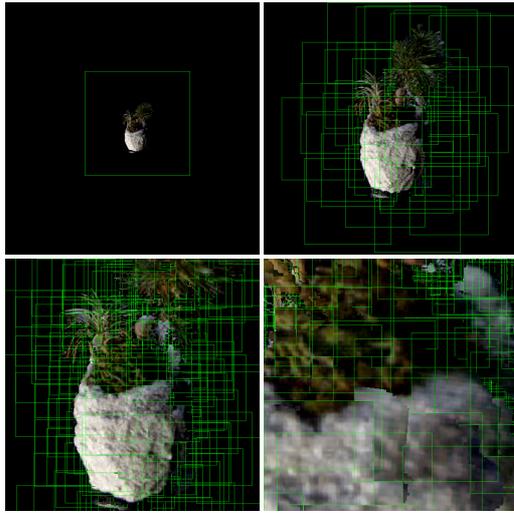


図9 視点移動に応じた微小面生成の詳細度制御。
Fig. 9 Level-of-detail control of microfacet generation.

$$= 2l \sin^2 \phi \left| \frac{\sin \delta}{\cos(\phi + \delta) \cos(\phi - \delta)} \right| \quad (6)$$

となり、任意の ϕ に関して c_1, c_2 における不連続性の差が最小値 0 をとるのは

$$\delta = 0 \quad (7)$$

のときである。即ち、微小面を常に視線と垂直に保つことにより、描画視点が移動する際のテクスチャの不連続性の変化を最小にすることができる。また、このときカメラ選択において入れ替わる各カメラの画像中の不連続性 e は

$$e = l \tan \phi \quad (8)$$

となり、視点に依存して変化することが分かる。許容される不連続性の最大値を E とすると、 e がこの値より大きくなならない条件は

$$l \tan \phi \leq E \quad (9)$$

である。式 (3)、式 (7)、式 (9) より、任意の視点において $e \leq E$ を満たすためには、微小面の発生間隔 l を

$$l \leq \min\left(\frac{E}{\tan \phi}, w \cos \phi\right) \quad (10)$$

とすればよい。

3.2 詳細度制御

描画の速度を決定するもっとも大きな要因のひとつ

表1 描画速度

Table 1 Rendering performance.

Size of volume	32^3	64^3	128^3	256^3
# facets	2460	11207	50560	208289
FPS	>30	>30	26.3	8.1

は、微小面の数である。この数を減らすことにより描画速度を向上させることが可能であるが、同時に結果画像の画質が悪化することが考えられる。そこで、描画面上で対象物体が占める領域が小さく、画質への影響が無視できる範囲で微小面の数を減らし、最終的な描画効率を上げるという、視点依存の詳細度制御できることが望ましい。

微小面の数を減少させる手段は2種類あり、ひとつはボクセル分割数を下げる方法であり、もう一つは微小面の発生間隔を上げる方法である。ただし後者に関しては、微小面の発生間隔を式 (10) で示される上限値以上に設定すると、微小面が生成されないボクセルが発生して結果画像に穴が観測される問題が生じる。

そこで、描画に必要な精度の変化に応じて、微小面の発生間隔を変更すると同時に、ボリュウムの解像度も変更して詳細度制御することを考える。ボリュウムの解像度が $i \times i \times i$ の時のボクセル幅を w_i とすると、微小面の発生間隔 l を変更する際に、

$$l \in \left(\frac{1}{2} w_i \cos \phi, w_i \cos \phi \right) \quad (11)$$

を満たすような解像度 i のボリュウムを用いて微小面を発生させることで、結果画像の現実感を損ねることなく詳細度制御できる。実際の表示の際には、あらかじめ最大解像度である L_{opt} 以下のすべての解像度のボリュウムを作成しておき、微小面間隔の変化に応じてこれらを切り替えて用いる。

描画に必要な詳細度は、式 (8) から決定できる。即ち、視点と描画面のサイズが決まったとき、 e を描画面に投影した値が許容範囲に収まる最も大きい l を用いればよい。図9に、視点移動して対象物体に近づいた時に詳細度が連続的に変化する様子を示す。この例で用いた不連続性の許容範囲は描画面上で8ピクセルである。

4. 実験結果

実験に用いる幾何形状、カメラ画像はミノルタ社製 VIVID900 を用いて計測した。この装置を使用することにより、対象物体の距離画像とカラー画像を同一視点で計測でき、両者の相対的な位置合わせを行う必要がない。計測して得られた距離画像を三角形メッシュの集合

に変換し, Neugebauer らの手法 [8] を用いて相対位置合わせを行うことで, カメラの相対姿勢を決定した. 描画処理の実験に使った計算機環境は, PentiumIII 1GHz, 主記憶 512MByte, グラフィクスカード GeForce4, ビデオメモリ 64MByte の PC である.

