H-014

深度情報の重ね合わせに基づく距離画像センサの移動軌跡推定

Trajectory Estimation of a Range Image Sensor Based on Depth Information Mosaicing

菅原 勝也 † Katsuya Sugawara　阿部 亨 ‡ Toru Abe　菅沼 拓夫 ‡ Takuo Suganuma

1. はじめに

現在，自走ロボットの自律制御や，AR 技術における実画像への仮想物体の重ね合わせなど，多くの分野で，移動する物体の移動軌跡（位置・姿勢）を推定する技術が必要とされている．しかし，屋内での移動物体を対象とした場合，従来の手法では，推定精度が十分ではない，環境内に専用の装置を事前に設置する必要がある等，各手法には様々な問題が存在する．

そこで本稿では，移動する物体に距離画像センサを搭載し，センサから得られる深度情報を用いることで，外界に専用装置を設置することなく，高精度に移動物体（センサ）自身の移動軌跡を推定する手法を提案する．また，距離画像センサの一つである Kinect と，そのライブラリ Kinect for Windows SDK の機能を利用した試作システムにより，本手法の有効性を検証する．

2. 関連研究

屋内を対象とした物体の移動軌跡の推定については，超音波や赤外線，電波などを利用した様々な技術 [1] が提案・開発されている．しかし，これらは，環境内に専用装置を事前に設置する必要があり，設置コストが高い，未知の環境に対応できない等の理由から，適用が困難な場合も多い．一方，移動する物体自体に加速度センサを搭載し，外界に専用装置を設置することなく，搭載センサの情報のみで軌跡推定を行う手法 [2] や，加速度センサの他にカメラを搭載し，連続して得られる外界の映像を順次重ね合わせることで決定されるパラメータを用い，推定精度向上を図る手法 [3] も提案されているが，十分な精度を得ることは未だに困難である．

3. 提案手法

提案手法では，移動する物体に距離画像センサを搭載し，センサから連続して獲得される深度情報を，既に獲得された深度情報と逐次重ね合わせ，その際に得られた深度情報の移動距離・回転角度を用いることで，基準位置に対する移動物体（センサ）自身の相対的な移動軌跡（位置・姿勢）の推定を行う．

これを実現するために，本稿では，Kinect のライブラリ Kinect for Windows SDK [4] に含まれる KinectFusion [5] の機能を利用する．KinectFusion は，図 1 に示すように，a) センサで新たに獲得された深度情報を 3 次元の点の集合に変換し，b) ICP アルゴリズム [6] により，既に獲得されている点の集合と重ね合わせ，c) その結果をボクセル空間内で統合することで，高品質・高精度な 3 次元形状の構築を行う機能である．また，

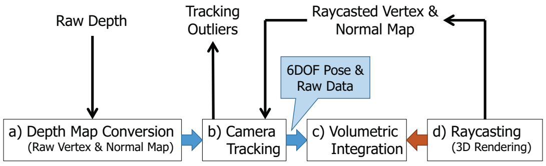


図 1: KinectFusion の処理の流れ

d) 3 次元形状に対しレイキャスティングを行った結果を表示する機能も備えている．深度情報（実際は，それを変換した 3 次元の点の集合）を重ね合わせるために b) で決定された変換パラメータは，ボクセル空間の座標系に対するセンサの位置・姿勢に等しい．従って，移動するセンサから連続して得られる深度情報に対し KinectFusion を順次適用すれば，決定された変換パラメータからセンサ自体の移動軌跡の推定が可能となる．

4. 評価実験

4.1. 実験内容

提案手法の有効性を検証するため，試作システムにより，距離画像センサの位置・姿勢を推定する実験を行った．距離画像センサとして Kinect for Windows を用い，試作システムは，Visual C++ 2010 と Kinect ライブラリ（Kinect for Windows SDK 1.7）の KinectFusion により，ノート PC（CPU Intel Core i5-3317U 1.7GHz, GPU Intel HD Graphics 4000, memory 4GB, OS Windows 8）上に実装した．

