

多視点映像からの実時間3次元形状復元とその高精度化

ウ小軍[†], 延原章平[†], 和田俊和[†], 松山隆司[†]

多視点映像から物体の3次元形状を復元する手法として、並列視体積交差法による高速な概形の復元を行い、この結果得られる物体の表面モデルを変形して詳細な形状を得る“弾性メッシュモデル”について述べる。並列視体積交差法では、並列化アルゴリズムとパイプライン化による実装技術について述べる。一方、弾性メッシュモデルでは、1) 物体表面上の各点で“photo-consistency”が成立する、2) モデルを観測画像上に投影した際の occluding contour と実際の対象の輪郭とが一致している、3) 変形による物体表面の交差が生じない、という3つの条件が成立するように変形を行う方法を提案すると同時に、実験によって詳細な物体形状が得られることを示す。

Real-time 3D Shape Reconstruction and Refinement from Multi-viewpoint Image Sequences

XIAOJUN WU[†], SHOHEI NOBUHARA[†],
TOSHIKAZU WADA[†], TAKASHI MATSUYAMA^{††}

This paper presents two methods for 3D shape reconstruction from multi-viewpoint images. One is volume intersection and the other is shape refinement by deformable mesh model. The volume intersection method is used for initial shape estimation, because of its stability and suitability for whole shape reconstruction. In this method, a parallel volume intersection algorithm and implementation method for PC cluster system are shown. In the next method, the result of volume intersection is converted to the surface mesh model, which dynamically changes its shape according to 1) “photo-consistency”, 2) silhouette constraint, 3) and smoothness constraint. Some experimental results demonstrating that this method improves shape accuracy are shown.

1. はじめに

本報告では、複数カメラを用いて、広い空間を動く対象を追跡している状況下で、対象の詳細な3次元形状を復元する手法について述べる。多視点2次元映像から立体形状を復元するアプローチとして、視体積交差法¹⁾²⁾⁵⁾⁶⁾とステレオ法がよく用いられている。視体積交差法はシルエット制約⁴⁾に基づく手法である。シルエット制約とは、2次元画像上のシルエットを3次元空間に逆投影し得られる錐体、視体積の内側に対象が含まれるということである。このことから、同時に得られた複数の2次元シルエットから各々視体積を求め、その共通部分を求めることで実際の対象の形状を復元する(図1)。一方、ステレオ法は、異なる視点から撮影される2次元画像上で、物体表面上の点の対応づけを行い、三角測量の要領で対象までの距離を

求め、奥行きを計算する手法である。

これらの手法を比較すると、視体積交差法の場合、ステレオ法のように対応づけ計算を行わないため、対応付けの誤りが生じず、安定な計算が行える。さらに、出力は距離画像のような2.5次元データではなく、3次元データであるため、張り合わせを行わなくても、全周囲形状が復元できるというメリットがある。しかしながら、原理的に視体積交差法では、原理的に形状のVisual Hull(図1)を求めているため、対象の詳細な形状は復元することができない。このことは、視体積交差法では、対象のシルエット情報という画像のgeometricな情報しか利用していないことに起因している。

逆にステレオ法の場合、画像のphotometricな情報の相関に基づいて対応付け計算を行い、対応付けの後には幾何学計算によって奥行きを算出している。この場合正しい対応づけができれば、物体表面上の窪みのある部分でもその三次元位置を計算することができるというメリットがある。しかし、カメラ視点間の距離

[†] 京都大学 大学院情報学専攻 知能情報学専攻
Department of Intelligent Science and Technology,
Graduate School of Informatics, Kyoto University

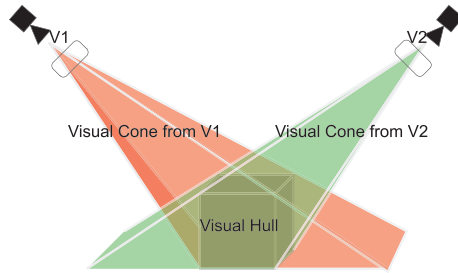


図 1 視体積交差法概念図
Fig. 1 Silhouette Volume Intersection

(基線長) が対象までの距離に対して長い場合、同じ物体表面上の点を観測するときでも視点による点の見え方が大きく異なるため、対応づけ計算が不安定になる⁷⁾。したがって、ステレオ法だけで広範囲を移動する対象の安定な形状復元を行うことは困難である。

