

RGB-Dカメラを用いた物体の体積推定手法の構築

A Method of Volume Measurement from Point Data Obtained by RGB-D Cameras

齊藤 圭佑[†]
Keisuke Saito

酒井 道[‡]
Osamu Sakai

宮城 茂幸[‡]
Shigeyuki Miyagi

1 はじめに

2010年にMicrosoft社がコンシューマ向けデブスカメラを含むセンサーデバイス(Kinect)を発売して以来、Kinectは本来のゲームコントローラとしてだけでなく、物体の3D復元[1, 2]、シーン解析[3]、物体検出[4]、あるいは動作解析[5]といった様々な分野で応用されている。

本稿ではデブスカメラの応用としてKinectを用いた体積推定システムの構築について述べる。体積推定は、流通における梱包物の大きさ推定[6]だけでなく、福祉分野の浮腫部位の体積推定への応用も考えられる。これまでも身体の形状計測手法は提案されているが[7]、測定時間がかかることや、測定装置のコストが増大することから、依然看護現場では浮腫の計測には巻尺による周囲径測定が用いられている。また、正確な浮腫体積の計測には容積計が必要である[8]。ところがコンシューマ向けデブスカメラの登場により、非接触、高速かつ低価格で計測システムの構築が可能になってきた。現在市販の3Dスキャナも存在するが、カメラを測定対象の周囲で動かしながら多数回撮影する必要があり、比較的長い計測時間が必要となる。浮腫部位の計測においては、患者への負担を減少させることも重要であって、特に在宅での自己管理を想定すると短時間で簡単に計測が完了するシステムが望ましい。そこで、図1に示すような2台のみのKinectを用いた体積推定システムの枠組みを提案する。対向する2台のKinectの中央付近に測定対象を設置し、得られた点群データから体積を推定することを想定している。

KinectにはデブスカメラとRGBカメラが同時に搭載されているので、両面に同じパターンが描かれた較正板を用いることにより、2台のRGBカメラの外部パラメータを推定する。2節において、これらを含めたキャリブレーション方法を簡単に説明する。3節では2台のデブスカメラから得られた点群データを1つのワールド座標系上に表現し、そのデータにボクセル化手法を適用した体積推定手法について述べる。Kinectでは、デブスカメラはRGBカメラの直近に置かれていることから、2節で得られた外部パラメータを利用し、2台のKinectのカメラ座標から得られる点群データを1つのワールド座標系上での座標点へ変換する。4節では、既知の物体に対して上述の方法を適用し体積推定を行った結果を示す。

[†]滋賀県立大学大学院, Graduate school of The University of Shiga Prefecture

[‡]滋賀県立大学, The University of Shiga Prefecture

また、節5では提案方法による推定結果の精度について検討を行う。

2 RGBカメラのキャリブレーション

カメラモデルとしてピンホールカメラモデルを考える。ある3次元座標系(以下ワールド座標系)の点 $M = (X, Y, Z)^T$ が透視投影変換を用いて画像平面上的点 $m = (u, v)^T$ に射影される時、 M , m に対応する拡張ベクトル $M' = (X, Y, Z, 1)^T$, $m' = (u, v, 1)^T$ を用いると式(1)のように表される。

$$sm' = A[R | t]M', \quad (1)$$

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, [R | t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}. \quad (2)$$

である。 A はカメラの内部パラメータ行列であり、 (c_x, c_y) , f_x , f_y はそれぞれ主点(光軸と画像平面の交点, 通常は画像中心)および X 方向, Y 方向に対応するピクセル単位で表される焦点距離である。 $[R | t]$ はカメラの外部パラメータ行列であり、 R が回転行列, t が平行移動ベクトルを表す。式(1)より、 A が既知であれば、複数の画像座標とワールド座標系の対応点(最低4組)を与えることで、外部パラメータ行列 $[R | t]$ を推定できる。

2.1 画像座標の対応点の検出

較正板を撮影したRGB画像より特徴点を検出し、ワールド座標系との対応付けを行う。図2に較正板の仕様を示す。較正板の四隅をそれぞれ赤色, 青色, 緑色, 黄色に着色する。較正板の厚さは5mmで、裏側にも同様の着色を行う。これを撮影したRGB画像をHSV色空間へと変換し、各色に対して色相H, 彩度S, 明度Vのしきい値をそれぞれ設けることで、各着色領域を抽出する。

