社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－CG－118 (5)
2005／2／7

色情報を利用した３次元形状復元処理の高速化

坂本 尚久†
小山田 耕二‡

†京都大学工学研究科
‡京都大学高等教育研究開発推進センター

あらまし 複数枚のカメラ画像から物体の３次元形状を復元する場合，物体の外形を表すシルエット画像をもとにして処理を行う視体積交差法が多く利用されている．一般に，この手法により求められる形状データは，テクスチャ情報を持たないため，テクスチャに関する処理が別途必要となる．本報告では，シルエット画像だけでなく，撮影画像から得られる色情報も考慮することにより，物体形状の復元処理とテクスチャ処理を区別なく行うことで，一連の３次元形状復元処理を高速化する手法を提案する．

Acceleration of 3D Reconstruction by Using Color Information

Naohisa Sakamoto †
Koji Koyamada ‡

† Graduate School of Electronic Engineering. Kyoto University
‡ Center for Promotion for Excellence in High Education. Kyoto University

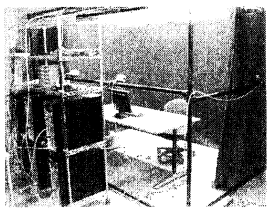
Abstract In this paper. we describe an acceleration technique for 3D reconstruction which uses color information acquired from captured images. Volume intersection method is one of the most efficient methods for reconstructing objects in real scenes from silhouette images. Generally. the reconstructed object by using this method has no texture information. Therefore. an additional texturing stage is required after the reconstruction process. To overcome this problem. we propose an acceleration technique for this 3D reconstruction process which takes into account not only the silhouettes but also color information from captured images.

1. はじめに

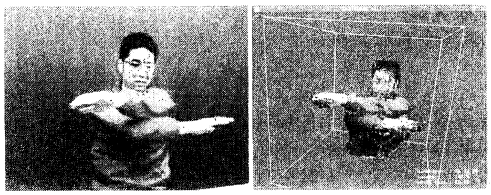
複数台のカメラ映像（多視点カメラ映像）からそこに映る物体の３次元形状を復元する技術は，文化遺産のデジタルアーカイブ化や遠隔地の物体に対するインタラクティブな可視化作業，また人物の動作解析などの分野において重要な要素技術となる．

一般に，多視点カメラ映像から物体の３次元形状を復元する問題に対しては，物体のシルエット画像から計算される視体積情報から物体形状の推定を行う視体積交差法[1][2]がよく利用される．また，実時間での処理を実現するためにさまざまな高速化手法[1]も提案されている．しかし，視体積交差法から計算されるデータは，テクスチャ情報を持たないボクセルデータであるため，後処理として，ポリゴン化とテクスチャマッピングなどの処理が必要である．

我々は，復元された物体を実時間で３次元表示することを目的として，ポリゴン化を行わず，物体を構成する各ボクセルに対して，多視点カメラ映像をもとにして直接色付けを行う実映像３次元表示システムを開発している[3]．本システム



(a) 撮影環境



(b) 撮影画像(左)と復元結果(右)

図1：従来システム

では，PC クラスタを用いて並列視体積交差法により生成されるボクセルデータに対して，ボクセル単位で色付けを並列処理することによりテクスチャ付き形状データの復元および表示を行っている．しかしながら，本システムは，まだ実時間処理には至っていない．

本報告では，これまで別々に処理されていた，形状復元処理とテクスチャ処理を区別なく行うことにより，一連の3次元形状復元処理を高速化する手法を提案する．

2. 従来システム

現在，我々が開発している実映像3次元表示システムは，5台のカメラと6台の PC からなる PC クラスタにより構成される．PC クラスタシステムは，3次元形状復元処理を行う5台の計算ノードとそれらの同期などを行う 1 台の制御ノードからなり，各計算ノードにはそれぞれ1台ずつカメラが接続されている．また，各ノード間はギガビットイーサネットで相互に接続されている．