本論文では、まず画像の geometric な情報に基づいて対象の概形を高速に復元し、その結果から対象の存在しうる範囲を限定し、さらに画像の photometric な情報に基づいて詳細な対象の三次元形状を得る方法を提案する。具体的には、平面間投影に基づく並列視体積交差法を提案し、実時間で形状復元を行うシステムを紹介する。後半では、得られた形状から物体表面形状を、“弾性メッシュモデル”を用いて表し、メッシュ上で“photo-consistency”³⁾が成立するように変形させることによって詳細な形状を求める手法について述べる。

2. 実時間形状復元

視体積交差法で復元可能な空間はすべてのカメラの共通視野に限られる。したがって、広範囲を動く対象を扱う場合、アクティブカメラを用いる必要がある。カメラをコントロールしながら、形状復元を行うためには、高速な形状復元方法が必要である。

計算を高速に行うには、並列処理が考えられる。一方、多視点映像から形状復元を実時間でいう場合、入力データを分散させる必要がある。この2点から、PC クラスタアーキテクチャが最も適しているといえる。

2.1 基本アルゴリズム

本論文では、シルエットを3次元空間に逆投影し、視体積を求め、これらの積集合を求める手法を用いる。これを高速化するためには(1)逆投影計算、(2)逆投影されたシルエットの交差計算、この2つの計算を高速化する必要がある。

2.2 逆投影計算の高速化

一般的に、平面から平面への透視投影計算は一般的

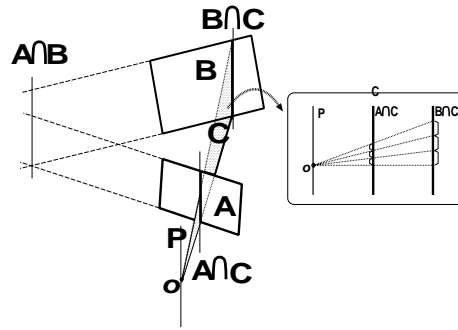


図 2 直線単位の平面間透視投影法
Fig. 2 Linear-wise Plane-to-Plane Perspective Projection

な点から点への投影に比べ計算量が少ないため、空間を平行平面群に分割し、平面間投影に基づいて、各平面上のシルエットの交差を求めることで形状復元した方が、処理速度が向上できる。したがって、平面間投影に基づく計算手法を提案する。さらに、平面間投影は次のように高速化が可能である。

2.2.1 平行でない平面間投影の高速化

投影計算に要する演算回数を削減するために、次の性質を利用することができる。

性質 1 3次元空間中に、2つの平面 A と B と点 o を考える。このとき、点 o を通る直線のうち、次の性質を満足する直線 P が少なくとも1本は存在する： P を含む任意の平面 C と平面 A, B それぞれとの交線 $A \cap C, B \cap C$ が存在する場合、これらの交線は互いに平行である。

この性質から、図2に示すように、平面 A と B が平行でない場合は、投影中心を点 o とし、点 o を通過し $A \cap B$ と平行な直線を P とすれば、 P を含む平面 C について、交線 $A \cap C, B \cap C$ は互いに平行になることが保証される。この場合、平面 A から B への平面間透視投影は A と B 上の互いに平行な直線間の投影計算に分解することができ、この平行直線間の透視投影は1次元のスケーリングに置き換えられる。したがって、平面 C を P を軸にして回転させれば、これら2平面間の透視投影が、1次元のスケーリング計算に分解できることになる。

2.2.2 平行平面間の投影の高速化

平行平面間の透視投影は2次元の拡大縮小と平行移動として簡単化できる。

これまでに述べた、直線および平面単位の平面間投影を以下のように組み合わせることにより、効率の良い逆投影計算が行える。(1)入力画像面から対象のシルエット画像をボクセル空間中の1つの平面(基準面)へ逆投影する。その際、前述の直線単位の平面間投影を適用する。(2)基準面からその他の平面への投影は面単位の平面間投影を適用する。

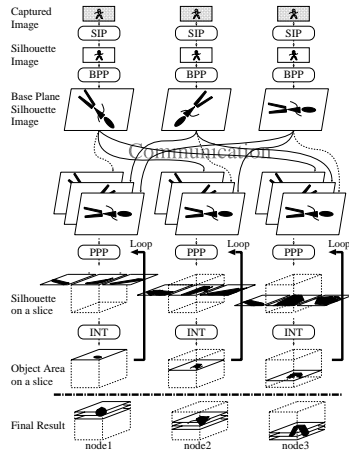


図 3 基準シルエット多重化法の処理の流れ

Fig. 3 Process flow of the base-plane duplication method

2.3 視体積交差の並列化

視体積の交差計算を高速化するため、次の並列アルゴリズムが用いられる。

- (1) 各計算機によって基準面に逆投影されたシルエット(基準シルエット)を通信によって相互コピーする。
- (2) 交差計算を行う平面集合を各計算機に割り当て、各計算機が独立計算する。

