

## 未校正多視点画像からの多面体表面モデルの直接復元法

山口 義隆<sup>†</sup>      斎藤 英雄<sup>†\*</sup>

<sup>†</sup> 慶應義塾大学理工学部情報工学科

\* 科学技術振興事業団，さがけ研究 2-1 「情報と知」領域

多視点画像から対象の 3 次元多面体モデルを生成する際には，まず，ボクセル空間で 3 次元形状を復元してから物体表面を抽出する，という手法が従来から用いられていた．しかし，細かくボクセル空間を設定すると，計算量が膨大になるなどの問題点があった．本手法では多角形近似されたシルエットからボクセル空間での形状復元を行わずに，物体表面の多面体モデルを直接復元する手法を提案する．本手法は Projective Grid Space 内において Visual Hull を画像上に投影し，投影した共通部分の各三角メッシュに対し，シルエットを近似した多角形との交点を求める．さらにメッシュ両端の投影前の 3 次元座標から線形的に交点の 3 次元座標を求め，三角メッシュ化する．この一連の処理を繰り返すことで 3 次元多面体表面モデルを生成する．

## Direct Reconstruction of Polygon Mesh Surface Model from Uncalibrated Multiple Views

Yoshitaka Yamaguchi<sup>†</sup>, Hideo Saito<sup>†\*</sup>

<sup>†</sup> Department of Information and Computer Science, Keio University

\* “Information and Human Activity”, PRESTO, JST

In this presentation, we will show a method for reconstruction of polygon mesh surface model from multiple views. Our method permit reconstruction of polygon mesh surface model without voxel space curving. First, we project visual hull to the image in projective grid space. Second, at each mesh we search intersection point of the mesh and silhouette approximated polygon, and calculate linearly 3D coordinate from 3D coordinate of either end of mesh. Finally, we implement triangulation to all points. Iterating this process, we reconstruct polygon mesh surface model.

### 1 はじめに

近年，マルチメディアの発展に伴い，ビジネス・教育・娯楽などでコンピュータ上に仮想物体を復元することに関心が集まっている．例としてはバーチャル美術館やサイバーショッピングモールなどが挙げられ，様々な分野への応用が期待されている．

しかし，現段階では人間の手作業による CG によって復元されており，作成するには非常に時間と労力を要するのが現状である．

実物体から形状を計測し，その形状データを入力す

ることができれば，省力化を図れるため，実際に存在する物体を複数台のカメラによって撮影した画像から，対象物体をコンピュータ上に復元する 3 次元モデリング・レンダリングに関する研究が盛んになってきた [1][2]．

3 次元モデリングの方法としては 3 次元スキャナを用いて収集された複数の距離画像を融合して，3 次元構造を体積空間で再構築する方法 [3][4] や，多視点画像から対象の 3 次元多面体表面モデルを生成する方法 [5][6] が挙げられる．

多視点画像から対象の 3 次元多面体表面モデルを生成する際には，まず，ボクセル空間で 3 次元形状を復

元してから物体表面を抽出する，という手法が従来用いられていた．しかし，詳細な 3 次元多面体表面モデルを作るために細かくボクセル空間を設定すると，計算量が膨大になるなどの問題点があった [7][8]．

また，多視点カメラから 3 次元復元を行う場合，すべてのカメラに対しカメラキャリブレーションを行う必要があったが [9]，近年キャリブレーションを必要としない多視点カメラからの 3 次元復元の枠組みとして Projective Grid Space [10][11] が提案されている．Projective Grid Space とは画像の中心射影によって定義される仮想 3 次元空間であり，3 つのカメラをベースカメラとして，ベースカメラ 1 の画像の  $x, y$  軸，ベースカメラ 2 の画像の  $x$  軸をそれぞれ  $P, Q, R$  軸とした 3 次元空間である．

そこで，本手法ではカメラキャリブレーションを取得しないカメラで同時に撮影された画像から，それぞれのカメラ間の  $F$  行列を用いて構成される Projective Grid Space 内において，ボクセル空間での形状復元を行わずに物体表面の多面体表面モデルを直接復元する手法を提案する．

また，CG 画像に対し，ボクセル空間で形状復元を行う従来法と本手法をそれぞれ適用して 3 次元多面体表面モデルを生成し，実際に与えた入力形状とそれぞれ比較することで従来法と本手法の比較を行う．