従来システムでは，1.並列視体積交差法[1]による初期ボクセルモデル生成プロセス，2.並列ボクセルカラーリング法[4]によるテクスチャ付きボクセルモデル生成プロセスの大きく2つのプロセスにより物体の3次元形状復元を行っている（図1）．

2.1 初期ボクセルモデル生成

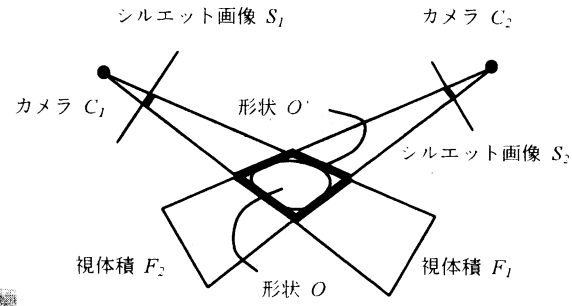


図2：視体積交差法

従来システムでは，初期ボクセルモデルの生成法として，視体積交差法（VIM: Volume Intersection Method）を利用している．

一般に，VIM では，キャリブレーション済みカメラ C_i で撮影された画像 I_i から得られるシルエット画像 S_i を3次元空間に逆投影することによって得られる視体積 F_i の積演算を行うことにより対象物体の形状 O の復元を行うことが可能である．しかし，復元される形状 O' は，シルエット画像のみから推定されるものであり，シルエット画像に反映されない凹形状部は正確に復元されない．また，カメラ台数 n の増加にともなって復元形状 O' は，真の形状 O に近づくが，一致することはない．つまり，復元形状 O' は，真の形状 O を内包する形状として推定される（式(1)，**図2**）．

$$O \subset O',\ O' = \bigcap_{i=1}^{n} F_i \qquad (1)$$

通常，VIM では形状 O' を推定するために，O' を包含する3次元ボクセル空間を定義し復元処理が行われる．このとき，この3次元ボクセル空間を部分ボクセル空間に分割することにより，処理を並列化することが可能である．中でも，Wada ら[1]は，3次元ボクセル空間を平面の集合で表し，シルエット画像の逆投影において，平面間投影を行うことによって計算量を減らし，それらを並列処理することによって実時間での形状復元を実現している．従来システムにおいては，この平面間投影を用いた並列視体積交差法を利用している．

2.2 テクスチャ付きボクセルモデル生成

前節で述べたようにして，初期ボクセルモデルが求められた後，それを構成する各ボクセルに対して色付けを行う．このとき，処理の対象となるのは，物体の表面に位置するボクセル（表面ボクセル）のみであるので，あらかじめ，それらのボ

クセルを表面ボクセルリスト（*SVL*: Surface Voxel List）に登録しておく．以後，*SVL*に対して処理を行う．

表面ボクセルへの色付けは，まず，対象としているボクセルがどのカメラから見ることが出来るのかという可視判定を行う必要がある．そして，可視と判定されるカメラに対応する画像から取得可能なピクセル値を用いて処理が実行される．

可視判定

可視判定は，カメラ毎での計算が可能なZ-Bufferを利用して次のように行う．

*SVL*に含まれるすべてのボクセルvを撮影画像面に逆投影する．このとき，カメラC_iに対して撮影画像面と同じに位置に同じサイズのバッファ領域$VMap(C_i)$（Visibility Map）を準備し，ボクセルvとカメラC_iの距離$distance(v,C_i)$が小さいものを優先的にこの$VMap(C_i)$に保存する．この際，$VMap(C_i)$上のvの投影像領域に対して，$distance(v,C_i)$とボクセルvのID番号$id(v)$を保存する．このようにして，すべてのボクセルに対して処理を行いうことで，カメラC_iに対する$VMap(C_i)$が完成し，最終的にこの$VMap(C_i)$上に残るボクセルIDを持つボクセルをカメラC_iからの可視ボクセルとして決定することができる．

