

データの効率的生成・表示のための低・高解像度 3D スキャンシステムの提案

坂田真太郎^{1,a)} 水野慎士^{1,b)} 山崎賢人² 岡原浩平² 皆川純²

概要: 本研究では、VR 空間などで点群データを活用することを目的として、点群データを効率的に生成して表示する手法を提案する。この手法では、固定カメラで空間全体を低解像度で三次元的にスキャンしながら、移動カメラを用いて注目領域を高解像度で三次元的にスキャンする。そして、それぞれのスキャンデータから生成された点群データを統合する。これにより、ユーザの注目する領域は高い解像度を持ちながら、全体としてはサイズの小さい点群データを生成することが可能となり、VR 空間で点群データを観察することも可能となる。

キーワード: 点群データ, 三次元スキャン, VR, データ統合, LOD

Proposal of a 3D Scanning System with Low and High Resolutions for Efficient Data Generation and Displaying

SHINTARO SAKATA^{1,a)} SHINJI MIZUNO^{1,b)} KENTO YAMAZAKI² KOHEI OKAHARA² JUN MINAGAWA²

Abstract: In this research, we propose a method to efficiently generate and display point cloud data with the aim of utilizing it in VR space. In this method, mobile cameras scan the region of interest three-dimensionally with high resolution, while fixed cameras scan the entire space with low resolution. Point group data is generated by integrating the respective scan data of each camera. As a result, the size of the generated point cloud data decreases while the user's region of interest has high resolution, and it is possible to observe the point group data in the VR space.

Keywords: Point cloud data, 3Dscan, VR, Data integration, LOD

1. はじめに

レーザスキャナやステレオカメラを用いて実空間の物体を三次元的にスキャンして得られた点群データは、文化財の三次元 CG 制作 [1] や市街地の三次元モデリング [2] など幅広い分野で用いられている。特に近年、Kinect や Intel RealSense などリアルタイムで空間を三次元的にスキャン

できる深度カメラの普及に伴って、インタラクティブシステムでも三次元スキャンによる点群データが用いられるようになり、エンタテインメントコンテンツ [3] や視覚障がい者の支援 [4] など、様々な分野で活用されている。

本研究では、深度カメラを用いて実空間を三次元的にスキャンして得られた点群データを用いて、VR 空間で実空間を再現して活用するコンテンツに着目する。VR コンテンツではユーザが VR 空間内を歩き回ることが考えられるため、実空間を再現するための点群データは広範囲で十分な解像度を有する必要がある。一般的な深度カメラは画素数が VGA 程度で、水平視野角は約 60~90 度となっている。そのため、VR 空間で使用する点群データを取得するには多くの深度カメラが必要になる。そして、多くの深度

¹ 愛知工業大学大学院経営情報科学研究科
Graduate School of Business Administration and Computer Science, Aichi Institute of Technology

² 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

a) b18715bb@aitech.ac.jp

b) s_mizuno@aitech.ac.jp

カメラで取得した点群データはサイズが膨大になり、インタラクティブコンテンツに必要なリアルタイム性が困難になるという問題も生じる。

そこで、本研究では効率的な点群データの生成と表示を行うことで問題の解決を図る。そして、少ない台数の深度カメラを用いながら、VRにも対応した点群データを取得して利用する手法を提案する。実空間を再現するために最低限必要となる点群データは、固定位置に設置した深度カメラで取得する。それに加えて、ユーザが注目して詳細情報が必要な領域の点群データは、ユーザの移動や操作に応じて移動する深度カメラを用いて近接で取得する。そして、各深度カメラで取得した点群データをリアルタイムで統合する。これにより、ユーザが注目する領域だけ高い解像度を持ち、それ以外の領域は低い解像度で構成された点群データが生成されるため、VRなどでも高速に表示して観察することが可能となると考えられる。