## 2 理論

ここでは本手法で使用する理論について述べる．

### 2.1 Shape from Silhouette

画像処理を用いて 3 次元形状復元を行う手法の中に物体のシルエットを利用した Shape from Silhouette[12] と呼ばれる手法がある．対象物体をいくつかの方向から撮影し，背景との差分をとることにより各方向の対象物体のシルエットを作成する．各画像のシルエット情報と各カメラのカメラパラメータが既知であれば，3 次元空間をシルエットに基づいて区切ることにより物体の形状を得ることができる．

結果として得られる物体の形状のことを Visual Hull と呼ぶ．Visual Hull は実際の物体を中に閉じた空間を形成し，対象物体を近似したものとなる．

対象となる 3 次元物体をシルエット画像群  $R$  により

形状を復元する場合を考える．各視点画像  $r$  において，カメラ座標系の原点と物体シルエット領域内の各画素を結ぶ直線により形成される 3 次元物体形状を  $v_r$  とすると，1 視点だけで作られる物体形状は円錐形となる．ここで，このカメラ画像系の原点，すなわち光学中心とシルエット画像により形成される円錐形の物体形状  $v_r$  を silhouette cone と呼ぶことにする．このとき，複数の silhouette cone が交差する領域を求めることは，次式で表すように各視点から得られた物体形状領域の論理積，つまり共通領域を求めることと等価であり，これにより物体の 3 次元形状を取得することができる．

$$v_R = \bigcap_{r \in R} v_r \quad (1)$$

$R$ : 各視点画像  $r$  の集合

式 (1) により表現される閉空間  $v_R$  を Visual Hull と呼ぶ．シルエット画像群  $R$  の大きさが無限に向かうとき， $v_R$  は対象物体に対して最適な Visual Hull となる．

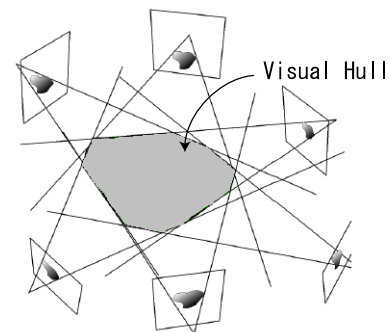


図 1: ShapeFromSilhouette と VisualHull

### 2.2 Projective Grid Space

Projective Grid Space は，基準となる 2 台のカメラの中心射影によって構成される 3 次元空間である．3 次元空間を離散的に扱う場合，2 次元画像における単位面積である画素 (Pixel) と同様に 3 次元の単位体積は Voxel と呼ばれる．またここでは Voxel を区切る直線をグリッドと呼ぶことにする．

図 2 はカメラシステムと，一般的に 3 次元空間を離散的に扱う場合の Euclidian Grid Space, Projective Grid Space との関係を示したものである．Euclidian Grid Space はカメラとは関係なく，ある 1 点を原点として定義された直交 3 次元空間であり各 Voxel の大きさも等しい．これに対し，Projective Grid Space は

複数台あるカメラのうち2台をベースカメラとし、そのカメラ2台のカメラによって中心射影的に決定される3次元空間である。この図のように Projective Grid Space は一般の Euclidian Grid Space から見ると各3軸及び各グリッドは直交するとは限らない。また各 Voxel の大きさも一定ではなく、ベースカメラに近いほど小さく、遠いほど大きくなるという性質を持つ。

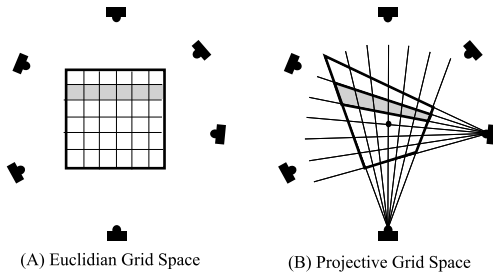


図 2: Euclidian Grid Space と Projective Grid Space

次に実際の Projective Grid Space の定義について述べる。まず複数台のカメラシステムの中の2台を選択し、ベースカメラ1、ベースカメラ2とする。ベースカメラ1から得られる画像の  $X$  軸、 $Y$  軸をカメラの視点から中心射影的に空間に投影したものを Projective Grid Space の  $P$  軸、 $Q$  軸とする。同様にベースカメラ2から得られる画像の  $X$  軸から Projective Grid Space の  $R$  軸を決定する。これら  $P, Q, R$  軸を3軸として定義される3次元空間が Projective Grid Space である。また Projective Grid Space における座標は各画像の画素とその視点とを結ぶ直線によって定義されるものとする。つまり Projective Grid Space の  $P$  座標  $Q$  座標がそれぞれベースカメラ1の画像の  $X$  座標  $Y$  座標に対応し、 $R$  座標が基底カメラ2の  $X$  座標に対応する。ベースカメラ以外の他の全てのカメラもこの座標系を用いるものとする。