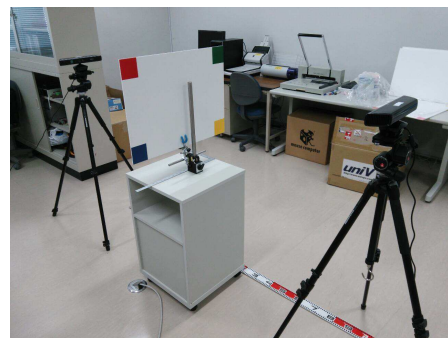


図1 測定の概要図

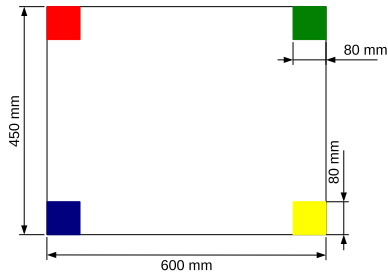


図 2 較正板の仕様

本稿では色抽出に用いるしきい値は実験的に求めた値を使用する。また、色抽出後の画像にはノイズとして細かい領域が抽出される場合があるため、ラベリング処理を用いて面積が最大の領域を取り出すことで、較正板の着色領域のみを抽出する。最後に抽出した領域の重心座標を計算し、これを画像座標の対応点とする。

2.2 外部パラメータ行列の推定

RGB カメラの内部パラメータ行列および歪み係数はキャリブレーションプログラムを用いてあらかじめ求めておく。キャリブレーションプログラムには GML C++ Camera Calibration Toolbox[10] を用いた。これは Chang の手法 [11] によるキャリブレーションを OpenCV をベースに実装したものである。

次に 2.1 で求めた画像座標およびこれに対応するワールド座標系の点の組によってカメラの外部パラメータ行列を推定する。本稿ではワールド座標系の原点は較正板の中心とする。較正板の仕様から、検出する画像座標に対応するワールド座標を設定する。外部パラメータ行列の推定は画像処理ライブラリ OpenCV[9] の solvePnP 関数を用いて行った。この関数は、内部パラメータ行列、歪み係数と共に、ワールド座標とそれに対応する画像上の投影点を与えるとき、再投影誤差を最小にするような外部パラメータ行列を推定する。

3 体積推定手法

それぞれのカメラから得られた深度画像より、深度カメラ座標系における 3 次元座標点群を計算することができる。このとき、RGB カメラと深度カメラのカメラ座標系が同一であるとみなせば、RGB カメラのキャリブレーションで求めた外部パラメータにより、得られた点群をワールド座標系の点に変換することができる。これにより、複数のカメラから得られる点群を共通の座標系に統合する。その後、点群のボクセル化を行い、ボクセル数の計数により体積推定とする。

3.1 深度画像の取得と 3 次元点群の計算

各カメラより測定対象の深度画像が得られる。本稿では深度画像取得の際に、Kinect の開発ツールキットである Kinect for Windows SDK[12] の NuiImageGetColorPixelCoordinatesFromDepthPixelAtResolution 関数を用いて深度画像と RGB 画像間の座標合わせを行う。その後、深度画像より深度カメラ座標系での 3 次元座標

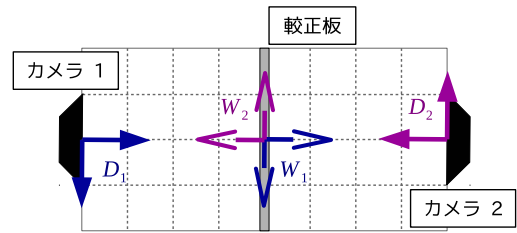


図 3 カメラ座標系とワールド座標系の関係

点群を計算する。透視投影変換の関係より、ある画像座標 (u_d, v_d) およびその深度値 w に対して、3 次元座標点 (X_d, Y_d, Z_d) は式 (3) で計算できる。 f はスケール係数を含む焦点距離、 (u_{d0}, v_{d0}) は深度画像の画像中心を表す。本稿では $f = 571.3$, $(u_{d0}, v_{d0}) = (320, 240)$ として計算を行う。

$$\begin{aligned} X_d &= (u_d - u_{d0})/f \times w, \\ Y_d &= (v_d - v_{d0})/f \times w, \\ Z_d &= w. \end{aligned} \quad (3)$$