これを基準シルエット多重化法と呼び、その処理の流れを図 3 に示す。

2.4 実験

2.4.1 PC クラスタシステム

試作システムで用いた PC クラスタの仕様は次の通りである。

- クラスタは 16 台の PC から構成される。各 PC には 2 つの PentiumIII 600MHz の CPU と 512MB のメモリが搭載されている。そのうち 1 台は撮影時刻の同期をとるために使用する。残りのうち、9 台にはカメラ (SONY EVI-G20) が接続されており、さらに 6 台を計算専用とする。これら PC の OS は LINUX である。
- すべての PC は Myrinet(Myricom 社製) と呼ばれる超高速ネットワークによって相互接続されている。このネットワークは全 2 重 1.28Gbps の通信帯域幅を持ち、2 台の PC 間の通信速度は実測値で約 100[MBytes/sec] である。

カメラの配置

9 台のアクティブカメラは図 4 に示すように、部屋の天井に設置してある。

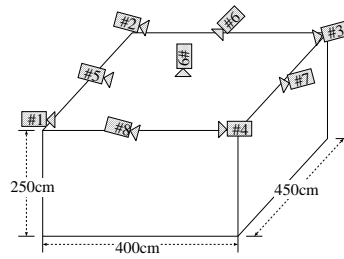


図 4 カメラの配置

Fig. 4 The camera settings

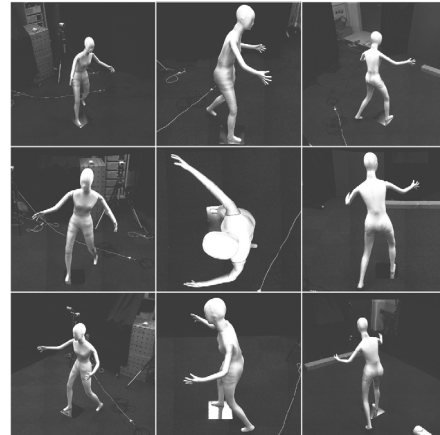


図 5 9 台のカメラで撮影した画像

Fig. 5 Examples of captured images by nine node PCs

2.4.2 システムの実装

試作システムは、基準シルエット多重化法を各 PC 上にパイプライン処理で実装した。

1 台の PC で行う処理を、画像キャプチャ(IC)、シルエット画像生成(SIG)、基準平面への逆投影(BPP)、基準シルエットの多重化(BPD)、基準平面から平行平面への投影と交差計算(PPPI)の 5 つのステージに分け、複数のスレッドによって実装した。

2.4.3 実験結果

最初の実験は、形状復元精度に関するものである。入力画像サイズは 640×480 ピクセルとし、対象は、等身大のマネキンである。この対象は、 $70 \times 70 \times 180 \text{cm}^3$ の直方体領域内に存在することを実測により確認している。まず、9 台のカメラで背景を撮影し、次に対象を置き、撮影された画像と背景画像との間の背景差分によって、対象のシルエット画像を計算した。図 5 に実際の入力画像を示す。空間解像度を 1cm^3 とした場合の復元結果は図 6 のようになる。この図において