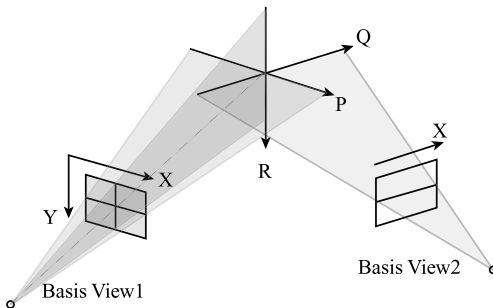


図 3: Projective Grid Space の定義

次に Projective Grid Space における視点位置の3次元座標の定義を示す。ベースカメラ1, 2の各視点を  $C_1, C_2$  としその他カメラの視点を  $C_i$  とする。Projective Grid Space の定義よりベースカメラ1から得られる画像の  $X$  軸、 $Y$  軸によってカメラ1の視点  $C_1$  から中心射影的に  $P$  軸、 $Q$  軸が定められているから、 $C_1$  の  $P$  座標、 $Q$  座標は一意には定まらずあらゆる値を取りうるが、ここでは処理の汎用性を考えてベースカメラ1の画像中央の  $X$  座標  $X_{1c}$ 、 $Y$  座標  $Y_{1c}$  をそれぞれ  $C_1$  の  $P$  座標、 $Q$  座標とする。Projective Grid Space の  $R$  軸は画像2の  $X$  軸として定義されることから、画像2に対するエピポールとなる  $C_1$  の  $R$  座標は  $e_{21}$  の  $X$  座標 ( $e_{21_x}$ ) となる。つまり、 $C_1$  の座標は  $C_1(X_{1c}, X_{2c}, e_{21_x})$  になる。同様に、 $C_2$  の Projective Grid Space における  $R$  座標も一意には定まらないが、ベースカメラ2の画像中央の  $X$  座標  $X_{2c}$  を用いて定義されるものとする。 $C_2$  は画像1に対するエピポールとなるので、その  $P$  座標、 $Q$  座標はそれぞれ  $e_{12_x}, e_{12_y}$  となる。つまり、 $C_2$  の座標は  $(e_{12_x}, e_{12_y}, X_{2c})$  と表すことができる。また、ベースカメラ1, 2以外のカメラの視点  $C_i$  の座標は、ベースカメラ1, 2のエピポールの座標を用いて、 $C_i(e_{1i_x}, e_{1i_y}, e_{2i_x})$  であることがわかる。

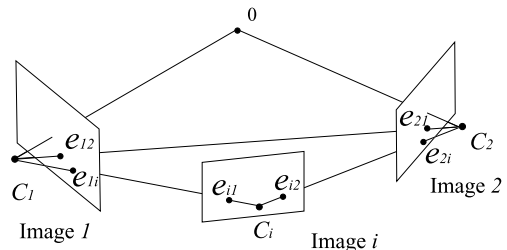


図 4: Projective Grid Space における視点座標

### 3 提案する手法

#### 3.1 処理の概要

ここでは、本手法の処理の概要について述べる。まず大まかな処理の流れを図5に示す。

本手法では、キャリブレーションされていない多視点カメラによって同時に撮影された画像を入力画像とし、それぞれ背景差分を行って得られたシルエット画像を多角形近似することにより多角形近似シルエット画像を生成する。また、 $F$  行列算出用の入力画像列が

