

# 共面性条件に基づく単視点画像からの3次元復元

## 3D Reconstruction from a Single View Based on Coplanarity Constraints

情報工学専攻 佐藤 駿

Information and System Engineering SATO Shun

**あらまし:** 本研究では、校正済みカメラで撮影した単視点画像から複数物体の3次元復元を行う。最初に、ユーザが画像上で物体の輪郭をなぞることで、物体ごとの3次元直交座標系を得る。次に、物体が同一平面上に存在することを条件として最適化問題を解くことにより、各物体の相対的な位置関係を計算する。現実的なシーンの画像に対して実験を行い、結果を考察する。

キーワード: コンピュータビジョン, 3次元復元, 共面性条件, 最適化問題

### 1 はじめに

デジタル画像には編集、加工、処理が容易であるという利点があるため、デジタル画像から様々な情報を取り出す技術が重要となってきた。特に、デジタル画像から3次元情報を復元する問題は、コンピュータビジョンにおける重要な課題の一つである[2]。複数枚の画像を用いるステレオマッチングでは、三角測量の原理に基づき3次元情報を復元する。一方、単視点画像から3次元情報を復元するために、画像に写る手掛かりやユーザの入力を用いるアプローチが近年増えつつある。

2013年にChenら[1]は、単視点画像からユーザの入力を基に3Dオブジェクトを抽出し、抽出したモデルを容易に編集できる3-sweepと呼ばれるツールを提案した。3-sweepでは、ユーザが画像に写る物体を幾つかのパーツに切り分け、パーツ毎になぞる操作を入力として物体の概形を推定する。さらに、パーツ同士の位置関係を制約式とした非凸最適化問題を2段階で解くことにより、物体のモデルを生成する。

本研究では、単視点画像を入力として、同一平面上に物体が存在することを利用して、物体間の相対的な位置関係を求める。本研究は校正済みカメラを使用することを仮定する。最初に、Chenらの手法[1]と同様の入力により、物体ごとの3次元直交座標系を得る。次に、物体が同一平面上に存在することを条件とした最適化問題を解くことで、各物体の相対的な位置関係を計算する。

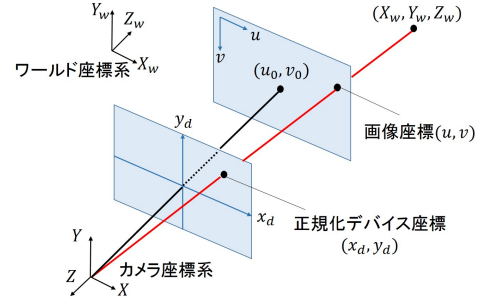


図1 透視投影モデルに基づく幾何学的関係

### 2 スクリーン座標系によるカメラ座標系の表現

#### 2.1 3次元空間と画像の関係

本研究では、射影モデルにコンピュータビジョンで最も広く用いられている透視投影モデルを用いる。透視投影モデルに基づく幾何学的関係を図1に示す。カメラ座標系  $(X, Y, Z)$  は、カメラを中心とした3次元空間を表す座標系である。カメラの光学中心を原点とし、画像の右方向を  $X$  軸、上方向を  $Y$  軸とする。  $Z$  軸はカメラの光軸方向の逆方向とする。画像座標  $(u, v)$  は画像の左上を原点とし、画像の右方向を  $u$  軸、下方向を  $v$  軸とする座標系である。

スクリーン座標系  $(x, y, z)$  は、画像の中心を原点とし、画像の右方向を  $x$  軸、上方向を  $y$  軸とする。  $z$  成分は  $[Z_n, Z_f]$  の範囲を取るデプス値であることから、画像座標と同様画像を表すのに加え、奥行き情報を持つ座標系である。ここで、  $Z_n, Z_f$  はそれぞれ透視投影を行う領域を定めるパラメータである。