2. 関連研究

点群データを高速かつ高品質に観察する方法は、一般的に2通りの方法が考えられる。

一つは、事前に高解像度の点群データを取得しておき、表示や使用の際に効率化を図る手法である。前野ら [5] は、取得した点群データを階層化しておき、視点と点群との距離に応じて表示する点群階層を変更したり表示モデルを点・線・メッシュなどに切り替えることでリアルタイム表示を実現する LOD(Level of Detail) を適用している。宮地 [6] は、あらかじめ設定した複数の視点から観察した描画結果やデプスバッファの値を利用して表示に不要な点を削減することで、描画結果を低下させずに点群データを軽量化する手法を提案している。

もう一つは、事前に取得した点群データは低解像度であるが、表示や使用の際に必要なに応じてデータを補間する手法である。金森ら [7] は、点群データからサーフェルを用いて表示用の面を生成する手順において、曲面を表示する際に穴が生じないようにするために点群をアップサンプリングする手法を提案している。中川ら [8] は、視点からのレンダリング結果に対して奥行き情報を考慮した画像補間手法を開発して、低解像度の点群データから VR 用の高品質の画像をリアルタイムで生成する手法を提案している。

本研究で提案する手法は、どちらかと言えば後者の方法に分類される。ただし、多くの手法が高解像度の点群データを補間などによって合成しているのに対して、本研究の提案手法は移動可能な深度カメラを追加して、ユーザが注目する領域だけに対して高解像度の点群データを実際に取得することで、点群データの表示の高速化と高品質化を両立させている。

3. 提案手法について

3.1 概要

本研究では、点群データを VR などで観察できるように高品質かつ高速に生成・表示することを目的とする。そのため、LOD の考え方を適用して、観察対象全体は低解像度の点群データで構成しながら、ユーザが注目する領域については高解像度の点群で構成する。その際、高解像度データは補間で合成するのではなく移動する深度カメラで実空間から実際に取得して、低解像度データに統合させる。これにより、VR に対応した高速かつ高品質な点群データのレンダリングを目指す。

3.2 システム構成

本研究で提案するシステムは、観察対象全体を低解像度で三次元的にスキャンするための固定カメラサブシステム、ユーザが注目する領域を高解像度で三次元的にスキャンするための移動カメラサブシステム、各カメラシステムの深度画像やカラー画像を受け取って処理および表示を行うデータ処理サブシステムで構成される。固定カメラサブシステムと移動カメラサブシステムはネットワークでデータ処理サブシステムと接続している。図 1 にシステムの構成を示す。

固定カメラサブシステムは、PC、深度カメラ、RGB カメラで構成される。深度カメラと RGB カメラは観察対象に対して固定されており、実空間中に K 台 (1 台または複数台) 設置する。そして、観察対象を深度カメラと RGB カメラで撮影して、得られた深度画像、および深度画像の各画素に対応する画素値を持つカラー画像をデータ処理サブシステムに逐次送信する。

移動カメラサブシステムは、PC、深度カメラ、RGB カメラで構成されており、固定カメラサブシステムと同じ構成である。観察対象を深度カメラと RGB カメラで撮影して得られた深度画像とカラー画像をデータ処理サブシステムに逐次送信する点も同じであり、実空間中に L 台 (1 台または複数台) 設置する。ただし、深度カメラと RGB カメラは移動可能であるため、カメラの位置と姿勢を取得するためのマーカを取り付けている。そして、移動カメラはユーザ自身または遠隔操作によってユーザの興味がある観察対象に自由に接近することが可能である。その結果、固定カメラと同じ画素数の深度カメラを用いても、観察対象を近くから撮影することで単位面積当たりの画素数は大きくなり、高解像度の深度画像を得ることができる。

データ処理サブシステムは、PC、マーカ追跡カメラ、モニタで構成されている。マーカ追跡カメラは移動カメラサブシステムのカメラなどに取り付けられたマーカの位置と姿勢を取得するために用いる。そして、移動カメラサブシ