3.2 外部パラメータを用いた座標系変換

各カメラについて求めた回転行列および平行移動ベクトルを用いて、深度カメラ座標系 D_1 および D_2 に得られる 3 次元点群をそれぞれワールド座標系 W_1 および W_2 に変換する (図 3)。RGB カメラのキャリブレーションにより求めた外部パラメータ行列は、ワールド座標系からカメラ座標系への変換を表す。よって、ここでは逆変換を行う必要がある。カメラ座標系の 3 次元点群から平行移動ベクトルの値を差し引き、続けて回転行列の逆行列との積をとる。それぞれのワールド座標系の関係は、 X 軸および Z 軸を反転したものである。よって、一方の点群について X および Z 要素の符号を反転させることで共通の座標系に統合する。

3.3 点群データのボクセル化

統合された点群データに対して、測定対象面および内部のボクセル化を行う。ボクセル化とは与えられた点群を、体積表現としてボクセルグリッド上に近似することである。ボクセルグリッドの大きさは点群全体に対して十分大きく、しかし非効率性を避けるために大きすぎないように決定する。ボクセル 1 つの大きさはボクセル表現の分解能と精度に関わる。精度と処理時間との最良のトレードオフとして決定する必要がある。

ボクセル化には Garcia と Ottersten の手法 [13] を改良した手法を利用する。Garcia らは不完全な点群データに対する表面ボクセル化、内部ボクセル化の手法を提案しており、内部ボクセル化に関しては表面ボクセルの補完、内部ボクセルの設定の順に処理している。

3.3.1 対象表面のボクセル化

求められた点群データは測定対象の表面のみを表している。3 次元のボクセルグリッド上にマッピングする

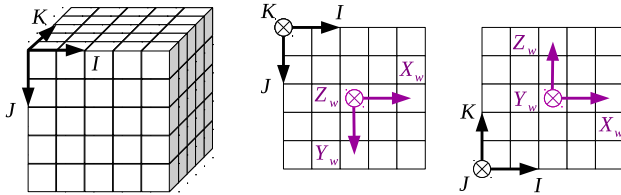


図4 ボクセルグリッドとワールド座標系の関係

ここで、対象表面のボクセル化を行う。ワールド座標系 $X_w Y_w Z_w$ をボクセルグリッド IJK の中心に想定する(図4)。各点にワールド座標系原点をボクセルグリッド原点に移動させるようなベクトルをオフセットとして加える。その後、対応するボクセルの幾何学的中心に近似する。

3.3.2 表面ボクセルの補完

表面ボクセルの補完処理を行うために切り出し画像を作成する。切り出し画像はボクセルグリッドにおいてある軸要素の値に合致する値をもつボクセルのみを取り出したものである。 IJ 平面での切り出し画像を考える場合は、ある K 軸要素の値を持つボクセルのみを取り出し、その I および J 要素を画像座標として、画素値を255に設定する。これは表面ボクセルの存在を表す。 K の全要素について走査し、それぞれ切り出し画像を作成する。

続けて、得られた切り出し画像に対して処理を行い、欠損などによって不完全な表面ボクセルの補完を行う(図5)。まず、切り出し画像中の表面ボクセルの画素位置を相対距離順にソートする(図5(b))。ソートは開始点から最も相対距離の短いボクセルを順に配列に格納していくことで行う。相対距離には2つの近傍ボクセルが許容される最大の距離として、しきい値 ε が与えられる。 ε よりも相対距離の短いボクセルが存在しなくなった場合は、開始点に戻り処理を再開する。再び ε 以下の近接ボクセルが存在しなくなった場合は、ソートを終了し配列を保存する。残った表面ボクセルを別の配列に格納しながら、同様のソートを継続する。全ての表面ボクセルが配列に格納されれば、ソート処理は終了する。続いて、作成された配列を順に走査し、ソートされた表面ボクセル間に存在するボクセルを新たな表面ボクセルとして設定する(図5(c))。

