

RGB-D カメラ映像からのリアルタイム平面領域抽出による 空間型 AR の実現

佐野 真之^{1,a)} 松本 一紀^{1,b)} 齋藤 英雄^{1,c)}

概要: 近年、動いている実物体の位置や形状に合わせてプロジェクタで映像投影を行う動的な空間型 AR の研究がなされている。RGB-D カメラ映像から得られる 3 次元点群同士の位置合わせによって物体の位置・姿勢追跡を行う際、立方体のような形状に特徴のない物体の場合では追跡の精度が安定しないという問題がある。そこで本論文では RGB-D カメラ映像から得られる 3 次元点群による立方体の位置・姿勢追跡の精度向上を目指し、立方体に動的な空間型 AR を行う手法を提案する。本手法では 3 次元点群を平面領域に分割し、対象となる立方体の面の領域ごとにモデルフィッティングを行うことで立方体の位置・姿勢を追跡し、プロジェクタで面に映像を投影することで実現する。

1. はじめに

近年、実物体に対してプロジェクタでコンピューターグラフィックス (Computer Graphics: CG) を投影する技術、空間型拡張現実感 (Augmented Reality: AR) の研究が盛んに行われてきた。空間型 AR では実環境と仮想空間を同時に目で見て臨場感のある体験ができるため、広告やゲームに用いられるほか、投影される物体の見え方を投影される映像によって様々な変化させられるため製品デザインの試作にも用いられている。動く実物体に対して空間型 AR を実現するためには投影される実物体とプロジェクタ間の位置関係を動的に把握する必要がある。空間型 AR が行われる環境は一般的に暗室で行われ投影光の影響を強く受けやすいことや、投影対象が動くことで物体の投影された見え方が変化してしまうことから RGB 情報による対象物体の位置・姿勢追跡は困難である。

RGB-D カメラ映像のデプス情報から得られる 3 次元点群を利用した動的な空間型 AR のための物体追跡を行った手法として D. Kobayshi らの手法 [1] があげられる。投影対象物体の 3 次元点群モデルが持つ Fast Point Feature Histograms (FPFH) 特徴量とオンラインで検出された対象物体の 3 次元点群が持つ FPFH 特徴量をマッチングさせることで大まかな位置合わせを行い、Iterative Closest Point (ICP) アルゴリズムによって高精度な位置合わせを

行う。このように 3 次元点群の細かな位置合わせでは主に ICP アルゴリズムが用いられるが、ICP アルゴリズムでは最近傍点を対応点として変換を行うため、平面や立方体のような形状に特徴のない物体では点群同士の対応がうまく取れず位置合わせの精度にばらつきが出てしまう。空間型 AR では実際に投影される実物体がディスプレイとなるため精度が不安定で投影映像がぶれてしまうと、空間型 AR の臨場感のある体験という利点が失われてしまう。

また立方体に対して空間型 AR を行う研究として Y. Tang らの手法 [2] があげられる。これは立方体の面にディスプレイを取り付けた立方体型ディスプレイ [3] におけるディスプレイの代わりとして各面にプロジェクタで投影することによってユーザー視点からは立方体の中にまるで仮想空間が実在するかのように見える、というものである。この手法ではユーザーの 3 次元位置を追跡することでユーザーから立方体の中の仮想空間が歪みなく見えるような投影映像をレンダリングしているが、立方体は手で持たずに固定されているという条件がある。より自然なインタラクションを得るためには立方体の位置・姿勢追跡をすることで実際に手で動かしながら立方体の仮想空間を体験できる必要がある。

そこで我々は、RGB-D カメラ映像から得られるデプス情報を用いた立方体の位置・姿勢追跡における精度の向上を目指し、立方体面への動的な空間型 AR を実現する手法を提案する。本研究の構成は以下の通りである。2 章で我々の提案する手法について述べる。3 章で本システムの精度評価実験とプロジェクタによる投影実験の結果を示し、4 章で結論を述べる。