らカメラ間の対応点を抽出し， $F$  行列を算出する．

この多角形近似シルエット画像と  $F$  行列を入力とし，Projective Grid Space 内において 3 次元多面体表面モデルを生成する．

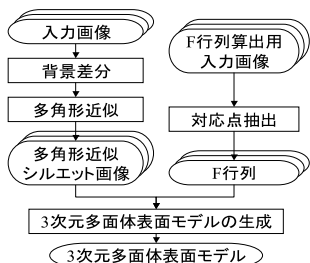


図 5: 本手法の流れ

### 3.2 多角形近似シルエット画像の生成

すべてのカメラの入力画像に対し，あらかじめ撮影した背景画像との背景差分を行い，ノイズ除去を行うことによって各画像のシルエット画像を生成する．入力画像が RGB ビットマップ形式のため，RGB それぞれ独自にしきい値を設定することでより正確な対象物体のシルエットを生成することが可能である．

そして，得られたシルエット画像を手動により多角形近似することにより，多角形近似シルエット画像を生成する．

### 3.3 $F$ 行列の算出

ベースカメラ 1 と他のすべてのカメラ，ベースカメラ 2 と他のすべてのカメラとの間で対応点を抽出する．対応点は 3 次元空間的に偏りなく取るようにする． $F$  行列は十分な精度であれば 2 画像間で 8 点の対応点があれば算出することができる．しかし画像上では 1 画素単位の精度でしか点の座標を取ることができないため，多くの対応点を取るによってその誤差を小さくする．

### 3.4 3 次元多面体表面モデルの生成

多角形近似シルエット画像，ベースカメラと他のカメラ間の  $F$  行列を用いて 3 次元多面体表面モデルを生成する．

今，ベースカメラ 1 の画像の  $x, y$  軸，ベースカメラ 2

の画像の  $x$  軸をそれぞれ  $P, Q, R$  軸として 3 次元空間を決定する．本手法は 3 次元空間内の Visual Hull を各カメラの多角形近似シルエットで削りとっていくことで 3 次元多面体表面モデルを生成していく．なお，初期 Visual Hull に関しては手動で与える．

図 6 にあるカメラ  $i$  における処理の流れを示す．現段階における Visual Hull を Projective Grid Space を用いた投影によりカメラ  $i$  の画像上に投影する．そして多角形近似シルエット画像を用いて画像上で共通部分を切り抜き，更新後の Visual Hull の 3 次元座標を求め，三角メッシュ化することで更新後の Visual Hull を求める．

この処理をすべてのカメラで行うことで 3 次元多面体表面モデルを生成していく．

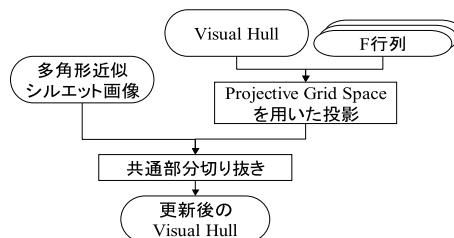


図 6: カメラ  $i$  における処理の流れ

#### 3.4.1 Projective Grid Space を用いた投影

実際に Projective Grid Space 内の 3 次元点を各カメラの画像に投影する方法について述べる．投影する 3 次元点を  $A(p, q, r)$ ，ベースカメラ 1, 2 から得られる各画像を画像 1, 画像 2，ベースカメラ以外のカメラから得られる画像を画像  $i$  とする．またそれぞれの画像上の  $A$  の投影点を  $a_1, a_2, a_i$ ，画像  $h$  の画像  $k$  に対する  $F$  行列を  $F_{hk}$  と表すものとする．

3 次元点  $A(p, q, r)$  は Projective Grid Space の定義より，画像 1 では  $a_1(p, q)$  に投影されることがわかる．また，画像 2 の画像 1 に対する  $F$  行列を用いて  $a_1$  を画像 2 に直線  $l$  として投影すると直線  $l$  は (2) 式のように定義される．

$$l = F_{21} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \quad (2)$$

Projective Grid Space の定義より 3 次元点  $A(p, q, r)$  の画像 2 における投影点  $a_2$  の  $x$  座標は  $r$  であるから，

$a_2$  はこの直線  $l$  上の  $x$  座標が  $r$  の点として定めることができる。

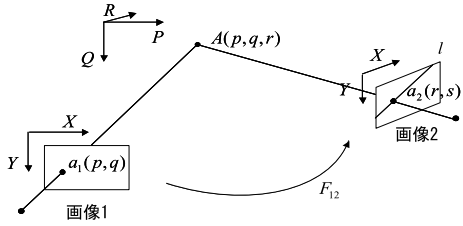


図 7: ベースカメラ画像への投影法

最後に画像  $i$  の投影点  $a_i$  の決定法を述べる。先ほどと同様に  $F_{i1}$  を用いて  $a_1$  を画像  $i$  に直線  $l_1$  として投影する。また  $a_2$  を  $F_{i2}$  で画像  $i$  上に直線  $l_2$  として投影する。この 2 直線  $l_1$  と  $l_2$  の交点が画像  $i$  の投影点  $a_i$  となる。

