第 29 回日本バーチャルリアリティ学会大会論文集（2025 年 9 月）

機械学習による深度推定を用いたリアルタイム実空間再構成

Real-Time 3D Reconstruction of Physical Spaces Using Monocular Depth Estimation Based on Machine Learning

藤井健太郎 [1)]，加藤巧己 [2)]，斎藤大智 [3)]，木島竜吾 [4)]
Kentaro FUJII, Takumi KATO, and Daichi SAITO, Ryugo KIJIMA,

1) 岐阜大学 工学部 電気電子・情報工学科 （〒501-1193 岐阜県岐阜市柳戸 1－1 , fujii.kentaro.z5@s.gifu-u.ac.jp）
2) 岐阜大学 工学部 電気電子・情報工学科(〒501-1193 岐阜県岐阜市柳戸 1－1 , kato.takumi.e6@s.gifu-u.ac.jp)
3) 岐阜大学 工学部 電気電子・情報工学科(〒501-1193 岐阜県岐阜市柳戸 1－1 , saito.daichi.k8@s.gifu-u.ac.jp)
4) 岐阜大学 工学部 電気電子・情報工学科(〒501-1193 岐阜県岐阜市柳戸 1－1 , kijima.ryugo.n4@f.gifu-u.ac.jp)

概要：単眼深度推定を行う Depth Anything と Apple Depth Pro を比較した結果, 精度は Depth Pro が優れていた. 真値 1.6～2.2m に対する平均誤差は実測で約 0.45m であり, 公称のδ_1スコア約 50%と整合的であった, そこで 3 台の単眼カメラと Depth Pro の深度を用いて, 実時間三次元空間の再構成を試みたが, この誤差では空間の張り合わせ精度は実用的なレベルでは無かった.

キーワード：拡張・複合現実, 計測・認識, カメラ, 単眼深度推定, デジタルツイン, 実空間再構成

1. はじめに

1.1 目的

現実空間のいわゆるデジタルツインをもとめることは, VR/AR の基盤的な技術である. そのために用いられる LiDAR や, ステレオ/アクティブステレオ方式による深度カメラは, 画像深度を物理的に計測する物であり, 利用可能なレベルまで技術的には確立している. 他方, 近年では Apple Depth Pro[1], Depth Anything[2]など, 単眼画像から深度推定をする機械学習手法も盛んに研究されている. 空間再構成のためには多数のカメラが必要であり, Web カメラなどの安価なカメラを用いる事が出来ればメリットは大きいであろう.

そこで, 本研究では, 現実空間をカメラで撮影し, 単眼深度推定により仮想空間をリアルタイムに再構成し, 最終的に VR ヘッドセットを通じてその空間を自由に観察できるシステムを構築することを試みる.

1.2 単眼深度推定

1.2.1 単眼深度推定とは

単眼深度推定とは 1 枚の RGB 画像から機械学習を用いて被写体までの奥行を推定する技術である. Depth Anything は大規模なラベル付き・ラベル無画像データセットを用いて事前学習されたモデルであり, 一般性の高さが特徴である. 一方, Depth Pro は Apple による高解像度単眼深度推定モデルで, Vision Transformer(Vit) をベースに, 複数解像度の画像パッチを統合して高精度な深度マップを生成する手法を採用している.

1.2.2 モデル選定

この 2 種類のモデルを比較する. 室内を撮影し、図 1.1 の計測点を隠さないように何度もカメラの前に手をかざして距離推定の精度を評価した.

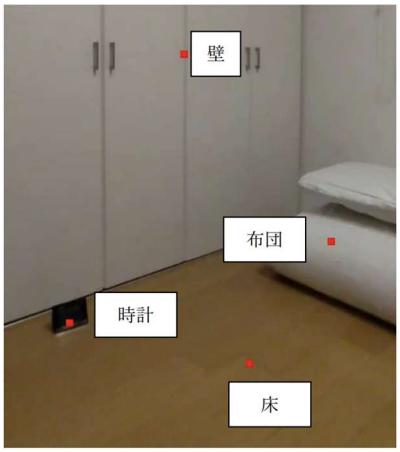

図 1.1 撮影したシーンと計測点

表 1.1 と表 1.2 にそれぞれ深度の真値, 各モデルで推定された深度の平均値, 範囲, 普遍分散を記す.