3次元空間から画像への射影の過程を逆順に辿ることによって、カメラ座標  $(X, Y, Z)$  は、スクリーン座標  $(x, y, z)$  を用いて

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \frac{Nx}{z+v} \\ \frac{Ny}{z+v} \\ \frac{u}{z+v} \end{pmatrix} \quad (1)$$

のように表現される。ここで、

$$N = -\frac{Z_n Z_f}{f}, \quad u = Z_n Z_f, \quad v = -(Z_n + Z_f)$$

とする。ただし、 $f$  はカメラの焦点距離である。カメラの焦点距離は、カメラの性質を表すカメラの内部パラメータの一つである。カメラの内部パラメータと合わせて、カメラの位置や姿勢を表す外部パラメータを求め、画像を補正する処理をカメラキャリブレーションという。本研究では校正済みカメラを使用することを仮定する。カメラキャリブレーションを事前に行うことで、カメラの焦点距離  $f$  を求める。

## 2.2 カメラキャリブレーション

カメラキャリブレーションの基本的な考え方は、空間の位置と画像上の位置はカメラの内部パラメータと外部パラメータを用いて関連付けられており、予め位置がわかっている空間点とその画像上への投影点を用いてこれらのパラメータを算出することである。空間点の座標が容易に判別できるようにキャリブレーションターゲットと呼ばれるものが用いられる。

本研究では、Intel 社が開発・公開したオープンソースのコンピュータビジョン向けライブラリ OpenCV を用いてカメラキャリブレーションを実装した。OpenCV のカメラキャリブレーションは、Zhang の手法 [3] を基にしている。

Zhang の手法 [3] では、2 次元位置が既知の参照点を描いた平面（平面状のキャリブレーションターゲット）をモデル平面と呼び、ユーザはモデル平面もしくはカメラを複数の異なる位置に動かして参照点を撮影し、平面と画像面の間の射影変換を計算してカメラのパラメータを抽出する。カメラキャリブレーションに必要な画像の枚数は決まっていないが、一般的には、ブレのない適切な画像を数十枚程度用いると十分な精度が得られるとされている。本研究で用いたカメラキャリブレーションの 15 枚の入力画像を図 2 に示す。

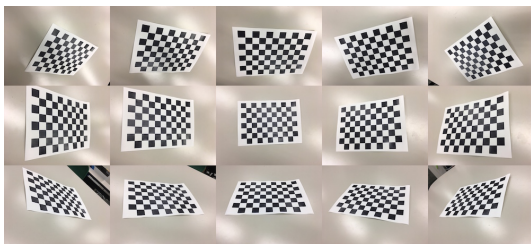


図 2 カメラキャリブレーションの入力画像

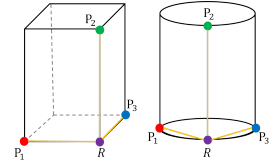
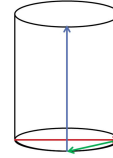
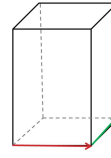


図 3 ユーザによる入力例

図 4 ローカル直交座標系

## 3 共面性条件を用いた複数物体の位置関係の推定

### 3.1 ユーザの入力

本研究では、Chen らの手法 [1] と同様、ユーザは画像に写る物体に対し、物体をなぞるような入力をする。直方体と円柱それぞれに対するユーザの入力の例を図 3 に示す。ユーザの入力は、物体毎にカメラ座標系におけるローカル直交座標系を与えるように入力される。ユーザの入力により得たローカル直交座標系とそれを成す 4 点の例を図 4 に示す。直交座標系の原点を  $R$ （紫の点）、軸端点  $P_1, P_2, P_3$ （赤、緑、青の点）とし、黄色い線分は座標軸を表す。