実験結果を図 10 に示す. 毛で覆われた物体を回転台上に設置し, 物体の周囲から 10° 間隔で, 合計 36 回の計測を行った. 計測したカラー画像と距離画像の解像度は共に 640×480 である. 入力幾何形状を第 2.2 節で述べたアルゴリズムでボリューム化して得られた 64×64×64 サイズの符号付距離ボリュームを図の左列に示す. 図中, 赤で示されている部分が符号付距離が正の領域, 緑が負の領域を意味し, 図示のために不透明度を部分的に変化させて表示している. このボリューム中で表面と交差するボクセルに対して視点依存の微小面を発生させて幾何形状を近似し, 対象物体を表示した結果が図の中央列, 最終的な描画結果が図の右列である. ここに示されている画像は, 隣接する 2 つの入力画像のカメラ視点の中間の位置で表示した結果であり, それら 2 つの画像と近似幾何形状から合成されている. また詳細度制御は行っておらず, 微小面の発生間隔 l は式 (10) で示した上限に設定している. 結果画像中において, 物体境界付近の幾何形状が不完全な部分も正確に表示できていることがわかる. また, この実験における描画速度は表 1 の通りであり, ほぼ対話的な速度で描画処理を行うことが出来る.

図 11 は入力のカメラ位置の中間に視点がある場合の見え方を従来法と比較した結果である. データの計測は対象物体の周囲から 10° 間隔で合計 36 回行い, それらを半分に間引いた 18 組の距離画像とカラー画像を用いて表示した結果と, そこに含まれていないカラー画像を比較した. 左図が, 結果描画の合成には用いられていないカラー画像である. 同一の視点において, 従来法の一つである固定のポリゴンモデルに対して視点依存テクスチャマッピングを行って表示した結果を中央図, 提案手法で表示した結果を右図に示す. 従来法では, 微小面の近似誤差が描画結果に大きく影響しているが, 視点依存の微小面を使うことにより現実感ある描画が可能になることが分かる.

5. まとめと考察

本論文では, 層状に生成される視点依存の微小面と視点依存テクスチャマッピングを利用して, 複雑な表面を持つ実物体を表示する手法を提案した. この手法

は正確な幾何形状を取得, または表現することが困難な物体に対して適用可能であり, 毛皮, 樹木など従来手法では扱うことが難しかった対象物を精度良く描画することができる.

提案手法では, 対象物の幾何形状を不連続な微小面の集合で近似することから, 特に入力画像が少ない場合には結果画像に不連続性が現れる可能性がある. これを防ぐ手法としては, 式 (2) 中のテクスチャの切り抜き幅 $\frac{w}{2}$ を大きめに設定する, 隣接するカメラ画像を用いて亀裂を埋める, などの方法が有効であるが, このことにより逆に結果画像中の不連続性が大きくなり, double imaging が増加する可能性がある. また, 遮蔽などの理由により表示する領域のテクスチャが存在しない場合は, 結果画像中に穴が現れる.

また, 選択されているカメラ毎に微小面の発生間隔を変更するため, 隣接ボクセル間で互いに異なるカメラが選択されている場合, 微小面が重なり合う領域で描画結果にテクスチャの不連続性が発生する. このような領域は画像中で少ない部分に限られるため, 最終的な画質への影響は多くの場合あまり問題にならないが, 同時に選択されるカメラ画像の数が増えると無視できなくなる可能性がある.

視点依存の詳細度制御に関しては, ボリュームを階層化することで実現したが, これをオクツリーに拡張することで, 入力形状の部分毎に必要な詳細度を評価し, 微小面の発生密度を適応的に変化させることが可能である. また, 描画結果に寄与しない微小面を描画しないように工夫することにより, 描画効率をさらに上げることが期待できるが, これらは将来の課題とする.

文 献

- [1] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. *Proc. SIGGRAPH*, pages 425–432, 2001.
- [2] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings, Annual Conference Series*, pages 307–318. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [3] P. Debevec, Y. Yu, and G. Boshokov. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*, pages 11–20, 1996.
- [4] A. Laurentini. The visual hull concept for silhouette-based image understanding. *Trans. PAMI*, 16(2):150–162, 1994.
- [5] W. Matusik, H. Pfister, P.A. Beardsley, A. Ngan, R. Ziegler, and L. McMillan. Image-based 3d photography using opacity hulls. In *Proc. SIGGRAPH 02*, pages 427–437, 2002.
- [6] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J.