¹ 慶應義塾大学理工学研究科 〒223-8522 神奈川県横浜市港北区目吉 3-14-1

a) sano@hvrl.ics.keio.ac.jp

b) kazuki@hvrl.ics.keio.ac.jp

c) saito@hvrl.ics.keio.ac.jp

2. 提案手法

本手法は、RGB-D カメラ映像から得られる 3 次元点群を Superpixel Segmentation とラベリングを用いて平面領域に分割した後、その中から投影対象となる立方体の面の領域部分を抽出する。そして得られた平面領域ごとにモデルフィッティングを行うことで立方体の位置・姿勢追跡を行う。さらにユーザー位置、プロジェクタ位置と合わせることでユーザーから CG が立方体内に存在するような映像をレンダリングしそれぞれの面への投影画像を生成する手法を提案する。図 1 に提案手法の概要を示す。

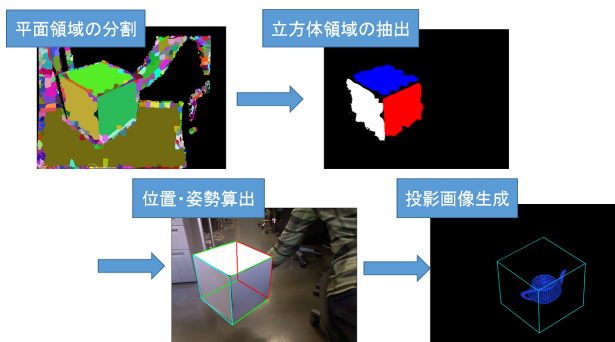


図 1 提案手法の概要

2.1 平面領域の分割

我々はまず RGB-D カメラ映像から得られた 3 次元点群を Joint Bilateral Filter[4] によりエッジの保存したデプス値の平滑化を行い、S.Holzer らの手法 [5] を用いて法線マップを作成した。ただし境界付近やデプス値の差が大きい部分においては S.Holzer の手法は適用することが出来ないため、ある注目画素とその近傍点 2 点との外積を計算することで法線ベクトルを算出した。次に我々は Depth Adaptive Superpixels[6] を改良した Normal Adaptive Superpixels を適用する。まず画像中で定期的にサンプリングを行った後、K-means アルゴリズムを注目画素 p とその近傍にあるクラスター k との距離 $D_k(p)$ に基づいて繰り返し行うことで距離画像を平面小領域を表す Superpixel に分割する。

$$D_k(p) = \frac{w_s D_{k_s}(p) + w_d D_{k_d}(p) + w_n D_{k_n}(p)}{w_s + w_d + w_n} \quad (1)$$

ここでの s, d, n は画素位置、デプス値、法線ベクトルを表し、各 w は各画素距離 D_{k_s} 、ユークリッド距離 D_{k_d} 、法線ベクトルの角度差 D_{k_n} に対する重みである。この時の領域分割と同時に、領域の平面方程式を算出する。領域内の点群の法線ベクトルの平均を平面の法線ベクトル、点群の 3 次元座標の平均値を平面に存在する代表値として平面と原点の距離を求めた。この領域分割では最初のサンプリングの数に応じて広範囲を平面領域も細かく分割するので、

K.A. Hawick らのラベリング [7] を用いて領域の統合を行う。ある注目領域と隣接する領域が平面方程式、中心位置を用いて幾何学的に同じ領域であるか、かつ領域内の法線ベクトルの分散から曲面なのか平面であるかを判定することによって同じ平面であると判定された領域は統合を行う。これによって統合された結果は図 3 のようになった。曲面上と認定された領域は統合されず細かい平面領域の集合で表され、平面領域のみ統合されていることがわかる。また一度細かい平面領域に分割することで立方体を持つ手と立方体の面を区別できたため統合された際にも別領域として分割されていることが確認できる。