### 3.2 スクリーン座標における直交座標系の表現

ユーザの入力により求めた物体のローカルな直交座標系の原点  $R$  のカメラ座標を  $(X_r, Y_r, Z_r)$  とする。同様に軸端点  $P_m (m \in \{1, 2, 3\})$  のカメラ座標を  $(X_m, Y_m, Z_m)$  とする。このとき、ローカルな直交座標系はカメラ座標において直交であることから、 $\overrightarrow{RP_m} \cdot \overrightarrow{RP_n} = 0 \ (m, n) \in \{(1, 2), (2, 3), (3, 1)\}$  が成り立つため、以下の等式を得る。

$$(X_m - X_r)(X_n - X_r) + (Y_m - Y_r)(Y_n - Y_r) + (Z_m - Z_r)(Z_n - Z_r) = 0$$

さらに、式 (1) から

$$\begin{aligned} & \left( \frac{Nx_m}{z_m + v} - \frac{Nx_r}{z_r + v} \right) \left( \frac{Nx_n}{z_n + v} - \frac{Nx_r}{z_r + v} \right) \\ & + \left( \frac{Ny_m}{z_m + v} - \frac{Ny_r}{z_r + v} \right) \left( \frac{Ny_n}{z_n + v} - \frac{Ny_r}{z_r + v} \right) \\ & + \left( \frac{u}{z_m + v} - \frac{u}{z_r + v} \right) \left( \frac{u}{z_n + v} - \frac{u}{z_r + v} \right) = 0 \end{aligned}$$

を得る。これら 3 つの式から、 $z_1, z_2, z_3$  は  $z_r$  を用いて以下のように表現することができる。

$$z_m = \alpha_m z_r + (\alpha_m - 1) v \quad (2)$$

ただし、 $\alpha_m$  はスクリーン座標  $x, y$  とパラメータ  $N, u$  を用いて表現される。したがって、原点  $R$  のスクリーン座標の第 3 成分  $z_r$  を計算することができれば、軸端点  $P_m$  のカメラ座標  $(X_m, Y_m, Z_m)$  ( $m = 1, 2, 3$ ) を

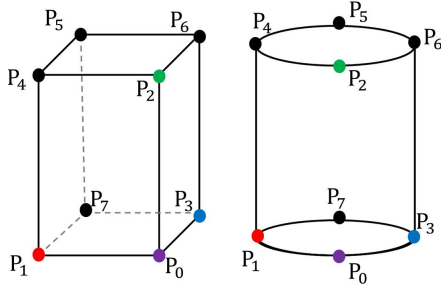


図5 ユーザが入力する4点と新たに生成した4点

式 (1) から求めることができる。

### 3.3 共面性条件

本研究では、複数物体が同一平面上に存在するという条件 (共面性条件) を利用して複数物体の相対的な位置関係を求める。同一平面上に存在する点のカメラ座標  $(X, Y, Z)$  について以下の平面の方程式が成り立つ。

$$aX + bY + cZ = 1$$

ここで、様々な面を表現するために、軸端点の線形結合によりローカル直交座標系において直方体を成すような点  $P_4, P_5, P_6, P_7$  を生成する。ローカル直交座標系の原点  $R$  を  $P_0$  とし、軸端点  $P_1, P_2, P_3$  にこれらの点を加え、総じて端点と呼ぶ (図5 参照)。この端点を用いてユーザはどの平面にどの端点が存在するかという入力を追加する。

### 3.4 物体間の位置関係を推定する最適化問題

平面の集合を  $M = \{0, \dots, m\}$ 、物体の集合を  $O = \{0, \dots, n\}$ 、端点の添字番号の集合を  $K = \{0, \dots, 7\}$  とする。  $i \in M, j \in O, k \in K$  に対し、平面  $i$  に物体  $j$  の端点  $P_k$  が存在するときの  $(i, j, k)$  の集合を  $S$  とする。  $X_{j,k}, Y_{j,k}, Z_{j,k}$  は、物体  $j$  の端点  $P_k$  のカメラ座標であり、式 (1) と式 (2) から物体  $j$  の原点のスクリーン座標の第3成分  $z_j$  を変数として表現される。  $a_i, b_i, c_i$  は平面  $i$  の方程式を決定する変数である。  $e_{i,j,k}$  は共面性条件に対する誤差を表す変数であり、目的関数では誤差の二乗和を最小化する。このとき、各物体における原点  $R$  の第3成分  $z_r$  を決定する問題は、以下の線形制約付き凸二次関数最小化問題として記述できる。