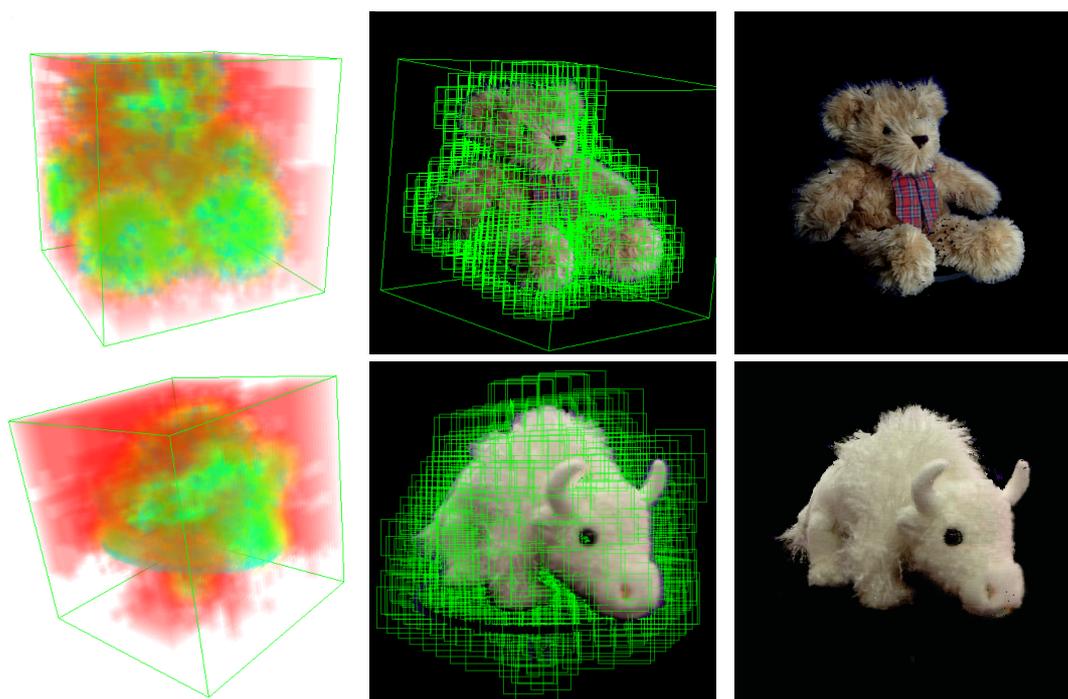


図 10 描画結果: (左列) ボリューム化された幾何形状, (中央) 視点依存微小面, (右列) 最終結果.

Fig. 10 Experimental results: (Left) Voxelized geometry, (Center) View-dependent microfacets, (Right) Final result.



図 11 入力画像のカメラ位置の中間から見たときの描画結果. 左: 実際にその視点で撮影した画像. 中央: 固定微小面に視点依存テクチャマッピングした結果. 右: 本手法の結果.

Fig. 11 The image rendered at a location between original camera positions. (Left) An image which is not used for rendering. (center) The image synthesized by texture-mapped surface rendering at the position of the image on left. (right) The image synthesized by microfacet billboarding, which successfully reproduced intricately shaped needles of the plant.

Gortler, and Leonard McMillan. Image-based visual hulls. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings, Annual Conference Series*, pages 369–374. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[7] Leonard McMillan and Gary Bishop. Plenoptic modeling: An

image-based rendering system. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings, Annual Conference Series*, pages 39–46. ACM SIGGRAPH, Addison Wesley, August 1995.

[8] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proc. Int. Conf. on*

- Shape Modeling and Application*, pages 130-139, Mar 1997.
- [9] Yizhou Yu Paul Debevec and George Borshukov. Efficient view-dependent image-based rendering with projective texture mapping. In *Proc. 9th Eurographics workshop on Rendering*, pages 105-116, 1998.
- [10] Gabriel Taubin. A signal processing approach to fair surface design. In *Proc. SIGGRAPH*, pages 351-358, 1995.
- [11] M.D. Wheeler, Y. Sato, and K. Ikeuchi. Consensus surfaces for modeling 3d objects from multiple range images. In *Proc. International Conference on Computer Vision*, January 1998.

(平成 x 年 xx 月 xx 日受付)



坂内 正夫 (正員)

1975 年 東京大学大学院工学系研究科博士課程修了。同年同大学工学部電気工学科専任講師。その後、横国大学工学部情報工学科助教授、東大生産技術研究所助教授を経て、現在、同大学生産技術研究所教授。1998 年より同大学生産技術研究所所長。マルチメディアデータベースなどの研究に従事。工学博士。



山崎俊太郎 (学生員)

1999 年 東京大学理学部情報科学科卒業、2001 年 東京大学理学部情報科学科修士課程修了、同年 東京大学大学院理学系研究科情報科学専攻博士課程入学。現在に至る。コンピュータグラフィックスに関する研究に従事。電子情報通信学会、IEEE、ACM 各会員



佐川 立昌

1998 年 京都大学工学部情報工学科卒業。2000 年 東京大学大学院工学系研究科情報工学専攻修士課程修了。現在、東京大学大学院工学系研究科電子情報工学専攻博士課程在籍。実時間視覚処理と物体のモデリングの研究に従事。



川崎 洋 (学生員)

1994 年 京都大学工学部電気電子学科卒業。2000 年 東大大学院修士課程修了。現在、同大学院博士課程在学中。主として画像処理、3次元モデリング、CG に関する研究に従事。



池内 克史 (正員)

1973 年 京都大学機械工学科卒業。1978 年 東京大学大学院情報工学博士課程修了。工学博士。MIT 人工知能研究所、電総研、CMU 計算機科学部を経て、1996 年より東京大学大学院情報学環教授。人間の視覚機能、明るさ解析、物体認識、人間による組立作業の自動認識、仮想現実感モデルの自動生成などの研究に従事。論文賞 (ICCV-90, CVPR-91, AIJ-92, 日本ロボット学会誌-97, IEEE R&A 誌-98, 日本 VR 学会誌-00) 受賞。情報処理学会、人工知能学会、OSA、IEEE(Fellow) 各会員。