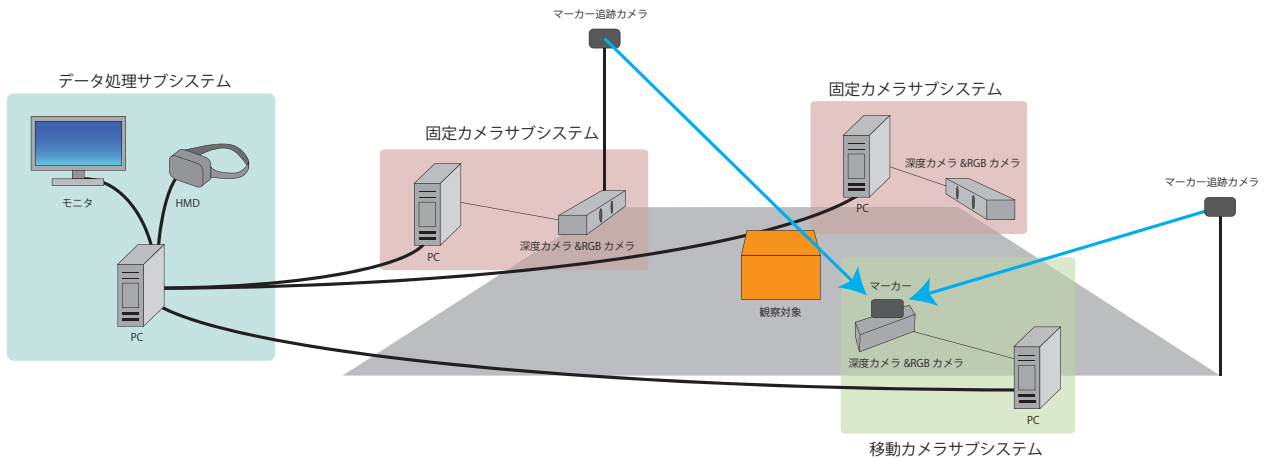


図 1 提案システムの構成

システムの位置と姿勢を考慮しながら各カメラシステムから送信されたデータを統合して点群データを生成して、点群データを 3DCG として表示する。モニタとして HMD を用いた場合には、点群データを VR 空間で観察することも可能となる。

3.3 各カメラサブシステムのデータの統合

データ処理サブシステムは、各カメラサブシステムを構成する深度カメラのカメラパラメータを用いて、深度画像の各画素を三次元座標中に変換する。そして、各画素に対応するカラー画像の画素値を用いて、カラー化された点群データを生成する。最後に、各カメラサブシステムに基づく点群データを統合することで、表示用の点群データが得られる。

ここで、各カメラサブシステムの点群データはそれぞれ個別の座標系での三次元座標となっているため、このまま統合することはできない。そこで、実空間に対応したワールド座標系を用意しておき、各カメラサブシステムで得られた点群の座標をワールド座標系での座標に変換する。

実空間に対応したワールド座標系は、データ処理サブシステムのマーカ追跡カメラに基づいて定義している。そして、事前に固定カメラサブシステム k ($k = 1 \sim K$) の深度カメラにマーカを取り付けておくことで、ワールド座標系でのカメラの位置と姿勢を表す 4×4 の座標変換行列 M_{A_k} を取得する。データ処理サブシステムは各固定カメラサブシステムで得られた点群データ p_{A_k} に各カメラサブシステムの座標変換行列 M_{A_k} を乗じて、ワールド座標系での点群データ p_W に変換する。

$$p_W = M_{A_k} p_{A_k} \quad (1)$$

その結果、すべての固定カメラサブシステムの点群データをワールド座標系で統合することができる。なお、座標変換行列を取得したあとは固定カメラサブシステムのマーカは不要である。

移動カメラサブシステム l ($l = 1 \sim L$) で得られた点群データ p_{B_l} も式 (1) と同様の手法でワールド座標系での座標に変換する。ただし、ここで用いる深度カメラは移動するため、システム実行中も深度カメラにマーカを取り付けておく。そして、データ処理サブシステムはマーカ追跡カメラを用いて深度カメラの位置と姿勢を追跡し続けながら、 4×4 の座標変換行列 M_{B_l} を逐次更新する。それにより、カメラの移動に関わらず、移動カメラサブシステムの点群データをワールド座標系の座標に変換することができる。

以上により、移動カメラサブシステムを含むすべてのカメラサブシステムの点群データをワールド座標系で統合することができる。そして、ユーザは興味がある対象については移動カメラサブシステムで高解像度の点群データを取得しながら、統合された点群データを VR 等で自由に観察することが可能となる。