すなわち、平行平面間の間隔を 1cm とし、各平面を 1cm^2 の単位で量子化している。

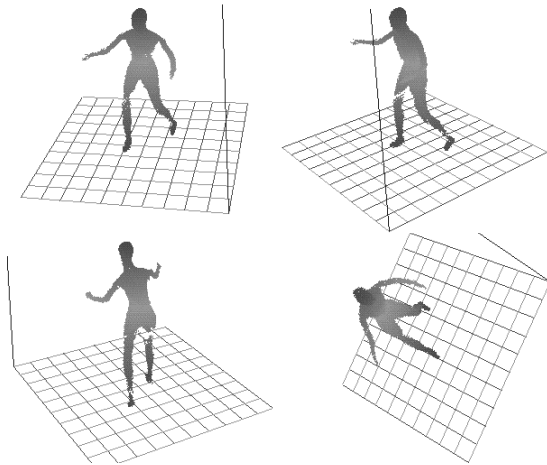


図 6 再構成された物体形状
Fig. 6 Reconstructed volume of the object

欠けている部分は、背景差分の処理で対象検出に失敗した領域である。復元結果を見ると、指などの細部まで復元されていることが分かる。

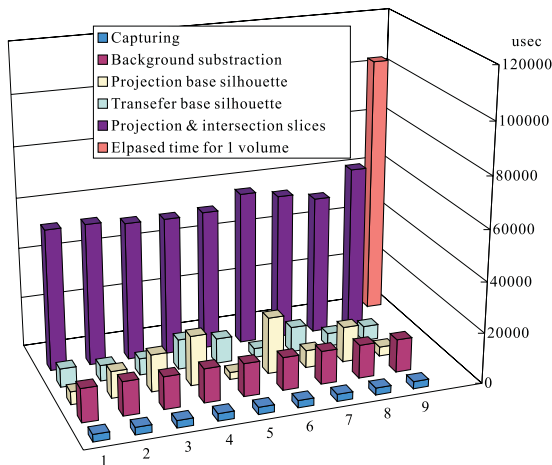


図 7 PC 毎の各処理ステージの計算時間
カメラ: 9 台, PC: 9 台

Fig. 7 Elapsed time for each stage, using 9 cameras and 9 PCs

次の実験は計算時間に関するものである。以下、画像の入力サイズはすべて 320×240 であり、空間解像度を 2cm^3 とした。

まず、9 台のカメラで撮影し、9 台の PC で処理を行った場合、計算速度は 9.78 Volume/sec であった。各 PC の各処理ステージの処理時間を図 7 に示す。横方向に各 PC について、奥行き方向に各処理ステージ

について、それぞれの処理時間を棒グラフで示した。画像入力およびシルエットの計算は入力画像サイズにのみ依存するので、各 PC 間でばらつきはないが、基準シルエットの投影計算はカメラの視線ベクトルと基準平面の法線ベクトルとがなす角度によって計算量が変わるので、各 PC 間で大きばらつき。最も時間がかかった処理は平行平面群への投影およびその上の交差を求めるステージである。前述の並列アルゴリズムは、原理上、このステージの処理は完全に独立並行処理が可能なので、計算する PC を増やせば、全体のパフォーマンスが向上できると考えられる。

これを検証するため、計算のみを行う PC を追加して、計算時間を測った。具体的に、9 台のカメラで撮影し、計算する PC を 9 台から 15 台まで増やして、各々の場合の処理時間を図 8 に示した。グラフから、計算用 PC の台数が増えるにつれ、各平面上の計算の処理時間が短縮され、全体のパフォーマンスも向上したことが確かめられた。15 台の PC で計算した場合、 12.5 volume/sec の速度で形状復元できた。

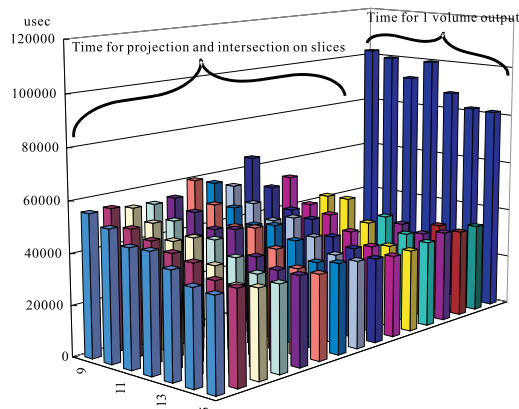


図 8 PC 毎の平行平面群への投影計算時間
カメラ: 9 台, PC: 9 ~ 15 台

Fig. 8 Elapsed time for generating silhouette intersections on slices, using 9 cameras and 9 ~ 15 PCs

最後に、撮影を行うカメラを 2 台から、9 台までのすべての場合について計算用 PC の台数をかえて、1 Volume の処理時間を図 9 に示す。図の横方向は計算する PC の台数について、奥行き方向に撮影を行うカメラの台数について、それぞれの場合の 1 Volume の形状復元にかかる時間を棒グラフで示した。

グラフに示した Capture にかかる時間は、キャプチャカードのフレームバッファから画像をコピーするのに消費されるものである。使用したキャプチャカードでは、連続した 2 つのキャプチャ要求の間に最短 $1/30 \text{ sec}$ のウェイトが発生する。

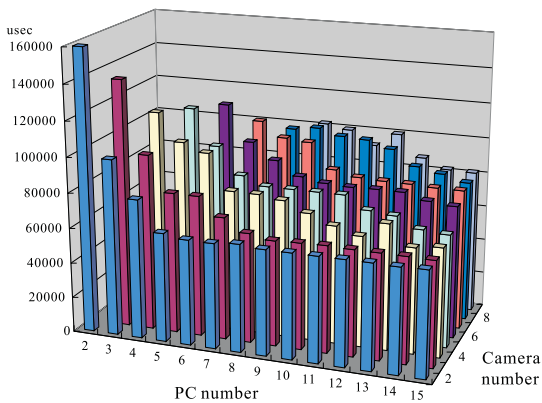


図 9 入力映像の個数と計算台数を変えた場合の 1 Volume の計算時間

Fig. 9 Elapsed time for reconstructing 1 volume, using m cameras and $m+n$ PCs

グラフから，計算する PC の台数が増えるにつれ，処理時間は短縮されるが，約 60 msec という一定値で飽和してしまう．これは以下のように考察することができる．

