

# 仮想空間への実世界環境のリアルタイム反映とその評価

服部 広大<sup>†</sup> 長尾 確<sup>‡</sup>

名古屋大学 工学部電気電子・情報工学科<sup>†</sup> 名古屋大学 大学院情報学研究科<sup>‡</sup>

## 1. はじめに

近年、ルームスケール VR という、部屋全体をプレイエリアとして歩き回ることが可能な VR が一般化している。さらにそれを拡大したビルディングスケール VR という、建物全体をプレイエリアとする VR が提唱されている[1]。ビルディングスケール VR ではあらかじめ床や壁などの建物の形状を VR 空間に取り込むが、椅子や机などの位置が変わる可能性がある現実物体を取り込み、逐一反映するのは困難である。そこで、本研究ではリアルタイムで実世界環境を仮想空間へ反映し、VR 体験中でも現実物体の操作や回避を可能にするシステムを作成した。このシステムを使うことで、従来のビルディングスケール VR では困難であった椅子や机などの物体の取り込みが容易になり、回避することも可能になり行動の幅が広がり、安全性が向上する。

このシステムの応用例として、現実環境を反映した VR 空間の作成支援が挙げられる。本システムでは現実物体は点群で表示されるが、より現実感を高めるためには 3D モデルでの表示が必要である。本システムでは現実物体を画像認識時に割り当てられるラベルごとに分割して点群を出力するので、ラベルごとに 3D モデルを検索することが可能で、3D モデルを使用した VR 空間の作成が容易になる。また、ルームスケール VR への応用も可能である。現在、ルームスケール VR では現実物体への衝突を防ぐため、設定したプレイエリア内の家具等を事前にどける必要がある。しかし本システムでは、プレイエリア内の物はそのままよく、VR 体験中の新しい安全対策の仕組みも実現することができる。

## 2. 実世界環境のリアルタイム反映システム

本研究では、YOLO v3 を用いた画像認識で実世界の物体を認識し、RGB-D カメラにより認識物体それぞれの点群を取得し、VR 空間上にリアルタイムで表示するシステムを作成した。システムの機能としては、主に画像認識と点群の生成・更新などを行う。

本システムは Python プログラムと主に点群の表示を行う Unity のプログラムで構成されている。

システムを使用して椅子と机を Unity 上に表示したときの様子を図 1 に示す。このとき Unity は Python から受け取った点群データをリアルタイムに表示する。

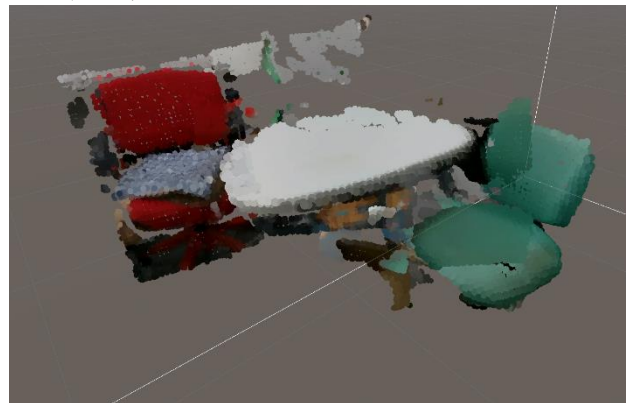


図 1 システムを使用した実世界物体の表示

ビルディングスケール VR で本システムを使用する際、壁や床などの背景に関しては事前に用意する必要がある。今回、背景の点群に関しては自律移動ロボットを用いて作成されたものを使用した[2]。

使用機材は VR 体験が可能な Head Mounted Display (以下 HMD)として HTC Vive Pro (以下 Vive)を使用する。なお Vive の前面に RGB-D カメラである Intel RealSense D435i (以下 RealSense) を装着する(図 2)。RealSense で取得したデプス、HMD の位置と向きに基づいて実世界物体の位置を計算して VR 空間に表示する。



図 2 RealSense を装着した Vive Pro

### 2.1. 点群の生成

本システムでは点群を生成、表示する対象を椅子と机に限定している。したがって、YOLO v3 がこの生成対象を認識すると、出力されたバ

Real-Time Reflection of Real-World Environment into Virtual Reality Space and its Evaluation

<sup>†</sup>HATTORI, Kodai(hattori@nagao.nuie.nagoya-u.ac.jp)

<sup>‡</sup>NAGAO, Katashi(nagao@nuie.nagoya-u.ac.jp)