4. 実験・考察

4.1 実験環境

提案手法の有効性を実際にシステムを実装して実験を行った。システムは、固定カメラサブシステムが 1 台、移動カメラサブシステムが 1 台の構成となっており、それぞれ Ethernet(1Gbps) でデータ処理サブシステムに接続している。

固定カメラサブシステムは PC(Windows 10) 上に C++ で実装した。深度カメラと RGB カメラとして Kinect V2 を使用する。深度カメラで取得する深度画像のサイズは 512×424 画素であり、RGB カメラで取得するカラー画像のサイズはクリッピングと解像度変換によって深度画像と同じ 512×424 画素としている。そして、両画像を UDP 通信でデータ処理サブシステムに送信する。

移動カメラサブシステムもシステム構成、画像生成、画像送信は固定カメラサブシステムと同じである。加えて、マーカとして Kinect V2 に VIVE トラッカーを取り付けている (図 2)。



図 2 移動カメラに取り付けられたマーカ

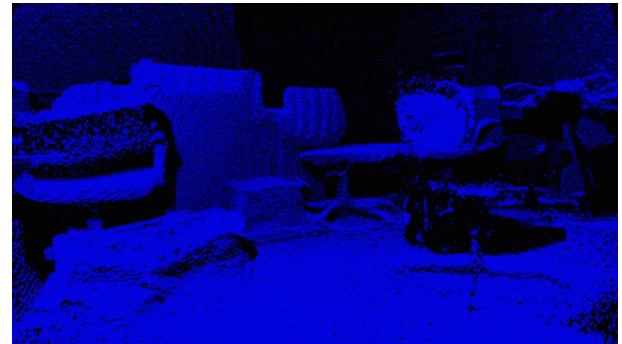
データ処理サブシステムは点群データを表示する必要があるため、ゲームエンジンである Unreal Engine 4 を用いて PC(Windows 10) 上に実装した。マーカ追跡カメラとしては VIVE ベースステーションを用いている。はじめに、事前に固定カメラサブシステムに VIVE トラッカーを取り付けて、固定カメラサブシステムのカメラの位置と姿勢を表す座標変換行列を計算する。その後、移動カメラサブシステムのカメラに取り付けられた VIVE トラッカーを追跡することで、移動カメラサブシステムのカメラの位置と姿勢を表す座標変換行列を逐次計算する。表示において点群に基づく面生成等は行わず、点をそのまま空間に配置して表示する。

ユーザはマーカ付きの VIVE HMD を装着して、ワールド座標系に統合された点群データを観察する。このとき、ユーザは移動カメラサブシステムの Kinect V2 を手に持つ。従って、ユーザは固定カメラサブシステムで得られた点群データが配置された VR 空間を自由に移動しながら、興味のある領域は手持ちカメラで取得した高解像度点群データとして観察する。

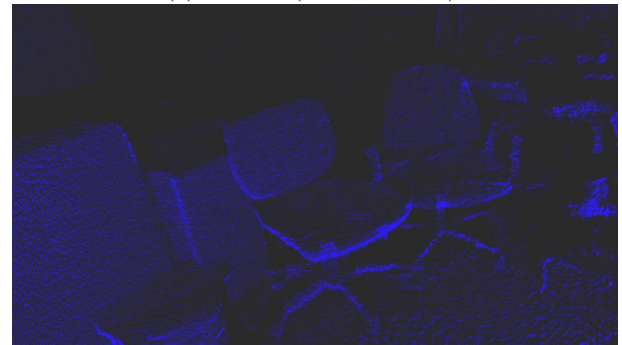
4.2 実験結果

最初に深度画像のみを用いた実験を行った。固定カメラは観察対象である椅子 3m の距離に設置しており、移動カメラを持ったユーザは椅子から 1m の距離から観察する。各カメラサブシステムは深度画像だけを送信して、データ処理サブシステムでは、固定カメラサブシステムの点群データは青色、移動カメラサブシステムのデータを赤色で表示している。