色付け

各ボクセルへの色付けは，ボクセル単位で処理を行うことが可能である．このとき，対象となるボクセルの可視状態を調べる必要があり，全てのカメラに対する*VMap*が必要なる．この*VMap*を用いた判定により，対象ボクセル$v \in SVL$を可視とするカメラ画像I_iに投影することによって，対応するピクセル値$color(v,I_i)$を候補色リスト（*CCL*: Candidate Color List）に追加する．従来システムにおいては，この候補色リストに対して，平均化処理を施すことによってボクセルの色とした．

以上のようにして，*SVL*上の全てのボクセルに対して色付け処理を行うことによって，色付きボクセルモデルが生成される．

従来システムでは，連続フレームの撮影映像を効率良く処理するため，処理を複数のステージに分割し，それぞれのステージをパイプライン処理することで，システム全体のスループット向上をはかっている．

3. 3次元形状復元処理の高速化

3.1 高速化の概要

VIMでは，カメラ毎に設定される視体積を計算するために，そのカメラに対応するシルエット画像を対象空間に逆投影する必要がある．この逆投影計算により，画像上のピクセルと空間に設定されるボクセルとの対応関係が成立するが，VIMの目的は，形状を再構成することであるため，画像上のピクセル値はそのボクセルに対して何の対応付けもなされない．しかし，復元された初期ボクセルモデルに対して，色付け処理を行う際には，対象となるボクセルに対して，撮影画像をもととして，ピクセル値の割り当て処理（撮影画像を参照カラーテーブルとして扱うならばテクスチャ座標値の割り当て処理に代用可能）が行われる．つまり，対象ボクセルの画像への投影計算が必要となる．撮影から表示までの一連の3次元形状復元処理の高速化を目的とすれば，VIMの逆投影計算時に空間中のボクセルにそれぞれのカメラから得られるピクセル値を色付けの際の候補色リストとして保持しておき，テクスチャ処理の際の投影によるピクセル値の参照にかかる計算コストの削減をはかることは有効的であると考える．本研究では，VIMに色情報を考慮することで，テクスチャ処理において重複する処理を減らし，3次元形状復元処理全体の高速化を計ることを目的とする．

また，本システムは，実時間処理を実現するために，PCクラスタシステムを構築し，並列処理を行っている．このようなシステムにおいて，処理途中に発生するノード間のデータ転送は，システム全体のスループットに大きく依存する部分であり，十分な配慮が必要である．

従来システムの3次元形状復元処理においても，複数のステージでノード間データ転送が発生する．本論文では，中でも，可視判定処理において必要となるバッファ領域のノード間データ転送に着目し，そのデータ量を減らすために，新たな可視判定手法について提案する．

3.2 候補色リストの生成

本システムは，初期ボクセルモデルの生成に平面単位で計算が可能な並列視体積交差法を利用している．この手法では，カメラC_iに対する物体のシルエット画像S_iを空間中の基準となる平面に逆投影した基準シルエット画像BS_iを以下のように扱うことにより，各カメラにおける視体積F_iの交差計算を高速に行っている．

1. 視体積F_iを基準シルエット画像BS_iに平行な平面群で分割したとき，その平面におけるF_iの断面（スライス画像）は，BS_iのスケーリングおよび平行移動計算を行うことにより計算可能である．