<sup>†</sup>Department of Information Engineering, Nagoya University

<sup>‡</sup>Graduate School of Informatics, Nagoya University

ウンディングボックスをもとに認識物体ごとの領域を見つけ、点群を生成する。

点群の生成は、RealSenseで取得したカラー画像とデプス画像からPythonのライブラリであるOpen3Dを用いて生成した。生成した点群にはセンサーのノイズが含まれていたり、壁などの背景が含まれていたりする場合がある。そのため、バウンディングボックスの中心、つまり認識物体の中心からの距離が閾値を超える点は背景やノイズであると判断して除去し、さらに統計的外れ値除去を行うことで背景やセンサーのノイズを除去した。

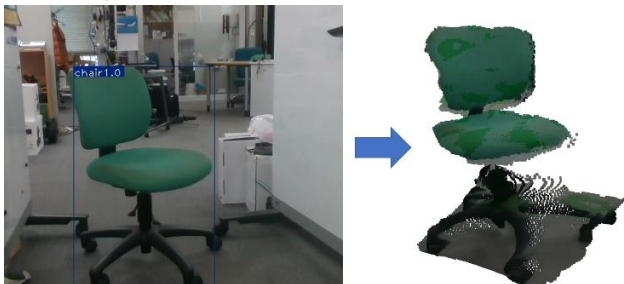


図1 認識結果からの点群生成

## 2.2. 点群の更新

時間が経過するにつれて点群の形や位置を更新していく必要がある。点群を更新する際には、認識した物体と、それ以前に認識済みの物体との対応付けを行わなければならない。また、前回と異なる方向から撮られた認識物体の点群を合成して1つの点群にするため最新の点群とそれまでに更新され続けた点群との位置合わせも行う必要がある。

対応付けに関しては、認識した物体と同ラベルかつ物体の中心位置が0.5m以内のものを同一物体と判定している。物体の位置情報はUnity内で絶対座標として保存されており、それをもとに対応付けが行われる。

対応付けが終わった後は、それまでに更新され続けた点群と最新の点群との位置合わせを行い、点群をマージし、1つの点群を生成する。位置合わせは、Fast Global Registrationを用いて行った[3]。これはRANSACベースのGlobal Registrationよりも高速なアプローチである。点群更新の過程で最も時間がかかる位置合わせを高速化することで、点群の更新頻度を上げることができ、リアルタイム性が向上した。

以上の手法を用いることで時間が経過するごとに点群が更新されていき、より正確な形状と位置を反映した点群を表示することが可能となる。また、センサーのノイズなどで適切に点群が生成されなかった対象に関しても、ある程度時間が経過すると自動的に修正される。

## 2.3. 点群データの受け渡し

点群の中心位置の座標や対応関係についてのデータはPythonとUnity間でTCP通信により共有している。ただし、点群データ(座標、色情報)に関してはサイズが大きいため標準出力でPythonからUnityに送信している。

## 3. 実験

今後、システムを評価するための実験を行う予定である。実験内容は、障害物として椅子を複数配置した環境を用意し、障害物を回避して歩行するときの行動データの取得と比較である。比較対象として、実世界を手動で忠実に再現したVR環境を用意する(図4)。これは事前に3DスキャナーであるStructure Sensorで形を取り込んだ実世界物体を、Viveトラッカーを使って正しい位置に表示したVR空間である。この2つを比較し、本研究で提案するシステムと事前に手作業で作成したVR環境が歩行において同等であるか、どのような差があるのかを検証する。



図2 実世界状況を手動で再現したVR環境

## 4. まとめ

本研究ではリアルタイムに実世界を仮想空間へ反映するシステム作成した。今後の課題としては、点群生成の精度向上や表示方法を工夫し、障害物回避時の行動が現実と同じように行えるようにすること、加えてVR体験中でも現実と同じように実世界の椅子や机が使えるようにすることが挙げられる。今後の展望としては、システムの応用例として挙げた、現実環境を改変したVR空間の作成支援や新しい安全対策の仕組みの開発も行っていく予定である。

## 参考文献

- [1] Nagao, K., Yang, M., Miyakawa, Y., Building-Scale Virtual Reality: Reconstruction and Modification of Building Interior Extends Real World, International Journal of Multimedia Data Engineering and Management (IJMDEM), 10(1), 2019.
- [2] 楊 夢龍, 長尾 確, 強化学習に基づく大規模屋内環境の3Dコンテンツの自動生成, 第81回情報処理学会全国大会講演論文集, 2019.
- [3] Q.-Y. Zhou, J. Park, and V. Koltun, Fast global registration, In Proc. of ECCV, 2016.