観察対象の点群データを表示の様子を図 3 に示す。固定カメラサブシステムだけの点群表示では、点群データの解像度が低いため観察対象である椅子の形状の詳細が確認できなかったり、カメラの死角の部分が欠落していたりした。それに対して、固定カメラサブシステムと移動カメラサブシステムを統合した点群データでは、移動カメラで取得した画像によって観察対象付近の点群データの解像度があがって、椅子の形状の詳細が確認できるようになった。また、固定カメラでは死角となっている領域も移動カメラ



(a) 空間全体 (固定カメラのみ)



(b) 固定カメラのみ



(c) 固定カメラ+移動カメラ

図 3 固定カメラと移動カメラに基づく点群データの統合

によって点群データが生成された。

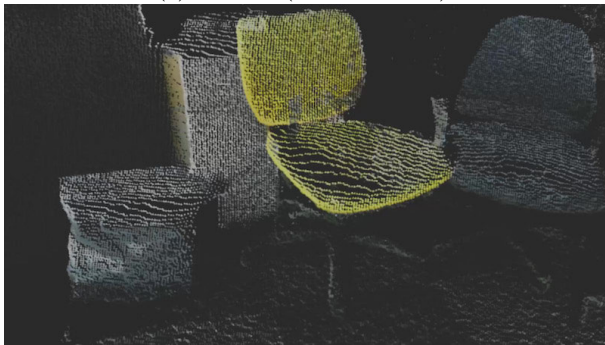
2 台のカメラサブシステムで取得して統合された点群データの総ポイント数は約 434,000 点となるが、点は 60fps で表示されており、HMD を用いた VR 空間での観察を問題なく行うことができた。ただし、深度画像を取得しながら逐次処理を行なっているため、送受信処理に起因する点データ更新の遅延が確認された。

なお、Kinect V2 の横方向の視野角は 70 度、縦方向は 60 度で、画素数は 512×424 画素である。従って、3m の距離から撮影した深度画像の画素間ピッチは 8.2mm となり、1m の距離から撮影した深度画像の画素間ピッチは 2.7mm となる。

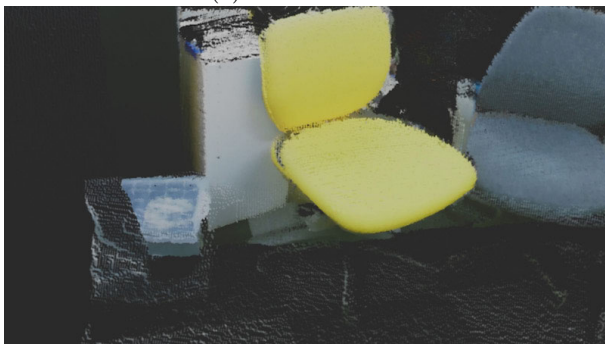
次に、深度画像とカラー画像を用いた実験を行った。カメラと観察対象の位置関係は深度画像のみの場合と同様であるが、各カメラサブシステムは深度画像とカラー画像を送信する。データ処理サブシステムでは、各カメラサブシステムの点群データをカラー画像の画素値をそのまま用いてカラー化して表示している。



(a) 空間全体 (固定カメラのみ)



(b) 固定カメラのみ



(c) 固定カメラ+移動カメラ

図 4 固定カメラと移動カメラに基づくカラー化点群データの統合

観察対象の点群データを表示の様子を図 4 に示す。深度画像のみの場合と同様に観察対象をである椅子をより高い解像度かつ少ない死角で観察できることが確認できた。さらに、観察対象の色や模様も確認できるため、より臨場感の高い観察が可能となった。

表示については、点をカラー化した場合でも 60fps は保たれており、HMD を用いた VR 空間での観察は問題なく行うことができた。ただし、2つのカメラサブシステムのカラー画像の色に若干の差が確認された。これは、カメラの個体差および撮影位置の違いによる映り方の違いによるものであると考えられる。また、深度画像に加えてカラー画像を送受信する必要があるため、深度画像のみ用いた場合より点データの更新に関する遅延が大きくなった。

4.3 考察

一般的に、深度カメラで解像度が n 倍の三次元スキャンデータを取得するには n^2 倍の深度カメラが必要となり、データ量も大幅に増加する。そこで、本研究では固定カメ