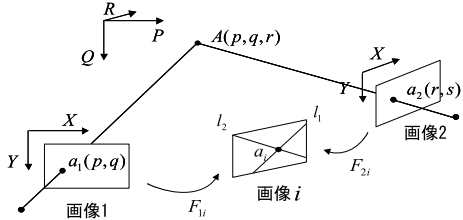


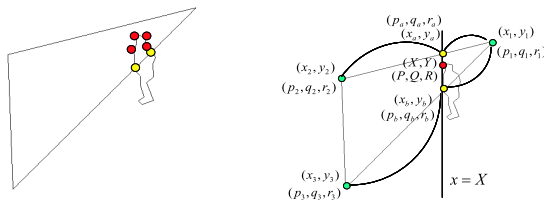
図 8: その他のカメラ画像への投影法

### 3.4.2 共通部分切り抜き

Visual Hull の各メッシュごと処理を行っていく。

今、Visual Hull のある三角メッシュを  $M_j$  とし、 $M_j$  の 3 次元座標を  $(p_1, q_1, r_1), (p_2, q_2, r_2), (p_3, q_3, r_3)$ 、投影された画像上の座標を  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$  とする。

まず、 $M_j$  と多角形近似シルエットとの交点と  $M_j$  内にある多角形近似シルエットの頂点を求める。これらが共通部分の頂点である。それぞれの頂点の 3 次元座標は  $M_j$  の投影前の 3 次元座標が既知であるからこれを用い線形的に求める。



(a) 共通部分の頂点 (b) 3 次元座標算出  
図 9:  $M_j$  における 3 次元座標算出

$M_j$  内の画像上の点  $(X, Y)$  に対する 3 次元座標  $(P, Q, R)$  は  $x = X$  と三角メッシュとの交点をそれぞれ  $(x_a, y_a), (x_b, y_b)$ 、それぞれに対応する 3 次元座標を  $(p_a, q_a, r_a), (p_b, q_b, r_b)$  とすると、

$$\begin{cases} P = \frac{p_b - p_a}{y_b - y_a}(Y - y_a) + p_a \\ Q = \frac{q_b - q_a}{y_b - y_a}(Y - y_a) + q_a \\ R = \frac{r_b - r_a}{y_b - y_a}(Y - y_a) + r_a \end{cases} \quad (3)$$

である。

求めた共通部分の頂点の 3 次元座標、すなわち  $M_j$  内の更新後の Visual Hull の頂点に対し、三角メッシュ化を行う。三角メッシュ化は画像上の座標を入力として、2 次元ドロネー網 [13] を適用することにより行う。 $M_j$  内の 3 次元頂点は同一平面に乗っているので、3 次元空間内でもメッシュ情報は同様である。

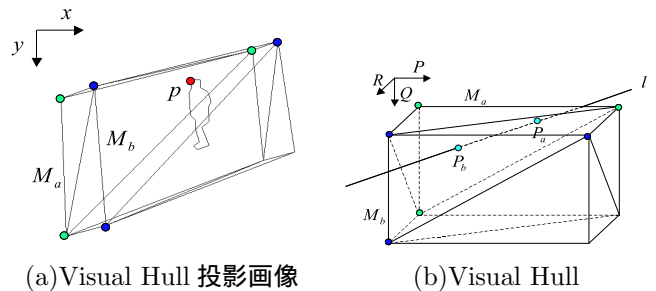


図 10: 三角メッシュ化

また、画像上では奥行き情報が失われているため、現在考えている共通部分の頂点の 3 次元空間内における奥行きを考慮する必要がある。

現在考えている  $M_j$  内の共通部分の頂点を  $p$  とすると、 $p$  は 3 次元空間内では直線  $l$  として表されている。投影した Visual Hull のすべての三角メッシュに対し、 $p$  が入っている三角メッシュを調べ ( $M_a, M_b$ )、それぞれの 3 次元座標を線形的に求める ( $P_a, P_b$ )。そして  $M_j$  内より求めた 3 次元座標  $P$  と 3 次元座標を比較することにより、 $p$  の奥行きを決定する。