使用したキャプチャカードは NTSC の同期信号で画像のキャプチャ開始時刻を決めるため，連続キャプチャ要求の間に，最短でも 1/30 sec のウェイトが生じる．各 PC に接続されたカメラは NTSC 信号の同期はとれないので，PC 間で同時撮影の同期をとるため，連続キャプチャ要求の間に，最短 1/15 sec のウェイトが必要になる．このため，全体の処理時間が約 1/15 sec で飽和する．

3. 弾性メッシュモデルによる三次元形状の高精細化

以上に述べた手法により，視体積交差法の高速度が実現できる．しかし，この手法によって得られる形状は，真の対象形状ではなく，それを含む visual hull⁴⁾ でしかない．以下では，visual hull から，真の対象形状により近い形状を求める手法について述べる．

visual hull から詳細な形状復元を行う際の手がかりとしては，以下のものが挙げられる．

- 1) 視体積交差法では画像上のシルエットという対象の geometric な情報しか用いておらず，photometric な情報は用いていない．すなわち，異なるカメラで観測された画素値を復元された形状の上に投影しても，それらが物体表面上で一致するという“photo consistency”³⁾ が成立する保証はない．
- 2) 物体表面は基本的には交差のない単純閉曲面で表現できる．
- 3) 真の形状は visual hull に内接しており，各画像上

のシルエットは，visual hull でも真の形状でも同じになるはずである．

このうち，1) は物体表面を完全拡散面と見なし，カメラの特性を均一にすれば“物体表面上の点は視点によらず同じ画素値を持つ点として観測されるはずである”という知見に基づいている．この考えに基づく形状復元の高精度化は photo consistency が成立するように物体表面上の点を移動させることによって実現できる．2) の制約条件は，上述の変形操作を行う際に，表面同士が互いに交差しないように変形を行う必要があることを示唆している．さらに，単に交差が起きないだけでなく，3) の各画像上の対象のシルエットを保存するという条件が成立する範囲で変形を行わなければならない．

3.1 弾性メッシュモデル

以上の考察に基づき，次の方針で形状復元の高精度化を行うことを提案する．

視体積交差法によって得られた対象の voxel モデルから対象の“表面モデル”を求め，それを photo consistency が成立するように変形する．但し，変形によって表面同士が交差せず，しかもモデルを各カメラで撮影した場合に，各画像上の対象シルエットと一致するように変形を行う．

このような変形が行える対象の表面モデルを“弾性メッシュモデル”と呼ぶことにする．以降，この弾性メッシュモデルの詳細な設計法について述べる．

3.1.1 モデルの表現

本手法では，対象の表面形状を三角形メッシュで表現し，初期形状は視体積交差法で得られた visual hull を，離散マーチングキューブ法⁹⁾によって変換したものとする．

変形

メッシュモデルの変形は，メッシュを構成する各頂点の移動によって実現する．この際，メッシュの数が膨大になるため，SNAKES⁸⁾のようにエネルギー関数を最適化する全頂点の移動を行うのではなく，各頂点に働く力を独立に定義し，それらによって変形を繰り返すという方式を採用する．

諸定義

まず，後述の議論で用いる記号の定義を行う．

頂点を v で表したとき，その 3 次元座標を q_v で

文献³⁾では，ある点から反射された光が同じ物体表面上の点から反射した光と見なせるという意味で“photo-consistency”という性質を議論しているが，ここでは画素値が一致するという性質を指す用語として用いる．