ラサブシステムで低解像度で広範囲の点群データを生成しつつ、移動カメラサブシステムで高解像度で狭い範囲の点群データを生成する手法を提案した。実験では、固定と移動のカメラサブシステムをそれぞれ 1 台ずつ用いて空間の三次元的スキャンを行い、2つの点群データの座標系を統一させてから統合して、VR 空間に配置して表示した。表示速度は十分に高速であり、ユーザは点群が表示された VR 空間を移動しながら、興味のある領域を高解像度データで観察することが可能であった。

ただし、今回はカメラサブシステムで取得した画像をデータ処理サブシステムに逐次送信しながら、リアルタイムで点群データを生成して表示したため、データ送受信に起因するデータ更新の遅延が発生して、表示に残像が残るなどの影響が見られた。今後は画像データの高速な転送方法を適用する必要がある。また、カメラ間の色の違いも見られたため、点群データの自然な統合には色の正規化などの処理も必要であると思われる。

移動カメラはユーザとともに移動して、観察対象はユーザ自身の周囲の実空間という設定であった。移動カメラをユーザが操作したりユーザ自身の動きに同期するロボットなどに装着することで、観察対象が遠隔であっても同様の手法が適用できると考えられる。

5. まとめ

本研究では、実空間を三次元的にスキャンして得られた点群データを VR 空間で観察するために、効率的な点群データの生成と表示を行う手法を提案した。空間を三次元的にスキャンする際に、固定位置から空間全体を低解像度でスキャンするカメラサブシステムと、移動しながら必要な領域を高解像度でスキャンするカメラサブシステムを用いて、各サブシステムの情報を統合した点群データを生成して表示する手法を提案した。これにより、ユーザの注目する領域は高い解像度を持ちながら、全体としてはサイズの小さい点群データを生成することが可能となった。そして、複数のカメラサブシステムに基づいて点群データを生成しながら VR 空間で観察できることを確認した。

今後の課題としては、カメラサブシステムからデータ処理サブシステムへのデータ送信の高速化や複数カメラ間のカラー正規化などを行い、表示の高品質化を行うことが挙げられる。また、固定カメラサブシステムを 2 台以上使用することで、より広範囲の空間に対して提案手法を適用した実験を行う予定である。

参考文献

- [1] 大石岳史, 増田 智仁, 倉爪亮, 池内克史: 創建期奈良大仏及び大仏殿のデジタル復元, 日本バーチャルリアリティ学会論文誌, Vol. 10, No. 3, pp. 429-436, 2005.
- [2] 石川貴一郎, 天野嘉春, 橋詰匠, 瀧口純一, 清水聡: モービ

- ルマッピングシステムによる都市空間モデリング, 計測自動制御学会論文誌, Vol. 8, No. 17, pp. 132-139, 2009.
- [3] 松末千佳, 水野慎士: “MO-LIVE” & “ぼっちセッション” - 身体動作とスキンシップでサウンドを生成するシステム, インタラクション 2016 論文集, 162C46, pp. 668-673, 2016.
- [4] 森大輝, 入江英嗣, 内原正一, 荒川明宏, 坂井修一: リアルタイム経路生成と振動通知による視覚障害者の歩行支援システム, WISS2018 論文集, 2018.
- [5] 前野峻, 伊達宏昭, 金井理, 適応的描画プリミティブ選択とLODによる大規模環境レーザ計測点群の効果的表示手法, 2013年度精密工学会春季大会論文集, M34, pp. 855-856, 2013.
- [6] 宮地英生: ソフトウェアレンダリングを用いた点群データの軽量化, 第8回横幹連合コンファレンス予稿集, E-4-2, 2017.
- [7] 金森由博, 高橋成雄, 西田友是: 点で表現された曲面の穴埋めのための効率的なアップサンプリング法, 画像電子学会論文誌, Vol. 36, No. 4, pp. 1108-1116, 2007.
- [8] 中川雅史, 倉持里史, 黒木雅人, 千葉則茂: LiDAR VRによる高速な点群データ補間法, 写真測量とリモートセンシング, Vol. 50, No. 4, pp.1243-251, 2011.