

複数の深度カメラを用いたリアルタイム 3D スキャニング

松本 依里紗[†] 藤田 悟[‡]

法政大学大学院 情報科学研究科[†] 法政大学 情報科学部[‡]

1. はじめに

近年、ヘッドマウントディスプレイの普及に伴い、VR への関心が高まりつつある。仮想空間に現実世界の物体を投影するには、物体のスキャニングが必要であるが、Kinect Fusion[1]などの既存の手法ではモデル生成に時間が掛かってしまう。人間の表情や動作を瞬時に投影するにはリアルタイムにスキャニングする必要がある。本研究では、3次元の遠隔対話システムの構築を目的とし、多視点から得られた深度情報を統合し、1つの3次元モデルをリアルタイムに生成する手法を提案する。

2. 提案システム

撮影物体を中心に半径 2m の円周上に Kinect を等間隔に配置する。事前処理として Kinect 間の位置合わせを行い、変換行列を各 Kinect に与える。Kinect から得られた RGBD データをソケット通信により描画を行う PC に送信した後、全ての RGBD データから 1つのメッシュデータを生成、描画をリアルタイムに行う。本論では、複数の深度情報を統合し、メッシュデータの作成を行う手法の提案と処理速度の検証を行う。

3. 提案手法

先行研究[2]では、Kinect から得た深度データをもとに 1枚の 3次元モデルを生成、6方向から生成したモデルを重ね合わせることでリアルタイムに全周 3次元モデルの生成を行った。しかし、Kinect 間の位置合わせのずれからいびつなモデルや多層のモデルが生成されてしまった。そこで本稿では、複数視点の深度データを 1つのボクセルデータに蓄積しその統合結果から表面形状を生成する手法に変更した。具体的には、複数の深度情報を統合する手法として、3次元空間をボクセルで分割し、各ボクセルの頂点に符号を与える

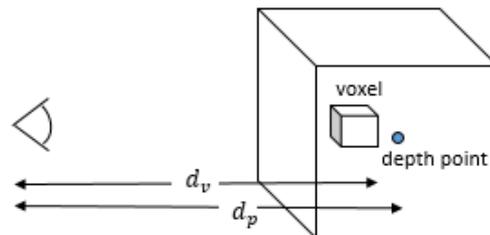


図 1 カメラ視点からの距離

符号付距離場 (signed distance field, SDF) と、立方体の 8つの頂点の符号からポリゴンを求めるマーチングキューブ法 (marching cubes algorithm) [3]を用いる。リアルタイム処理を行う為に、符号付距離場とマーチングキューブの計算には CUDA による並列処理を行う。Kinect のキャリブレーションについては、事前処理として ICP (Iterative Closest Point) を用いて変換行列を求めておく。

3.1. 符号付距離場

Kinect の視点座標から深度座標上にあるボクセル頂点に対して、Kinect の視点座標から深度座標までの距離とボクセル頂点までの距離の差を求める。この距離の差を重みとしてボクセル頂点に与える。全ての Kinect から得た深度座標に対して距離の差を求め、重みを更新していく。重みの符号が正であれば、ボクセルは深度データより視点座標側にある。ボクセルの 8つの頂点の符号が同一でなければ、深度データがある境界となり、そのボクセルに対してマーチングキューブ法によりメッシュを生成していく。

3.2. マーチングキューブ法

前節で求めた符号からマーチングキューブ法で各ボクセルに描画するポリゴン数を求める。ポリゴン数の prefixSum [4]を求めることで、配列のインデックス番号となるので並列処理内で頂点情報の格納が可能となる。また、prefixSum の末尾が頂点情報を格納する配列のメモリ数となる。ポリゴン数が 0 のボクセルについてはポリゴンの座標計算や頂点情報の格納処理が不要となるので、以降の処理

Real time 3D scanning with multiple depth cameras

[†]Erisa Matsumoto, Graduate School of Computer and Information Sciences, Hosei University