比較の結果, Depth Pro が精度面で優れていると判明した.

表 1.1: Depth Anything のデータ

計測点	真値[m]	平均値[m]	範囲[m]	普遍分散
壁	2.23	3.32	0.21	1.36E-3
床	1.67	2.42	0.15	6.84E-4
時計	1.90	2.69	0.14	8.38E-4
布団	1.91	2.75	0.23	1.46E-3

表 1.2: Depth Pro のデータ

計測点	真値[m]	平均値[m]	範囲[m]	普遍分散
壁	2.23	2.50	0.47	8.11E-3
床	1.67	1.76	0.29	3.81E-3
時計	1.90	1.97	0.33	4.27E-3
布団	1.91	2.18	0.38	6.34E-3

図 1.2 と図 1.3 に，それぞれのモデルの計測点”床”における推定値のヒストグラムを示す.

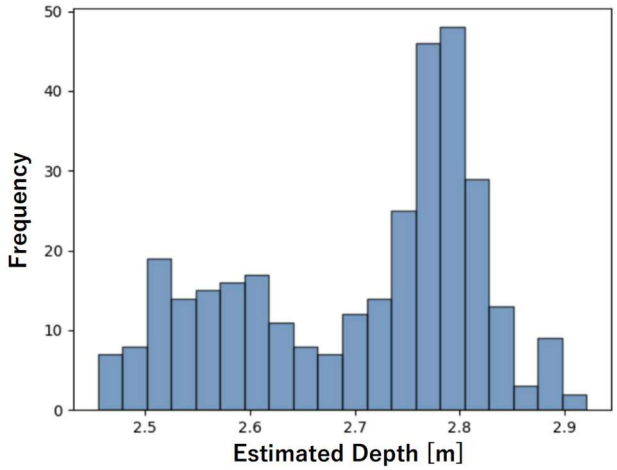

図 1.2: Depth Anything の”床”の推定値のヒストグラム

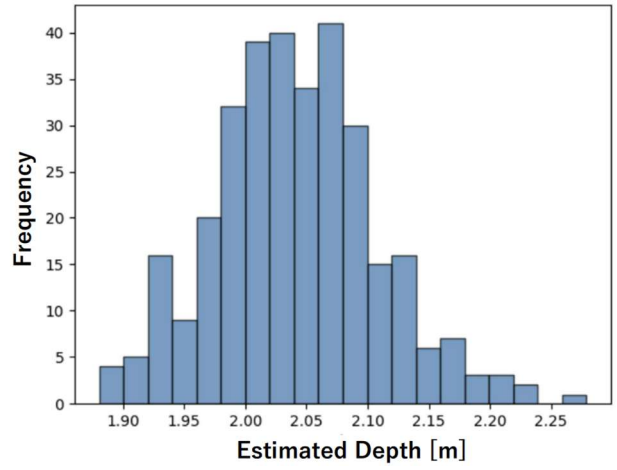

図 1.3: Depth Pro の”床”の推定値のヒストグラム

Depth Pro は単一ピークの分布を示していたのに対し，Depth Anything は二つのピークを持つ分布であった．これは，カメラに手をかざした際, Depth Anything が手前の物体に反応し，深度推定値が前方にずれる現象に起因すると考えられる．このような誤差を安定的に除去する手法が確立できなかったため，本研究では処理速度よりも推定精度を優先し, Depth Pro を採用した.

また, Depth Pro の性能について，実際に得られた深度推定値の平均誤差は約 0.45m であり，これは真値が 1.6~2.2m 程度の対象に対しておおよそ 20%前後の相対誤差に相当する．この精度を, Apple の発表論文[1]で報告されている δ_1 スコアと比較する．δ_1 スコアは画素数(n)，観測値(y_i)，真値(y_i^*)を用いて推定値が真値の 1.25 倍以内に収まる画素の割合として(1)で定義される[3]. 本観測では約 50%と推定され，公称値$\delta_1 \fallingdotseq 50\%$と整合的であった.

$$\delta_1 = \frac{1}{n}\sum_{i=1}^{n}\left(max\left(\frac{y_i}{y_i^*},\frac{y_i^*}{y_i}\right) < 1.25\right) \qquad (1)$$

2. 作成したシステム

2.1 概要

3 台の単眼 RGB カメラからのキャプチャと Depth Pro による深度推定に 1 台の PC を用い, RGB 画像と深度情報を別の描画用 PC へと送信し，三次元点群を生成してゲームエンジンである Unity 上で描画した.