実験環境を図 2 に示す．壁面の本棚から距離 d の箇所にセンサを設置し，x, y 軸は本棚前面に平行で，z 軸は垂直となるよう座標系を設定した．実験では，センサを，基準位置から x, y, z 軸方向に各々移動した場合の距離，および，x, y 軸に関し各々回転した場合の角度について逐次推定を行った．なお，Kinect は，Default モードで使用し，深度情報を 640×480 画素，30fps で出力するよう設定した．また，KinectFusion が深度情報を統合するためのボクセル空間のサイズは，$512 \times 384 \times 512$ ボクセル（256 ボクセル/m）とした．

4.2. 実験結果

処理時間

まず，移動距離・回転角度を推定する際の処理速度を求めた．CPU のみで処理を行った場合の平均処理速度は 2.47 回/s であった．これに対し，KinectFusion の設定により，GPU の機能（DirectX 11）を用い処理を行った場合の平均処理速度は 15.2 回/s であった．

移動距離の推定

次に，センサを，基準位置から x 軸方向に $0 \rightarrow 1.0 \rightarrow 0$m（0.05m 単位），$y$ 軸方向に $0 \rightarrow 0.24 \rightarrow 0$m（0.02m

†東北大学大学院情報科学研究科，Graduate School of Information Sciences, Tohoku University
‡東北大学サイバーサイエンスセンター，Cyberscience Center, Tohoku University

Copyright © 2013 by Information Processing Society of Japan and The Institute of Electronics, Information and Communication Engineers
All rights reserved.

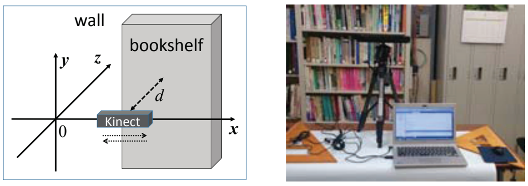


図 2: 実験環境

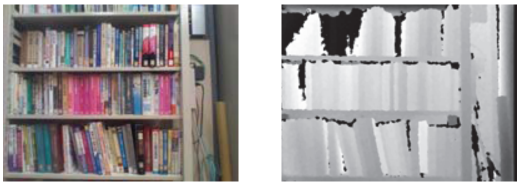


図 3: センサで得られる画像（左），深度情報（右）の例

単位），z 軸方向に $0 \to 1.0 \to 0$m（0.05m 単位）各々移動し，その移動距離を逐次推定する実験を行った．なお，x, y 軸方向の移動を行う場合は，本棚とセンサとの距離を $d =$ 約 1m，および，約 1.8m に設定した．

センサから得られる画像と深度情報の例を図 3 に，移動距離推定の結果を図 4 に示す．この実験結果から，基準位置からの移動距離が大きいほど推定誤差（実際の移動距離-推定移動距離）が大きくなる傾向が確認できる．また，x, y 軸方向に移動した場合，距離 d を変えても，推定誤差の大きな変化は確認できなかった．

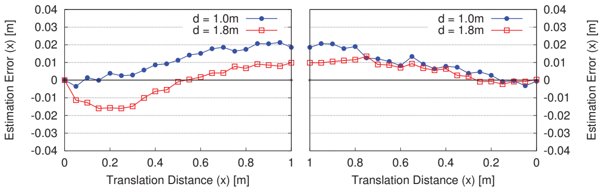
回転角度の推定

センサを，基準位置から x, y 軸に関し $0 \to -40°$，$0 \to 40°$（10° 単位）各々回転し，その回転角度を逐次推定する実験を行った．その結果を図 5 に示す．なお，x 軸に関する回転では，$d = 1.8$m に設定した場合，負方向の回転角度が大きくなると深度情報の重ね合わせが実行できなかったため，図 5 では，$0 \to -20°$ での推定結果のみ示している．