[‡]Satoru Fujita, Faculty of Computer and Information Sciences, Hosei University

表 1 表面積の違いによる処理速度の比較

	SDF [ms]	Marching cubes [ms]	ポリゴン数
物体 1	13.44	185.92	298748
物体 2	19.06	195.50	407034
物体 3	21.14	206.35	506392

表 2 Kinect の台数によるフレームレート比較

Kinect 台数	フレームレート [fps]
1 台	4.0
2 台	3.3
3 台	2.9
4 台	2.6

からは除外する．ポリゴン数が 1 以上のボクセルについて再度並列処理によりマーチングキューブ法を行い，ポリゴンの頂点座標を計算，格納していく．

4. 実験

4.1. 実験内容

表面積が異なる物体での符号付距離場とマーチングキューブの処理速度の比較と，Kinect を複数台用いた時の描画速度の比較を行う．GPU にはコア数 640 の GeForce GTX 850M を用いる．生成するボクセルは 1 辺 0.01mm，256 分割とする．

4.2. 結果

表 1 は生成したメッシュデータに含まれるポリゴン数の違いによる符号付距離場とマーチングキューブの速度の比較結果を示している．物体の表面積による処理速度の大きな差は見られなかった．

Kinect を 1 台のみ用いた時のフレームレートは 4.0fps となったが，4 台用いた時には 2.6fps となった．結果を表 2 に示す．

4.3. 考察

1 フレーム当たりの処理時間 250ms に対してマーチングキューブの処理時間が約 200ms と大半を占めているが，これは $256 \times 256 \times 256$ のすべてのボクセルに対して処理を行っていることが原因と考えられる．このため，ポリゴン数が異なる物体と比較しても大きな差は見られなかった．また，符号付距離場，マーチングキューブともに処理速度はポリゴン数に大きく依存しないことから，Kinect の台数が増えることにより描画面積が増加しても処理速度への影響は少ない．よって，Kinect の台数の増加によるフレームレートの低下の原因は通信処理にあると考えられる．

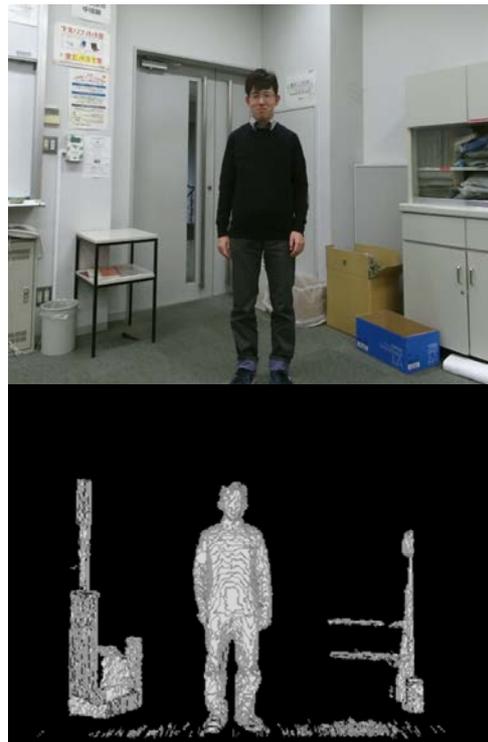


図 2 生成したメッシュデータ

5. おわりに

本稿では，複数の Kinect から得た深度情報を用いて，リアルタイムにメッシュデータの生成を行う手法を提案した．

今後の課題として送信データの圧縮や不要なボクセルの除去による処理速度向上を目指すほか，RGB データを用いてより実物体に近いモデルを生成することを試みる．また，本研究の目標である 3 次元の遠隔対話システムの構築も進める．

参考文献

- [1] Izadi, Shahram, et al. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera." Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 2011.
- [2] 松本 依里紗・藤田 悟, 3次元動体のリアルタイムスキャニング, 電子情報通信学会 2016 年総合大会, 2016.
- [3] Lorensen, William E., and Harvey E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm." ACM siggraph computer graphics. Vol. 21. No. 4. ACM, 1987.
- [4] Harris, Mark, Shubhabrata Sengupta, and John D. Owens. "Parallel prefix sum (scan) with CUDA." GPU gems 3.39: 851-876, 2007.