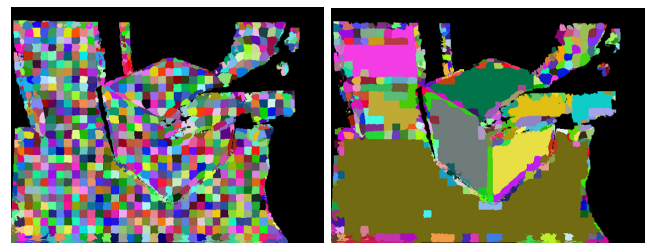


図 2 Superpixel Segmentation 図 3 ラベリングによる統合

2.2 立方体領域の抽出

平面領域に分割した後、投影対象となる立方体の面の領域を抽出する必要がある。最初のフレームでは、立方体の 3 つの面が RGB-D カメラから見えている状態で、全平面領域の中からお互いの法線が直交し、領域中心と法線ベクトルから推定される立方体の中心位置が近傍にあることを条件に 3 つの領域の探索を行うことで立方体の面の抽出を行う。これ以降では前フレームで抽出された立方体の面領域と前フレームの立方体の位置・姿勢をもとに探索を行う。前フレームにおける立方体の面領域 j と現フレームで分割された平面領域 k を式 1 に領域の大きさの情報に加えた距離 $D_k(j)$ を算出することで同じ平面領域であるかどうかのフレーム間での対応付けを行うことで抽出をする。しかし 2 面見えていた状態から 3 面見えるようになる場合や、フレーム間の対応付けに失敗する場合は考えられる。この時、抽出できなかった面に対しては、前フレームでの立方体の位置・姿勢から各面がどこに存在していたのかを推定できるため、同様に推定された面 j と現フレームでの平面領域 k の距離 $D_k(j)$ を計算することで同じ平面であるかどうかを判定し抽出を行う。これらにより立方体のどの面が現フレームで抽出されているのかを取得することが出来る。

2.3 位置・姿勢算出

抽出された立方体の平面領域内に存在する 3 次元点群を用いて位置・姿勢を算出する。まず大まかな位置合わせとして、立方体の位置は 3 平面の交点、姿勢は平面の法線ベクトルを用いて算出する。さらに本手法では、細かな位置

合わせのために立方体の面領域ごとにモデルフィッティングを行うことで位置姿勢を算出する。前項により抽出された領域と合わせたい立方体モデルの面の対応を取れたので、領域内に存在する3次元点群と立方体モデルの面とのユークリッド距離の総和が小さくなるように最急降下法を用いてフィッティングを行う。図4のようにこれを3つの面で行うことで立方体位置の細かな位置合わせを行う。しかし立方体面が3つ抽出されている場合は位置・姿勢を一意に算出することができるが、2面しか見えていない場合には、見えていない面の法線方向にモデルフィッティングを行っていない。ここで見えていない面の法線ベクトルは他の2面の外積ベクトルに当たるのでこの方向に立方体のエッジとなる位置の検出を行う。まず抽出された2平面でモデルフィッティングを行った後、図5のように等間隔にサンプリングした点から外積ベクトル方向にデプス値が急激に変化する点を探しエッジであるとする。数点からエッジの探索を行うことで見つからなかった場合や、手などのオクルージョンによって間違った位置に検出された値を他の推定値と比較することで除外し、残った値の平均を取ることで見えていない面方向へのフィッティングを行う。

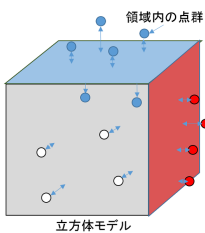


図4 3面による推定

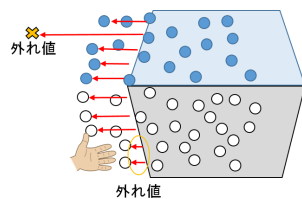


図5 2面による推定

2.4 投影画像生成

立方体の位置・姿勢とユーザーの位置と合わせることで、ユーザーから立方体はどう見えるかをレンダリングし、立方体の各面に投影すべきテクスチャを作成する。RGB-Dカメラとプロジェクタの内部、外部パラメータによってプロジェクタ視点からの立方体位置を算出しテクスチャマッピングを行うことで投影画像を生成する。これを実際に投影することでユーザーの視点からは立方体の中に仮想空間があるように観測することができる。

3. 実験

提案手法の有効性を示すために我々は実際に手に取って一辺25cmの立方体を動かしたときどの程度の精度で位置・姿勢追跡が行えていたのかに関する評価実験を行った。また提案手法によって得られた立方体の位置・姿勢に対してプロジェクタで投影した結果を示す。