$$\begin{aligned} \min. \quad & \sum_{(i,j,k) \in S} e_{i,j,k}^2 \\ \text{s.t.} \quad & e_{i,j,k} = a_i X_{j,k} + b_i Y_{j,k} + c_i Z_{j,k} - 1 \\ & ((i, j, k) \in S) \\ & Z_n \leq Z_{j,k} \leq Z_f \quad (j \in O, k \in K) \end{aligned}$$

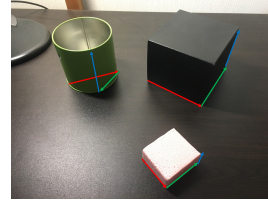


図6 入力画像

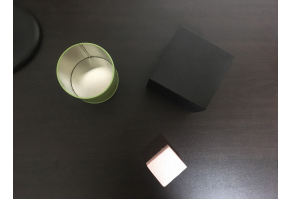


図7 真上からの写真

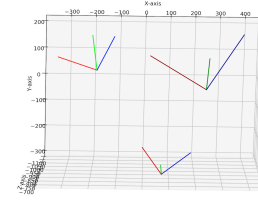


図8 実験結果

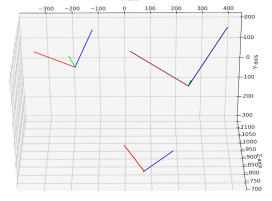


図9 実験結果 (真上)

一つ目の制約式は共面性条件を表している。二つ目の制約式は入力したすべての物体について、端点のスクリーン座標の第3成分を透視投影を行う領域内に収めるための制約である。

## 4 評価実験

iPhone6S (Apple 社) 内蔵のカメラを使用して撮影した写真を  $1/4$  に縮小した  $1008 \times 756$  画像もしくは  $756 \times 1008$  画像を入力とする。透視投影変換で用いるパラメータは  $Z_n = 10, Z_f = 1000$  と設定した。2.2節のカメラキャリブレーションを行った結果、焦点距離は  $f = 872$  となった。最適化問題の計算には数理計画ソルバー IBM ILOG CPLEX 12.6 を使用した。

最初に、机にある3つの単純な物体が写る画像に対して実験を行った。入力画像とユーザ入力を図6に、入力画像のシーンを真上から撮影したものを図7に示し、机を平面とした共面性条件を含む最適化問題を解いて得た各物体の直交座標系を図8に、この直交座標系を真上から見たものを図9に示す。さらに、ピンクの直方体を物体1、黒い直方体を物体2、円柱を物体3とする。物体1の原点を  $R_1$ 、軸端点を  $P_1, P_2, P_3$  とし、物体2と物体3の原点を  $R_2, R_3$  とする。このとき、線分  $R_1 P_2$  の長さを1としたときの各線分の長さを表1に示す。推定値と実測値はほぼ一致しており、物体の寸法

表1 線分  $R_1 P_2$  を基準とした場合の各線分の比較

	$R_1 P_1$	$R_1 P_3$	$R_1 R_2$	$R_2 R_3$	$R_3 R_1$
推定値	1.51	1.61	5.25	4.54	5.87
実測値	1.50	1.60	5.33	4.67	6.17





図 10 入力画像 (家)



図 11 Google Maps

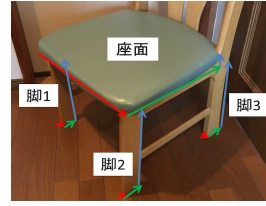


図 14 入力画像 (イス)