3.4 内部ボクセルの設定

Garcia らは表面ボクセルを補完後、隣接するボクセルにその状態を伝播するアルゴリズムによって内部ボクセルの設定を行っている。しかし、表面ボクセルの状態によってはこのアルゴリズムはうまく動作せず、望まない結果を得る。本稿では、ラベリング処理を用いた内部ボクセルの設定手法を提案する。

表面ボクセルの補完により、内部領域は画像の外周部分には存在しない。よってラベリング処理を利用して候補領域の判別を行う(図5(d))。切り出し画像の画素

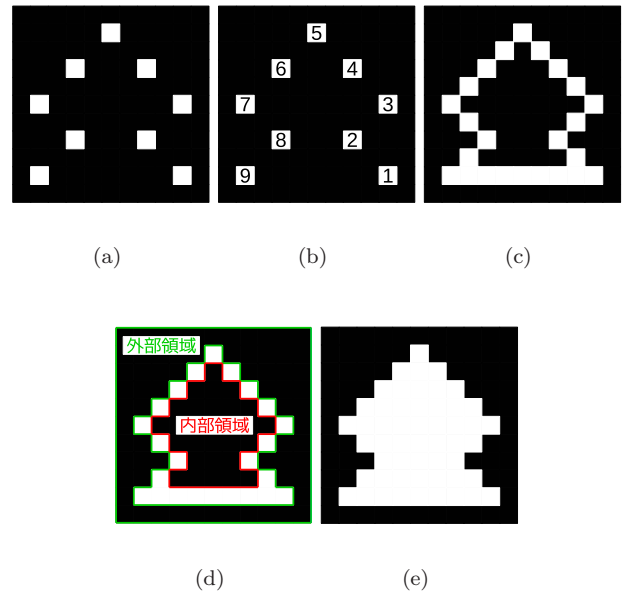


図5 ボクセル化処理の流れ。(a):切り出し画像の例。(b):表面ボクセルを相対距離順にソートする。(c):ソートされた表面ボクセル間に存在するボクセルを補完する。(d):ラベリングにより領域の判別を行う。画像外周に達している緑色領域は外部領域、そうでない赤色領域は内部領域とする。(e):内部領域を新たなボクセルに設定する。

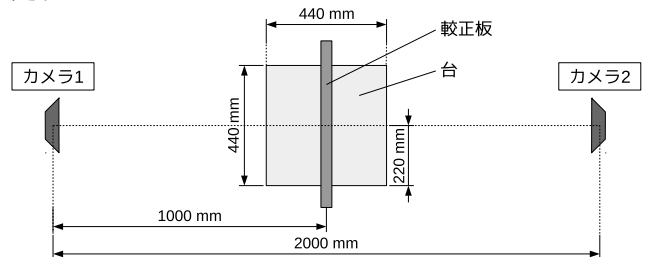


図6 各機器の配置図

値を反転した画像に対して、ラベリング処理を行うことで、候補領域を決定できる。候補領域のうち、領域の左上画素または右上画素が画像の外周部分に達しているものは物体の外部とし、それ以外は内部と判定する。切り出し画像において、内部と判定された領域を、物体が存在するボクセルとして設定する(図5(e))。この処理を全ての切り出し画像に対して行う。

4 評価実験

既知の大きさの箱を対象にして、体積推定の実験を行う。実験時の機器配置を図6に示す。

測定対象は直方体の箱とした。ただし、今回の実験では、測定対象以外の背景部分も点群に含まれてしまうため、測定対象の一部分に着色を行うことで、RGB画像を用いて測定対象領域を抽出し、その部分のみ点群として計算する(図7)。体積推定部分は $15\text{cm} \times 12\text{cm} \times 7\text{cm}$ となる。

測定対象の撮影により得られた点群データに対してボクセル化を行う。ボクセル化に関して、表1に示す条件

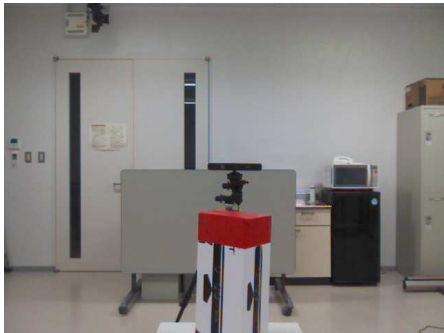


図7 対象撮影時のRGB画像。赤色部分が体積推定対象である。

でそれぞれ実験を行う。またその結果を表2に示す。ただし、今回の実験では、測定対象の全周について点群を取得することができていないため、ボクセル化が可能な平面での切り出し画像のみから評価している。