実装環境は以下の通りである。CPU: Inter Core i7-4770k 3.50GHz, RAM: 16.0GB, GPU: NVIDIA GeForce

GTX 760.RGB-Dカメラ: Microsoft Kinect v1.0(640 × 480), プロジェクタ: EPSON EB-W8(1280 × 1024)



図6 実験環境

3.1 立方体追跡の精度評価実験

提案手法とICPアルゴリズムをARマーカーによる追跡[8]との立方体の3次元中心位置誤差、姿勢の角度誤差を176フレーム間で算出した。その結果を図7, 8, 表1に示す。

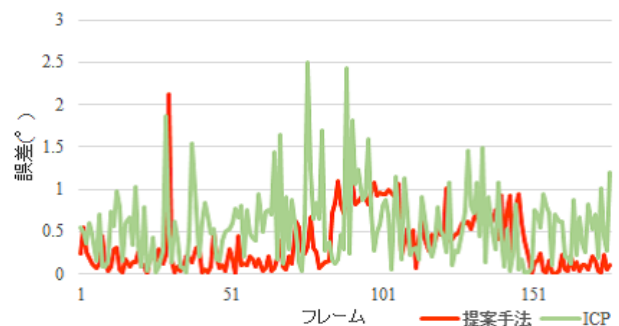


図7 立方体の姿勢の角度誤差

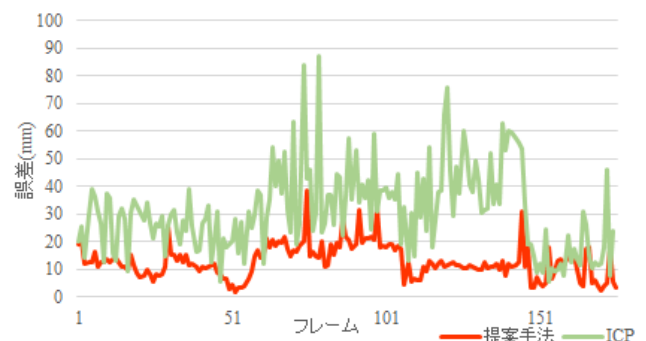


図8 立方体の中心位置誤差

表1によると立方体の姿勢推定ではマーカーによる追跡手法との角度の誤差は両手法でほとんど見られなかった。提案手法においては平面領域の分割の際に法線ベクトルの似た点群をクラスタリングし平面の法線ベクトルを算出し