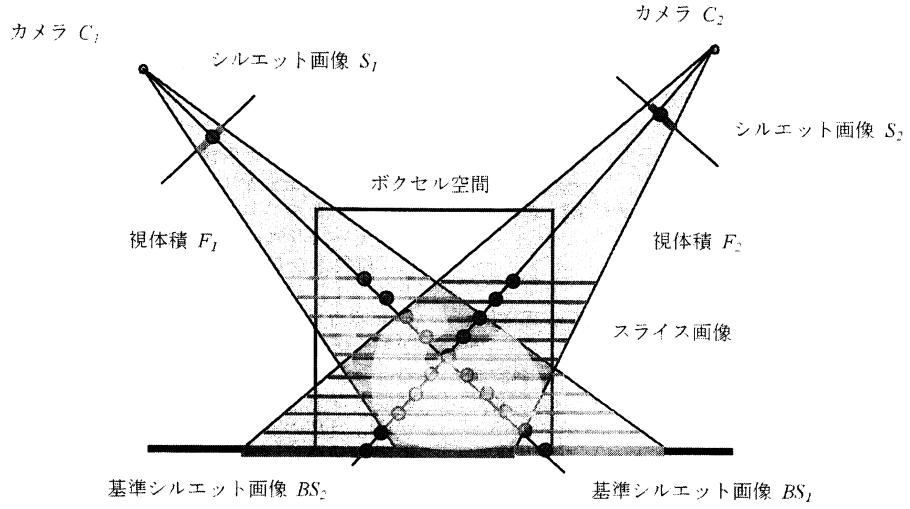


図3：平面を基とした視体積交差計算と色付け処理

2. 全てのカメラの視体積をスライス画像単位で積演算を行うことで物体の形状が復元される（図3）.

提案手法では，各ボクセルへの色付け処理を効率的に行うため，上記処理における逆投影計算時に，撮影画像のピクセル値も同時に投影する．ただし，最終的な色付け処理において，処理の対象となるのは，表面ボクセルのみであるので，スライス単位での積演算を行う際に，あらかじめ，そのようなボクセルは SVL に登録しておく．このとき，ボクセル $v \in SVL$ を全てのカメラ画像 I_i に投影することによって，対応するピクセル値 $color(v,I_i)$ を候補色リスト CCL に登録する．このとき，CCL はボクセル $v \in SVL$ に対して定義されるものであり，式(2)のように記述することができる．

$$CCL(v) = \{color(v,I_1), color(v,I_2), \cdots, color(v,I_n)\} \tag{2}$$

ただし，n はカメラ台数を表すとする．

3.3 可視判定法

ボクセル $v \in SVL$ に対して色を決定するには，撮影カメラ全てに対する可視判定の結果が必要となる．そのため，従来システムでは，判定に必要となる $VMap$ のノード間転送が発生する．この $VMap$ は，撮影画像と同じサイズとして定義していたため，ボクセルが投影されない領域（判定に必要がない）を多く含み，不要な通信コストを費やす原因となる．このような問題を解決するために，ボクセル $v \in SVL$ に対して通過ボクセルリストと可視判定マスクを定義する．

可視判定マスクは，ボクセル $v \in SVL$ に対して，カメラ台数分用意され，式(3)のように定義する．

$$visibility(v,C_i) = \begin{cases} 1 & v \text{ is visible from } C_i \\ 0 & otherwise \end{cases} \tag{3}$$

この可視判定マスクは，最初はすべて1となっている．

一方，通過ボクセルリスト（RVL: Ray oriented Voxel List）は次のようにして計算される．

Step 1. ボクセル $v \in SVL$ に対し，カメラ台数分の通過ボクセルリストを用意する．

Step 2. ボクセル v の法線ベクトル $n(v)$ とカメラ光軸ベクトル Z が $n(v) \cdot Z \leq 0$ のとき，Step3ーstep4 の処理を行う．それ以外の場合，$visibility(v,C_i) = 0$ とし，Step1 にもどる．ただし，$n(v)$ は，SVL 生成時において v の近傍26ボクセルの存在パターンにより容易に計算可能である．