例えば、 $M_j = M_a$  のときは  $P = P_a$  となるので、 $p$  は奥行き軸  $R$  に対して手前であると決定される。



(a) Visual Hull 投影画像 (b) Visual Hull  
図 11: 奥行き の 考慮



以上の  $M_j$  における処理を Visual Hull のすべての三角メッシュに対して行う。

ここにおいて、更新後の Visual Hull の 3 次元座標はすべて求められたが、三角メッシュ化は奥行き軸方向に対してされていない。そこで、奥行き軸方向の三角メッシュ化を行う。多角形近似シルエットの一边は 3 次元空間内では平面として表させるので、この平面上にある頂点を入力として、2 次元ドロネー網を適用することにより行う。

ここで 3 次元座標を平面上の 2 次元座標に変換する必要がある。今、平面上に乗っているある 2 点間の単位ベクトルを求めて  $x$  軸ベクトル  $(P_x, Q_x, R_x)$  とし、これと直交する  $y$  軸ベクトル  $(P_y, Q_y, R_y)$  を求める。 $y$  軸ベクトルは  $x$  軸と直交するため  $y$  軸ベクトルとの内積が 0 であり、絶対値が 1 であり、平面上にある。以上の条件をふまえると

$$\begin{cases} P_x P_y + Q_x Q_y + R_x R_y = 0 \\ P_y^2 + Q_y^2 + R_y^2 = 1 \\ a P_y + b Q_y + c R_y = 1 \end{cases} \quad (4)$$

の連立方程式から  $y$  軸ベクトルを求めることができる。ここで、 $a, b, c$  は平面の式を表す係数であり、平面上の 3 点の 3 次元座標より求める。

そして、3 次元座標と求めた  $x, y$  軸ベクトルの内積をとることにより、平面上での 2 次元座標が得られるので、これらを入力として 2 次元ドロネー網を適用して、三角メッシュ化を行う。

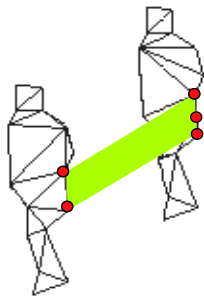


図 12: 奥行き方向の三角メッシュ化

## 4 実験と考察

本手法を多視点画像に対し適用した。実験は以下に示す 2 種類のものを行った。

実験 1 では POV-Ray を使用し生成した CG 画像を入力とすることで、ボクセル空間で形状復元を行う従来法と本手法の比較を行った。

実験 2 では実際にカメラキャリブレーションを取得しないカメラで同時に撮影された画像に対し、本手法を適用した。

### 4.1 CG 画像への本手法の適用

入力として用いた CG 画像は CG 作成ソフト POV-Ray によって作成した。入力画像は  $640 \times 480$  画素の RGB24 ビット画像である。3 次元空間内に作成した球を 5 方向から見た画像を入力画像とした。また、これとは別に  $F$  行列算出用の入力画像を同じ 5 方向から作成し、対応点を抽出することによって  $F$  行列を求めた。そして、5 枚の画像のうちの 2 枚をベースカメラ画像とし、残りの 3 枚の画像からそれぞれ 3 次元多面体表面モデルを生成した。

CG は入力形状が既知であるので、従来法と本手法により生成された 3 次元多面体表面モデルを入力形状を比較することにより、従来法と本手法の生成形状の比較を行った。また、生成時間の比較も行った。この際 PC は Intel(R) Pentium を使用し、CPU は  $1400MHz$ 、メモリは  $512MB$  であった。

従来法は Projective Grid Space 内にボクセル空間を設定し、1 つ 1 つのボクセルを画像に投影してそれぞれがシルエットの内外であるかを判定することにより形状復元を行い、その後マーチングキューブを適用することにより物体表面を抽出した。今回の実験ではボクセルサイズを  $350 \times 300 \times 250$  とした。