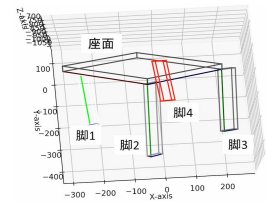


図 15 実験結果

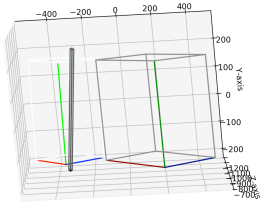


図 12 実験結果

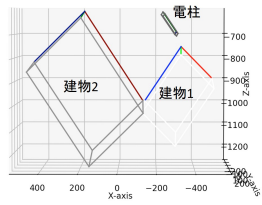


図 13 実験結果 (真上)

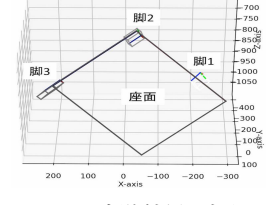


図 16 実験結果 (真上)

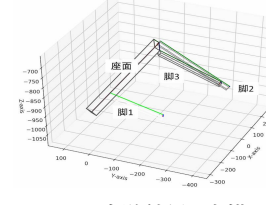


図 17 実験結果 (真横)

と物体同士の位置関係を再現できていることがわかる。

より現実的なシーンについて考察するため、2 軒の建物と 1 本の電柱が写る画像に対して実験を行った。建物を直方体、電柱を円柱とみなし、各物体が地面に接地することを用いる。入力画像とユーザ入力を図 10 に、Google Maps での位置関係を図 11 に示し、実験結果を正面から見たものと真上から見たものを図 12 と図 13 に示す。電柱がやや傾いているのは、電柱の領域が狭いため、ユーザの入力が正確に行えていないことが原因であるが、建物と電柱の位置関係はほぼ再現できている。

最後に、複数の面を持つイスが写る画像に対して実験を行った。イスの脚、座面を直方体とみなし、各脚が床と座面に接触することを共面性条件として用いた。入力画像とユーザ入力を図 14 に示し、実験結果を正面、真上、真横から見たものを図 15、図 16、図 17 に示す。他の画像と同様、位置関係をほぼ再現できているが、図 16 から脚 1 が本来あるべき座面の四隅よりも内側に位置づけてしまっている。また、図 17 から、脚 1 が他の脚に比べて斜めに短くなってしまっている。これらは、斜めに撮影していることから、脚 1 が座面で隠れてしまっていることが原因である。

図 15 に写る赤い直方体は、イスの脚は等間隔に取り付けられているという仮定の下、画像では写らない脚 4 を他の脚の情報に基づいて複製したものである。3 次元空間を表すカメラ座標を求められているため、今回の例だけでなく任意の場所に複製したり、スケールを変えたりといった様々な操作が可能である。

## 5 結論

本研究では、単視点画像から複数物体の 3 次元復元を行った。画像をなぞるユーザの操作から物体ごとのローカル直交座標系を求め、共面性条件を用いて最適化問題を解くことでそれらの位置関係を推定した。

最初に単純な物体が写る画像に対し実験を行い、物体の寸法・位置関係の実測値と推定値がほぼ一致することを確認した。次に現実的なシーンの画像について実験を行った。単純な物体での実験よりも画像に写る物体の領域の差が大きいため、ユーザの入力が正確でない場合に座標系が傾くなどの事例が生じやすくなるが、位置関係自体は再現できていることを確認した。また、イスの例を用いて、単純な操作で画像に写らない物体を複製できることを示した。その他にもスケールを変えるとといった操作も実装可能であるため、3D コンテンツの作成などの様々な応用に繋がることが期待される。

## 参考文献

- [1] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, 3-sweep: Extracting editable objects from a single photo, *ACM Trans. Graph.*, vol. 32, pp. 1–195, 2013.
- [2] R. Szeliski, コンピュータビジョン —アルゴリズムと応用, 玉木徹ら (訳), 共立出版社, 2013.
- [3] Z. Zhang, A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, 2000.