Step 3. ボクセル v とカメラを結ぶレイが通過するボクセルを通過ボクセルとしてその ID を通過ボクセルリストに登録する．

Step 4. このとき，RVL に他の表面ボクセルが含まれる場合，$visibility(v,C_i) = 0$ とし，通過ボクセルリストをクリアする．

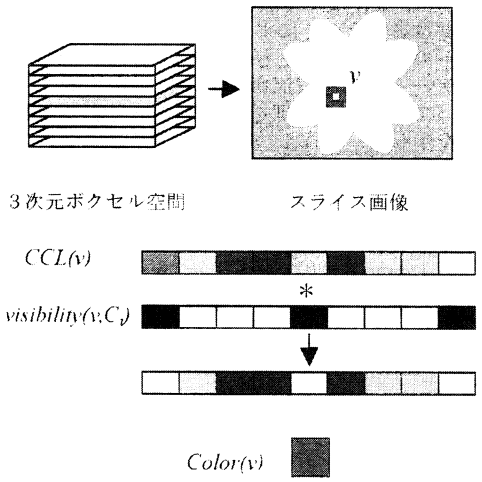


図４：色付け処理

RVL は，平面で構成される部分ボクセル空間毎に計算が可能であり，部分ボクセル空間を統合する際に，他の領域に属する表面ボクセルがリストを遮るかどうかを判定することにより，対象とするボクセルの可視判定マスクを更新することができる．

3.4 色付け

対象とするボクセル v の全てのカメラに対する可視判定マスク $visibility(v,C_i)$ が決定されれば，その候補色リスト CCL から，ボクセル v の色 $Color(v)$ は，式(4)のように決定される（図４）．

$$Color(v) = \begin{cases} \dfrac{\sum_{i=1}^{n} visibility(v,C_i) * CCL(v,C_i)}{N} & N \neq 0 \\ non-color & otherwise \end{cases} \quad (4)$$

ただし，n はカメラの台数であり，N は可視判定されるカメラの台数を表し，$N=0$ となるときは，どのカメラからも見えないボクセルであるため，色付けを行わない．また，$CCL(v,C_i)$ は，候補色リスト CCL から，ボクセル v のカメラ C_i に対応するピクセル値を取得する関数である．

4. データ量に関する考察

提案手法では，表面ボクセルリストに対して情報を持ち，処理が行われることになる．本章では，それぞれの処理ステージで必要となるデータの量を検討する．

候補色リストについて

候補色リストは，カメラ台数分の色のリストを持つ．色データを，RGB 各色 8 [bit]で表すなら表面ボクセル 1 つにつき $24 \times n$ [bit]の容量が必要となる．

一方，撮影画像をカラーテーブルとして扱うなら，色データをカメラ番号と表面ボクセルを画像面に投影したときの座標値 (u,v) を用いて表すこともできる．このとき，カメラ番号はカメラ台数分のビット配列を用意して特定できるとし，(u,v) をそれぞれ 4 [byte] = 32 [bit]で表すなら，表面ボクセル 1 つにつき $n+64$ [bit]の容量が必要となる．

以上より，カメラ台数が 3 台以上であれば，カメラ番号と (u,v) を用いてデータを持つ方が，容量が少なくて済むことが確認できる．

通過ボクセルリストと可視判定マスクについて

通過ボクセルリストと可視判定マスクもともに，表面ボクセル毎に定義される．可視判定マスクは，カメラ配列分のビット配列を用意すれば十分であるため，n [bit]の容量が必要である．一方，通過ボクセルリストは，カメラ台数分必要となるが，不可視とされるカメラに対しては，データを持たないため，理論値を出すことができない．しかし，ボクセルの法線とカメラの Z 軸ベクトルの位置関係から，半数近くの表面ボクセルは不可視と判定される可能性があるため，比較的容量は少なくて済むと予想される．また，通過ボクセルリストは，ボクセル ID のリストであるため，このボクセル ID をどのように保存するかが重要である．

多くの場合，撮影で用いるカメラの解像度は，320×240 もしくは 640×480 である．このような解像度においては，対象とするボクセル空間の解像度を 256×256×256 以上にしても，1 ボクセルの大きさが，画像の 1 ピクセルよりも小さくなってしまう可能性が大きくなるため，空間解像度は，256×256×256 以下に想定することが可能である．このような場合，ボクセル空間のそれぞれの軸方向のインデックスを考えることにより，ボクセル ID は，1 + 1 + 1 = 3 [byte]で表すことができる．

表面ボクセル数について

従来システムにおいて行った実験では，人物 1 人を対象とし，5 台のカメラを用いて 100 フレーム分の撮影を行い（撮影画像の解像度は 320

×240，ボクセル空間の空間解像度は 96×96×96)，復元したところ，生成される表面ボクセルの数は，平均約 15,000 個であった[3].