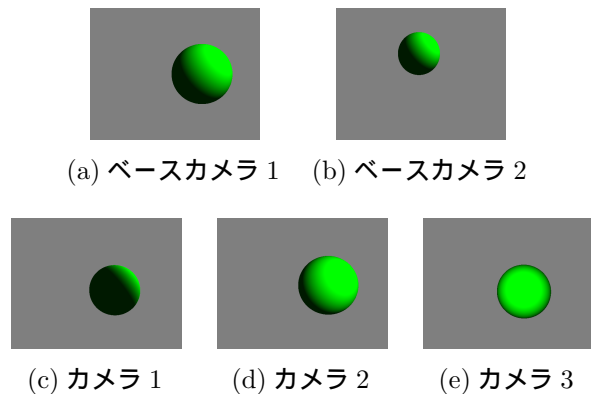


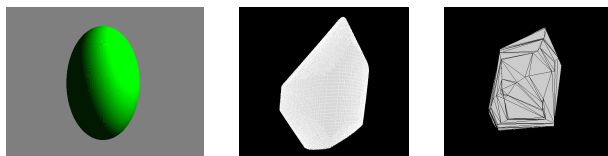
図 13: 入力画像

図 14 は従来法と本手法の比較を示したものであり、(b),(e),(h) と (c),(f),(i) はそれぞれ従来法と本手法により生成された 3 次元多面体表面モデルを同じ 3 方向から見たものである。(a),(d),(g) は入力形状であるが、図 13(a),(b) に示したベースカメラ画像に写った円の半径がそれぞれ 122,78 であることから、球は Projective Grid Space 内では

$$\frac{p^2}{122^2} + \frac{q^2}{122^2} + \frac{r^2}{78^2} = 1 \quad (5)$$

であらわされる楕円球としてあらわされる。

また、従来法によるシルエット画像からの 3 次元多面体表面モデル生成時間は 0.47sec、本手法による多角形近似シルエット近似画像からの 3 次元多面体表面モデル生成時間は 82.39sec であった。



(a) 入力形状 (b) 従来法 (c) 本手法

図 14: 従来法と本手法の比較

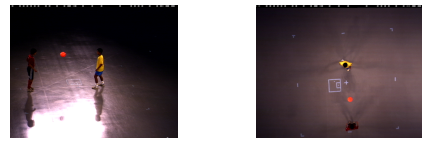
図 14 を見ると、入力形状は楕円球にもかかわらず生成された 3 次元多面体表面モデルは尖っている部分が見受けられる。これは入力画像が 3 枚であったために 3 視点の情報のみしか用いることができなかつたことと、投影の際の  $F$  行列の誤差に起因するものと思われる。誤差に関しては従来法と本手法は同様に受けるので、比較の際には同条件といえる。

従来法と本手法の形状を比較すると、本手法のほうが入力形状に近いと言える。これは従来法はボクセル空間内で一度形状復元を行ってから、物体表面を抽出しているのに対し、本手法は物体表面を直接復元しているためである。

また、生成時間に関しては、従来法に比べ本手法は非常に速い結果が得られた。これは従来法はボクセル空間内のボクセルそれぞれに関してシルエット内外判定を行っており、3 次元上での処理となるのに対し、本手法は画像上の各三角メッシュごとに多角形近似シルエットとの交点を調べているので、2 次元上での処理となっているためである。

## 4.2 実画像への本手法の適用

実際にカメラキャリブレーションを取得しないカメラで同時に撮影された 5 枚の画像に対し、本手法を適用した。入力画像は 640×480 画素の RGB24 ビット画像である。5 枚の画像のうちの 2 枚をベースカメラ画像とし、残りの 3 枚の画像から 3 次元多面体表面モデルを生成した。なお、入力画像には複数の物体が写っているが、今回は黄色の服の人物を対象物体とした。



(a) ベースカメラ 1 (b) ベースカメラ 2



(c) カメラ 1 (d) カメラ 2 (e) カメラ 3

図 15: 入力画像

実画像に対して本手法を適用することにより、図 16 の結果を得た。また、3 次元多面体表面モデルの生成時間は 0.58sec であった。

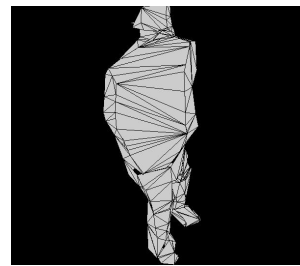


図 16: 3 次元多面体表面モデル



図 17: 図 16 にテクスチャマッピングを行ったもの

結果画像に関しては、その理論値を算出することができないため、正確な精度の分析はできないが、ある程度良好な対象人物の3次元多面体表面モデルを得られたといえる。生成時間も短く、少ない計算量で3次元多面体表面モデルを生成することができた。