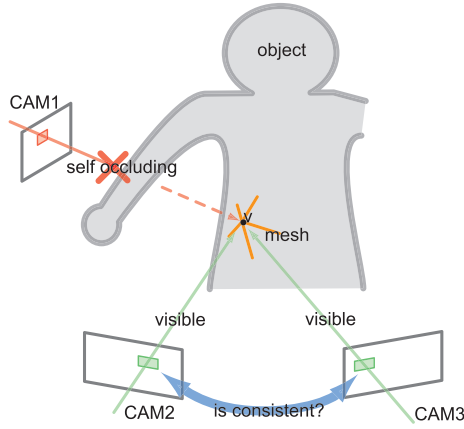


図 10 外力項と Visibility
Fig. 10 External Force and Visibility

表す．また，頂点 v を観測することができるカメラの集合を C_v で表す．例えば図 10 の場合， $C_v = \{CAM_2, CAM_3\}$ となる．

外力 F_e

まず，弾性メッシュモデルを対象形状に近づけるための力として外力項 F_e を導入する．

$$F_e \equiv \nabla E_e(v). \quad (1)$$

ここで $E_e(v)$ は，その頂点における撮影画像間の相関値 (図 10) である．つまり， F_e は，より撮影画像間の相関が高くなる方向へと作用する力である．このために， $E_e(v)$ は， C_v に含まれるカメラ c から， v に投影される画素値 $p_{v,c}$ と，その平均 \bar{p}_v を用いて，

$$E_e(v) \equiv \frac{1}{N(C_v)} \sum_{c \in C_v} \|p_{v,c} - \bar{p}_v\| \quad (2)$$

と表される．

内力 F_i

上記 F_p のみで変形を行うと，メッシュの交差が起きる可能性がある．これを防ぐために，次式で定義される内力項 F_i を導入する．

$$F_i \equiv \frac{\sum_j^n q_{v_j} - q_v}{\eta_v}, \quad (3)$$

但し v_j は v に連結する頂点とする．これは頂点間の張力として働き，メッシュ形状の滑らかさを保つ．

シルエット保持力 F_p

弾性メッシュモデルは常に visual hull に内接し，撮影視点から見たときの輪郭を保つべきである．このような働きをする力として，シルエット保持力 F_p を導入する．カメラ c で撮影画像から得られる対象本来のシルエットを $S_{o,c}$ ，得られているメッシュを投影して得られるシルエットを $S_{m,c}$ としたとき， C_v に含まれるカメラ c にとって， v が下記の場合にモデルと対象のシルエットを一致させる力 $f_p(v, c)$ を v に加える．

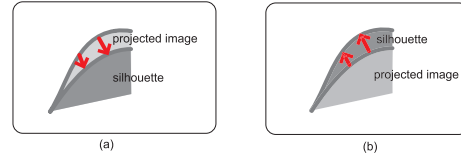


図 11 シルエット保持力
Fig. 11 Silhouette preserving force

(a) 輪郭外に投影されたとき (b) 輪郭内に occluding contour として投影されたとき

- v が $S_{o,c}$ の領域外に投影されるときは， v の投影位置を $S_{o,c}$ の輪郭に一致させる v に対する最短移動ベクトルを $f_p(v, c)$ とする (図 11(a))．
- v が $S_{o,c}$ の領域内に投影され，且つ， $S_{m,c}$ の輪郭を成すときは， v の投影位置を $S_{o,c}$ の輪郭に一致させる v に対する最短移動ベクトルを $f_p(v, c)$ とする (図 11(b))．
- 上記以外の場合， $f_p(v, c) = 0$ とする．

$f_p(v, c)$ は，各カメラごとに求められ，この合成力によって実際の頂点移動の力 F_p が次式のように定義される．

$$F_p \equiv \sum_{c \in C_v} f_p(v, c). \quad (4)$$

以上から，係数 α, β, γ を用いて，頂点に作用する合成力 F を

$$F = \alpha F_i + \beta F_e + \gamma F_p \quad (5)$$

と定義する．

3.1.2 変形計算

変形計算は，先に定義した力の平衡する位置を求めることである．本手法ではこれを反復計算で求める．これは，各頂点に働く力の間の相互作用 (ある頂点の移動による visibility の変化は，他の頂点の外力項を変化させる) を，各反復の微小移動では無視することで，計算を簡略化するためである．

3.2 アルゴリズム

以上から，本手法のアルゴリズムをまとめると以下のようなになる．

- step 1 視体積交差法によって，visual hull (volume 表現) を得る．
- step 2 離散マーチングキューブ法によって，上の volume 表現から surface 表現に変換し，弾性メッシュモデルを得る．
- step 3 メッシュの変形を行う．
 - step 3.1 各頂点に働く力を計算する．
 - step 3.2 全頂点を力に沿って移動させる．
 - step 3.3 移動が許容量以下なら終了．そうで