表1 ボクセル化に関する実験条件

条件	ボクセルグリッドが表現する領域	グリッドの分解能	近傍ボクセルの許容距離 ϵ
(1)	250 mm × 250 mm	10 mm	5
(2)	250 mm × 250 mm	5 mm	10
(3)	250 mm × 250 mm	1 mm	50

表2 体積推定の実験結果

条件	ボクセル数 [個]	体積推定結果 [cm ³]	実際の体積 [cm ³]
(1)	1707	1707	
(2)	11272	1409	1260
(3)	1228093	1228.093	

5 考察

実験結果より、測定対象の実際の体積とその推定結果には誤差が存在する。ボクセルの分解能が10 mm, 5 mmの場合には、実際よりも大きく推定され、1 mmの場合には実際よりも小さく推定される。この結果より、RGB画像による姿勢推定が不完全である可能性が考えられる。姿勢推定が不完全であると座標系変換に誤差が生じ、測定対象を表す点群が正確さを失う。また、ボクセルの分解能が粗い場合は、余分に推定される部分が生じ、推定結果が実際より大きくなる原因になると考えられる。分解能が1 mmの場合は、カメラの取得できる点群の精度に対して分解能が細かすぎるため、正しくボクセル化されない部分が存在し、推定結果が小さくなると考えられる。このことから、ボクセルの分解能などのパラメータをカメラや測定対象に応じて適切に設定する必要があるといえる。

6 おわりに

本稿ではRGB-Dカメラ2台を用いた物体の体積推定手法について検討した。測定対象の形状により、2台のカメラにより対象全体の点群が得られないため、手法を適用できない場合が存在することや、推定結果の不正確

さが問題である。今後は、姿勢推定の正確さの向上や処理の見直しにより推定精度の向上を図る。

参考文献

- [1] Alexander Weiss et al., "Home 3D Body Scans from Noisy Image and Range Data," *IEEE International Conference on Computer Vision 2011*, pp. 1951–1958, Nov. 6–13, 2011.
- [2] Jing Tong, et al., "Scanning 3D Full Human Bodies Using Kinects," *IEEE Trans. on Visualization and Computer Graphics*, 18(4), pp. 643–650, April., 2012.
- [3] Richard A. Newcombe et al., "KinectFusion: Real-Time Dense Surface mapping and Tracking," *10th IEEE International Symposium on Mixed and Augmented Reality 2011*, pp. 127–136, Oct. 26–29, 2011.
- [4] Lu Xia, et al., "Human Detection Using Depth Information by Kinect," *International Workshop on Human Activity Understanding from 3D Data 2011*, pp. 15–22, Jun. 24, 2011.
- [5] Lulu Chen, et al., "A survey of human motion analysis using depth imagery," *Pattern Recognition Letters*, 34(15), pp. 1995–2006, Nov., 2013.
- [6] Beatriz Quintino Ferreir et al., "VOLUMNECT—Measuring Volumes with Kinect," *Proceeding of SPIE*, Vol. 9013, pp. 901304-1–901304-4, 2013.
- [7] 山内航一郎 他, "多視点距離画像を用いた人体の三次元形状計測," 電子情報通信学会論文誌, *D-II*, 情報・システム, *II-パターン処理*, J88-D-II(8), pp. 1564–1572, Aug., 2005.
- [8] 中村隆夫 他, "浮腫評価のための体肢容積計の開発," 岡山大学医学部保健学科紀要, 14(2), pp. 31–35, 2003.
- [9] OpenCV, <http://opencv.org/>
- [10] Graphics and Media Lab, <http://graphics.cs.msu.ru/en/node/909>
- [11] Zhengyou Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), pp. 1330–1334, Nov., 2000.
- [12] Kinect for Windows SDK, <http://www.microsoft.com/en-us/kinectforwindows>
- [13] Frederic Garcia, Bjorn Ottersten, "CPU-Based Real-Time Surface and Solid Voxelization for Incomplete Point Cloud," *22nd International Conference on Pattern Recognition*, pp. 2757–2762, Aug., 2014.