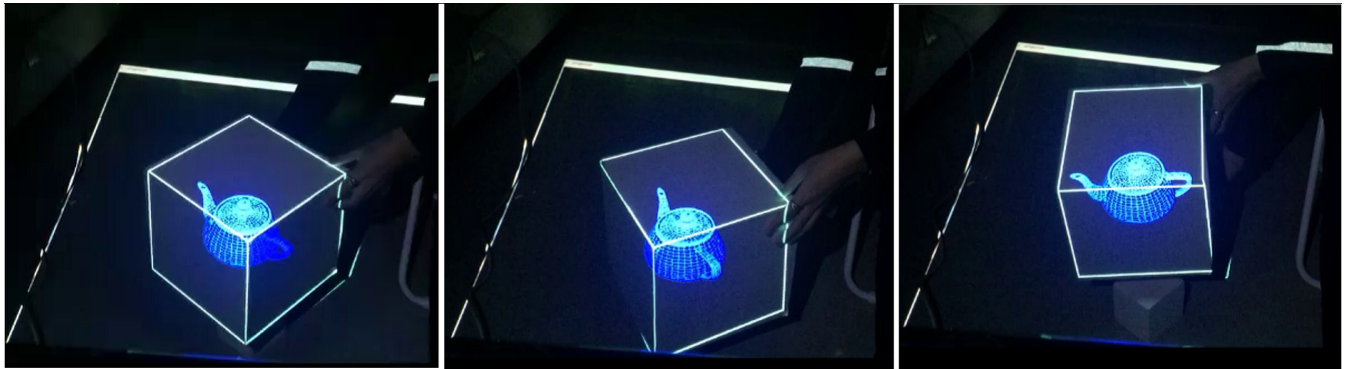


図 9 投影結果画像

ているため法線ベクトル方向のずれがあまり見られなかったと考えられる。ICP アルゴリズムにおいても同平面状の点に対応点になれば姿勢の誤差への影響を与えないためであると考えられる。一方、立方体の3次元中心位置推定では ICP アルゴリズムに比べて誤差平均値を大幅に軽減することができた。なおかつ標準偏差の点でも軽減できたため精度の安定した推定結果を得ることが出来た。フレーム間でのばらつきが軽減したことは図8のグラフからも確認できる。このように ICP アルゴリズムによる推定よりも精度を向上させマーカによる追跡に近い精度での推定ができた。また提案手法の処理速度は 12.4 フレーム毎秒で、立方体を手で動かしても十分に追跡できる速度であった。

表 1 マーカーによる追跡との誤差平均

	提案手法	ICP
位置 (mm)	12.8 ± 6.1	31.2 ± 15.2
角度 (°)	0.3 ± 0.3	0.6 ± 0.4

3.2 立方体への動的な空間型 AR の投影実験

提案手法で算出した位置・姿勢を用いて実際にプロジェクタで立方体に対して動的に投影を行った結果を図9に示す。なお投影した仮想空間は立方体の中心にティーポットを配置し、立方体の辺の部分に線が見えるような投影を試みた。

図9によりユーザーとなるカメラ視点からは立方体の中に仮想空間が存在するかのように投影されていることがわかる。デプス情報のみを用いて追跡を行っているために、暗室での投影光の影響を受けることなく立方体の位置・姿勢を推定し動的な空間型 AR を実現できた。辺の部分の線が横に漏れている部分が見られる理由のひとつとしてまず位置・姿勢推定における誤差の影響があげられる。また立方体を動かしている際に、プロジェクタ投影における遅延により推定された位置と実際の立方体の位置が異なるためにズレるといふ理由も考えられる。

4. まとめ

本研究では RGB-D カメラ映像から得られた 3 次元点群を Superpixel Segmentation とラベリングにより平面領域に分割し、投影したい立方体の平面領域を抽出する。そして領域内の 3 次元点群に対して立方体のモデルフィッティングを行うことによりリアルタイムで立方体の位置・姿勢追跡を行う手法を提案した。これを用いて立方体面に対しプロジェクタで映像を投影することで立方体への動的な空間型 AR を可能にした。

今後の課題として、実際に手に取って立方体を動かす際 1 面しか抽出されない場合も十分にあるためそのときの位置・姿勢推定を行いたい。またユーザーが動いてもいいようにユーザーの頭部追跡を行いたい。

参考文献

- [1] D. Kobayashi, N. Hashimoto: "Spatial Augmented Reality by using Depth-Based Object Tracking", In *Proc ACM SIGGRAPH Posters* (2014)
- [2] Yichen Tang, Billy Lam, Ian Stavness, Sidney Fels: "Kinect-based Augmented Reality Projection with Perspective Correction", In *Proc ACM SIGGRAPH Posters* (2011)
- [3] Ian Stavness, Billy Lam, Sidney Fels: "pCubee: A Perspective-Corrected Handheld Cubic Display", 2010 Computer Human Interaction (CHI), pp. 1381-1390(2010)
- [4] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, K. Toyama: "Digital Photography with Flash and No-flash Image Pairs", *ACM Transaction on Graphics(TOG)*, Vol. 23, No. 3, pp.664-672(2004)
- [5] S. Holzer, R.B. Rusu, M. Dixon, S. Gedikli, N. Navab: "Adaptive Neighborhood Selection for Real-time Surface Normal Estimation from Organized Point Cloud Data Using Integral Images", 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.2684-2689 (2012)
- [6] D. Weikersdorfer, D. Gossow, M. Beetz: "Depth-adaptive Superpixels", 2012 21st International Conference on Pattern Recognition, pp.2087-2090(2012)
- [7] K.A Hawick, A. Keist, D.P. Playne: "Parallel Graph Component Labeling with GPUs and CUDA", *Parallel Computing*, Vol. 36, No. 12, pp.655-678(2012)
- [8] <http://www.hitl.washington.edu/artoolkit/>