従来システムで用いた可視判定法では，ボクセル ID を 3 [byte]，距離データを 4 [byte]で表し，ボクセル空間解像度を D，カメラ台数を N とするとき，$320\times240\times(3+4)\times N\times(D/96)^2$ [byte]のデータ領域を必要としている．今，D=96，N=5 とするとき，約 2.5 [Mbyte]のデータ領域が必要である．

それに対して，提案手法では，候補色リストを 24 x 5 [bit] = 15 [byte]，可視判定マスクを 5 [bit] = 0.625 [byte]で表し，$15.000\times(15+0.625)\times(D/96)^2$ [byte] + α のデータ領域が必要である．しかし，実際には，ボクセル空間をカメラ台数で等分割した領域だけを処理するため，$\{15.000\times(15+0.625)\times(D/96)^2$ [byte] + α$\}$ / N で済む．ここで，α は，通過ボクセルリストに要するデータ領域であり，理論値を算出することが困難である．しかしながら，例えば，1 つの表面ボクセルにつき 20 個のボクセルからなる通過ボクセルリストを持っていると想定するならば，表面ボクセル 1 つあたりのデータ容量は $20\times3\times N\times(D/96)$ [byte]となる．また，ボクセルの法線とカメラ光軸の方向の関係から，表面ボクセルのうち，およそ半数のボクセルの通過ボクセルリストがクリアされると想定すると，$\alpha=20\times3\times N\times(D/96)\times(15.000/2)\times(D/96)^2$ [byte]と表すことができる．このとき，D=96，N=5 とすれば，全体として，約 0.5 [Mbyte]となる．

以上の結果からだけでは，従来システムと提案手法とを単純に比較することはできないが，カメラ台数の増加を考慮すれば提案手法の有効性が高いことは容易に確認できる．

5. おわりに

本論文では，色情報を用いた 3 次元形状復元システムの高速化について，2 つの手法を提案した．1 つは，視体積交差法の計算時に，色情報も同時に逆投影することによって，後の色付け処理のコストの軽減をはかるものであり，処理の対象およびデータ構造を表面ボクセルに限定することによって効率良く処理を行うことが可能であることを示した．もう 1 つは，PC クラスタシステムにとって全体の処理のスループットに大きな影響をおよぼすノード間のデータ転送量を考慮した，可視判定法である．この手法では，表面ボクセルに対して，通過ボクセルリストと可視判定マスクを定義することによって，無駄なく効率的に可視判定ができるデータ構造を定義した．このデータ構造により，従来システムに対してカメラ台数の増加に有効的に処理が行えることが確認できた．

今後は，実際にシステムを実装し，実データをもとにして，本システムの有効性を評価していく予定である．

参考文献

[1] T. Wada et al.: Homography Based Parallel Volume Intersection: Toward Real-Time Volume Reconstruction Using Active Cameras. In Proc. of Computer Architectures for Machine Perception. pp.331-339. 2000.

[2] 飯山 将晃，亀田 能成，美濃 導彦，"4π 計測システム：複数カメラを用いた動物体の 3 次元形状計測"，情報処理学会 第 65 回全国大会，Vol.4T7A-1 No.5 P.411-414, 2003-3.

[3] 安原 幸生，坂本 尚久，久木元 伸如，江原 康生，小山田 耕二，"全方位型表示装置を用いた実映像 3 次元表示システム" 第 9 回日本バーチャルリアリティ学会全国大会論文集. CD-ROM. 2004.

[4] N. Sakamoto et al.: A Parallel Approach for Volumetric Reconstruction. In Proc. of 3rd IASTED Int. Conf. Visualization. Imaging. and Image Processing. 2004.