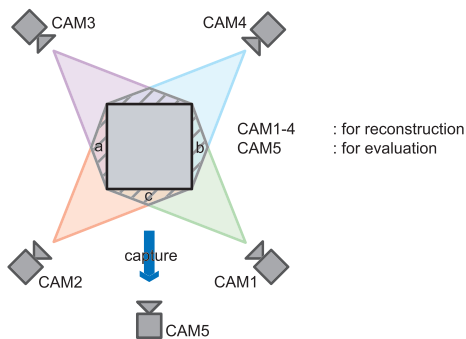
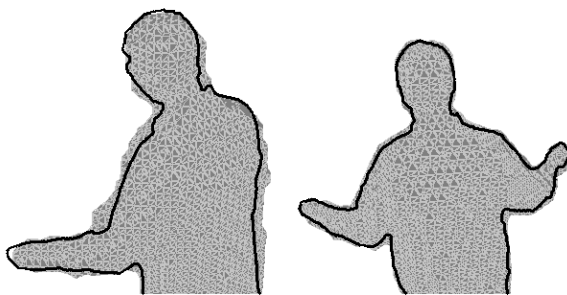


図 12 評価実験時のカメラ配置
Fig. 12 Camera Allocation



(a) CAM5 (b) CAM1

図 13 初期形状

Fig. 13 initial form

(a) は図 12 CAM5 の視点, (b) は同図 CAM1 の視点

なければ step 3.1 へ

3.3 評価

図 12 のようなカメラ配置で, CAM1 から 4 を用いて上記の手法で形状復元を行い, CAM5 を用いて元の撮影画像と復元結果の比較を行う. このとき CAM5 では, 同図 (a)(b) の領域が, $S_{m,c}$ の輪郭として捉えられ, これが $S_{o,c}$ に近づくはずである.

まず, 初期形状の場合の CAM5 および 1 での撮影画像を図 13 に示す. $S_{o,c}$ の輪郭が, 太線で示されている. 図 13(a) の CAM5 において, 図 12 の領域 (a)(b) に対応する部分 (本来の輪郭の外に投影される部分) が存在することがわかる.

実験では, F として, (a) F_i のみの場合, (b) F_i, F_s を用いた場合, (c) F_e, F_p, F_i すべて用いた場合, について計算を行った. この実験結果を図 14 に示す.

各列上段の左側が図 12 での CAM5 の視点, 右側が CAM1 の視点で見たときの復元結果である. それぞれ $S_{o,c}$ の輪郭が赤線で示されている. また下段グラフは, 反復回数に対する, 復元された形状を投影した像とのずれ量 (図 11 の ③ の長さの平均) を表している.

この結果から,

- 内力項のみの場合, 全体として縮小し, 輪郭も崩れる (図 11(a))
- 各視点 (図 12CAM1 ~ CAM4) での $S_{o,c}$ の輪郭を保つようにした場合, その視点での $S_{o,c}$ の輪郭が保たれる一方で, これ以外の視点 (図 12CAM5) では実際の対象形状との差異が大きくなる (図 11(b)).
- 更に外力項を加えた場合, 復元に用いなかった視点 (図 12CAM5) においても実際の対象形状に近づく (図 11(c))

ことがわかり, 詳細な形状復元のためにはこれら 3 つの力全てが必要であることが確認できる.

ここで述べた手法と同様のことは, Space-Carving³⁾ にシルエット制約を導入することでも実現できると考えられる. この場合, 本手法との主な違いは, 形状モデルの表現法と, 形状をモデルフィッティングによって求めるか削り込みによって求めるかという 2 点になる.

しかし, モデルフィッティングを基本原理とする本手法は, 異なる時刻に撮影された画像系列に対してもフィッティングを行えるように拡張可能である. これによって対象の動作解析や, データ圧縮などの問題を扱うことができるようになって考えられる.

4. まとめ

本論文では, 平面間投影に基づく並列視体積交差法と, PC クラスタ上での実時間形状復元システムの試作, さらに, 弾性メッシュモデルを導入し, 視体積交差法で求めた対象の概形を詳細化する手法を提案した.

並列視体積交差法と弾性メッシュモデルを組み合わせることは, まず画像の幾何学的情報のみを用いて対象の概形 (visual hull) を高速に復元しておき, その visual hull に内接して含まれる真の形状を求めるため, さらに画像の photometric な情報を利用して表面形状の変形を行うということを意味している. これは, いきなり画像間の対応点探索を行うと, 特に長基線ステレオの場合, 不安定な結果しか得られないため, まずシルエット制約に基づいて解空間を限定し, その後で得られた形状の変形によって詳細化を進めることが妥当な結果をもたらすという考えに基づいている.