また図17を見ると、入力画像が遠方から撮られたものであるためテクスチャの解像度は低いが、対象人物のシャツ、ズボンなどのテクスチャが正しい位置にマッピングされている。これは本手法により生成した3次元多面体表面モデルの3次元座標が精度よく求められていることを示しており、3次元多面体表面モデルの3次元座標を画像上で線形的に求める本手法が有効であるといえる。

## 5 おわりに

本論文ではカメラキャリブレーションを取得しないカメラで同時に撮影された画像から、それぞれのカメラ間の $F$ 行列を用いて構成されるProjective Grid Space内において、ボクセル空間での形状復元を行わずに物体表面の多面体表面モデルを直接復元する手法を提案した。

また、提案した本手法をボクセル空間で形状復元を行う従来法と比較し、本手法の有用性を示した。さらに本手法を実画像に適用して、良好な3次元多面体表面モデルを生成することに成功した。

本手法は従来法に比べ生成時間が少なく、同じ計算時間より詳細な3次元多面体表面モデルを生成できるのが特徴である。現在シルエットの多角形近似を手動により行っているが、今度これを自動化することにより、入力画像からのリアルタイムでの3次元多面体表面モデル生成を目指していきたい。

## 参考文献

- [1] 小川軍, 和田俊和, 東海彰吾, 松山隆司, “平面間透視投影を用いた並列視体積交差法”, 情報処理学会 CVIM 研究会論文誌, Vol.42, No.SIG 6(CVIM 2), pp.33-43, 2001.6.
- [2] W.Matusik, C.Buehler, R.Raskar, S.Gorlter and L.McMillan, “Image-based visual hulls”, Proc. of SIGGRAPH 2000, pp.369-374, 2000.

- [3] B.Curless, M.Levoy: “A Volumetric Method for Building Complex Models from Range Images”, *Proc. of SIGGRAPH '96*.
- [4] Michel Petrov, Andrey Talapov, Timothy Robertson, Alexei Lebedev, Alexander Zhilyaev, Leonid Polonsky: “Optical 3D Digitizers: Bringing Life to the Virtual World”, *IEEE Computer Graphics and Applications*, pp.28-37, May/June 1998.
- [5] Steven M.Seitz, Charles R.Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring”, *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp.1067-1073, 1997.
- [6] S.Vedula, P.Rander, H.Saito, T.Kanade, “Modeling, Combining, and Rendering Dynamic Real-World Events From Image Sequences”, *Proc. 4th Conf. Virtual Systems and Multimedia*, Vol.1, pp.326-322, 1998.
- [7] Peter Eisert, Eckehard Steinbach, Bernd Grid, “Automatic Reconstruction of Stationary 3-D Object from Multiple Uncalibrated Camera Views”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.10, No.2, March 2000.
- [8] H.Saito, S.Baba, M.Kimura, S.Vedula, T.Kanade, “Appearance-Based Virtual View Generation of Temporally-Varying Events from Multi-Camera Images in 3D Room”, *Computer Science Technical Report, CMU-CS-99-127*, Apr 1999.
- [9] 北原格, 大田友一, 金出武雄, “多視点映像の融合によるスポーツシーンの自由視点映像生成”, 画像の認識・理解シンポジウム (MIRU'2000) 論文集, vol.I, pp.I3-I8, 2000.7.
- [10] H.Saito, T.Kanade, “Shape Reconstruction in Projective Grid Space from Large Number of Images”, *IEEE Proc. Computer Vision and Pattern Recognition*, Vol.2, pp.49-54, 1999.
- [11] 矢口悟志, 木村誠, 斎藤英雄, 金出武雄, “未校正多視点カメラシステムを用いた任意視点画像生成”, 情報処理学会コンピュータビジョンとイメージメディア研究会論文誌, Vol.42, No.SIG06-003, pp.9-21, 2001.6.
- [12] A. Laurentini, “The Visual Hull Concept for Silhouette-Based Image Understanding”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.16, no.2, pp.150-162, 1994.
- [13] Joseph O'Rourke, “Computational Geometry in C”, Cambridge Univ Pr, 1998.