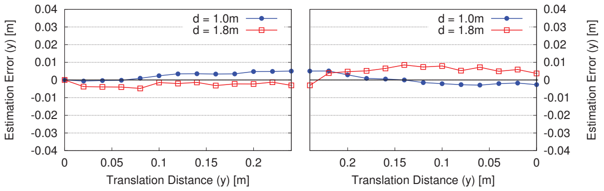
移動距離推定の場合と同様，基準位置からの回転角度が大きくなるほど推定誤差が大きくなる傾向が確認でき，本棚とセンサとの距離 d を変えても，推定誤差の大きな変化は確認できなかった．

5. おわりに

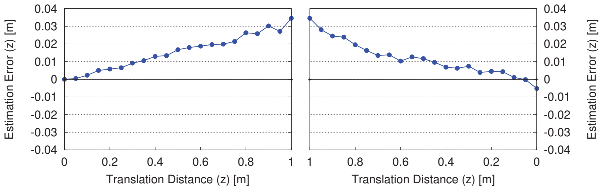
本稿では，距離画像センサで得られる深度情報を重ね合わせることでセンサ（センサを搭載した物体）自体の移動軌跡を推定する手法を述べた．Kinect と KinectFusion を利用した評価実験の結果から，GPU の機能を利用することで，センサの深度情報出力フレームレート（30fps）の約 1/2 の処理速度でセンサの移動距離・回転角度を逐次推定できることが確認できた．また，移動距離 1m に対する推定誤差は最大 0.03m，回転角度 40° に対する推定誤差は最大 4.1° であり，センサの移動軌跡を高い精度で推定できた．今後，様々な環境・設定で，提案手法のより詳細な実験・検討を行い，自走ロボットや AR 技術，複数の距離画像センサから得られる深度情報の統合等，具体的応用への適用を図る予定である．



x 軸方向の移動:（左）$0 \to 1.0$m,（右）$1.0 \to 0$m

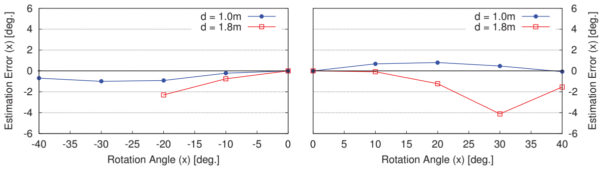


y 軸方向の移動:（左）$0 \to 0.24$m,（右）$0.24 \to 0$m

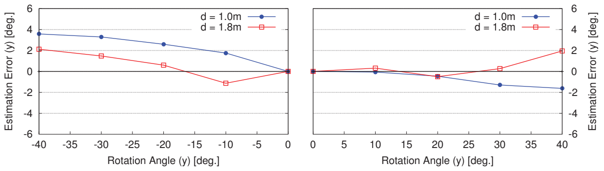


z 軸方向の移動:（左）$0 \to 1.0$m,（右）$1.0 \to 0$m

図 4: x, y, z 軸方向の移動距離推定結果



x 軸に関する回転:（左）$0 \to -40°$,（右）$0 \to 40°$



y 軸に関する回転:（左）$0 \to -40°$,（右）$0 \to 40°$

図 5: x, y 軸に関する回転角度推定結果

参考文献

[1] 別所 他, “ユビキタスコンピューティングと屋内環境の位置認識,” 信学会誌, Vol.92, No.4, pp.249–255 (2009).

[2] 興梠 他, “装着型自蔵センサモジュールを用いた歩行者の位置・方位計測技術,” 信学会誌, Vol.92, No.4, pp.268–275 (2009).

[3] S.-H. Jung and C.J. Taylor, “Camera trajectory estimation using inertial sensor measurements and structure from motion results,” Proc. Conf. Comput. Vision Pattern Recognit., Vol.II, pp.732–737 (2001).

[4] Kinect for Windows, `http://www.microsoft.com/en-us/kinectforwindows/`, Online; accessed 1-June-2013.

[5] 3D Surface Reconstruction – Microsoft Research, `http://research.microsoft.com/en-us/projects/surfacerecon/`, Online; accessed 1-June-2013.

[6] J. Salvi, et al., “A review of recent range image registration methods with accuracy evaluation,” Image Vision Comput., Vol.25, No.5, pp.578-596 (2007).

Copyright © 2013 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.