視体積交差法ではカメラの視線方向をコントロールするための情報を高速に得るため, 実時間処理を目指し, 弾性メッシュモデルはこの結果をオフラインで詳細化するために用いることが考えられる. また, 弾性メッシュモデルをさらに拡張すれば, 異なる時刻に撮影

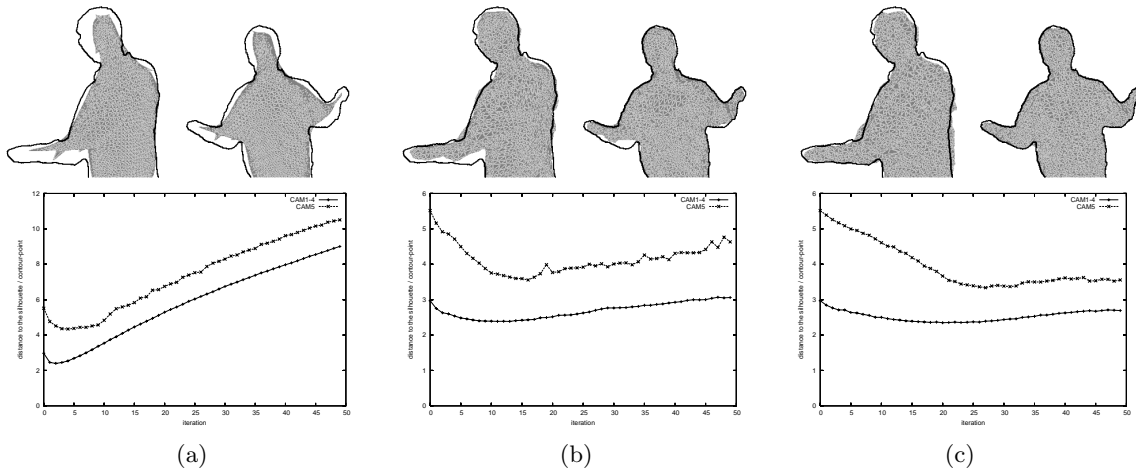


図 14 実験結果

Fig. 14 deformation results

(a) F_i のみの場合 ($\alpha = 1.0, \beta = 0.0, \gamma = 0.0$), (b) F_i, F_s を用いた場合 ($\alpha = 0.5, \beta = 0.0, \gamma = 0.5$), (c) F_e, F_p, F_i すべて用いた場合 ($\alpha = 0.3, \beta = 0.4, \gamma = 0.3$). それぞれ左が図 12 での CAM5 の視点, 右が CAM1 の視点. 左右ともに 1 行目が変形前, 2 行目が変形後, 太線は撮影画像から得られるシルエット $S_{o,c}$ の輪郭 (反復 50 回). グラフ横軸は反復回数, 縦軸は $S_{o,c}$ と $S_{m,c}$ の間の誤差.

された画像データに対して弾性メッシュモデルをフィッティングすることも可能になると考えられる. このような手法が実現できれば, 復元された対象の 3 次元形状を単に時間軸上に並べるだけでなく, 異なる時刻間での対応点の情報も得られることになり, 3 次元物体の情報圧縮や動きの解析へと研究のステージを進めることが可能になるはずである.

我々は, これらの手法をベースとして, 3 次元運動物体の“見え”に関わる全ての情報を復元・圧縮・伝送・再生する“3 次元ビデオシステム”の構築を行っている. 今後は, カメラアクションの導入と, ビデオレートでの物体形状復元システムの構築, および異なる時刻間での弾性メッシュモデルのフィッティングを行う予定である.

参 考 文 献

- 1) H. Baker. Three-dimensional modelling. In *Fifth International Joint Conference on Artificial Intelligence*, pages 649–655, 1977.
- 2) B.G. Baumgart. Geometric modeling for computer vision. Technical Report AIM-249, Artificial Intelligence Laboratory, Stanford University, October 1974.
- 3) K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. In *IEEE International Conference on Computer Vision*, pages 307–314, 1999.
- 4) A. Laurentini. How far 3d shapes can be understood from 2d silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):188–195, 1995.
- 5) W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1987.
- 6) P. Srivasan, P. Liang, and S. Hackwood. Computational geometric methods in volumetric intersections for 3d reconstruction. *Pattern Recognition*, 23(8):843–857, 1990.
- 7) Philip Pritchett and Andrew Zisserman. Wide Baseline Stereo Matching. *International Conference on Computer Vision* pp.754-760, (1998)
- 8) M. Kass and A. Witkin and D. Terzopoulos. Snakes: Active Contour Models. *Proc. of IEEE Conference on Computer Vision*, 8-11, London, England, 1987, pages259–268
- 9) 剣持 雪子, 小谷 一孔, 井宮 淳. 点の連結性を考慮したマーチング・キューブ法. 信学技報 *TECHNICAL REPORT OF IEICE*, IE98-218 (1999-01), pages 197-204