3 つの深度の更新レートは 0.1~0.3FPS, Unity における描画レートは 14~17FPS であった.

表 2.1 に使用したカメラ，表 2.2 に深度推定に使用した PC，表 2.3 に描画に使用した PC の諸元を示す.

表 2.1: カメラの諸元

	カメラ 1, 2	カメラ 3
製造社	ELP	ELP
製品番号	Elp-usb8mp02g-sfv	Elp-usbgs720p02c-I180
外形寸法	幅 35[mm]×高さ 35[mm]×奥行 85[mm]	幅 40[mm]×高さ 40[mm]×奥行 25[mm]
質量	300[g]	50(g)
最大解像度	1280×768 [ピクセル]	1280×720 [ピクセル]
有効画素数	98 万画素	92 万画素
最大 FPS	30	60
画角	38[deg]	35[deg]

表 2.2: 深度測定に使用した PC の諸元

名前	DL3090
CPU	Intel®Core™i7-10700
GPU	NVIDIA GeForce RTX 3090
メモリ	32.0GB
Unity	2022.3.52f1
Python	3.10.0

表 2.3: 描画に使用した PC の諸元

名前	DESKTOP-9TVP8N0
CPU	Intel®Core™i7-11700
GPU	NVIDIA GeForce RTX 3060
メモリ	16.0GB
Unity	2022.3.52f1
Python	3.10.0

2.2.1 ピンホールカメラモデル

深度マップを三次元点群に変換する際，各画素の位置をピンホールカメラモデルに基づき三次元座標に変換する．焦点距離(f)と画像中心からのピクセル距離(x, y)，推定深度(d)を使い，点の三次元座標(X, Y)は(2)で表される．

$$X = \frac{x \cdot d}{f}, \quad Y = \frac{y \cdot d}{f} \tag{2}$$

2.2.2 スカート問題

点群をメッシュ化すると，深度差が大きな箇所でポリゴンが引き延ばされ，描画の破綻が生じることがある．この現象をスカート問題と呼称する．本研究では，隣接したピクセル間の深度差が一定以上の場合にポリゴンの生成を抑制し，破綻の緩和を図った(図 2.1).

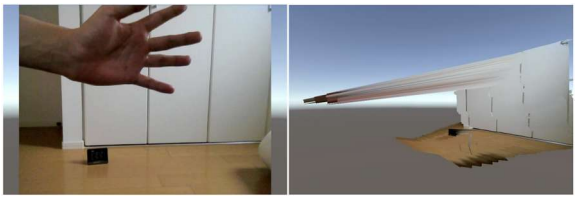

図 2.1: スカート問題が発生しているメッシュを
前から見た画像(左)と横から見た画像(右)

2.2.3 オーバーラップ問題

人物などの前景が手前に存在する場合，それらに遮られた背景の情報はカメラで取得できない．そのため，仮想空間上では背景に穴が開く．この問題をオーバーラップ問題と呼称する．本研究ではこの問題への対策として，機械学習を用いた人物セグメンテーションを適用し，前景と背景を 2 パスで分けて描画する手法を導入した.(図 2.2)

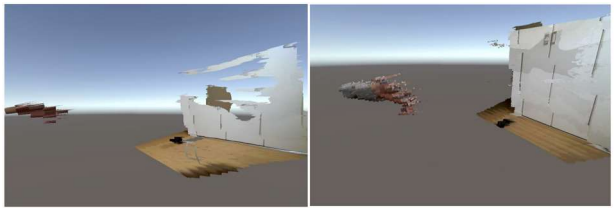

図 2.2: オーバーラップ問題により背景に穴が開いている
様子(左)と 2 パスで分けて描画した様子(右)

2.2.4 深度推定の誤差

深度推定には誤差があり，静止物体でも，フレームごとに前後に揺れて見えることがある．特に，複数の深度画像を合成する場面ではこの誤差が顕著になり，仮想空間の再構成精度を損なう要因となる．

一般的な対策として深度の平均値を取る方法があるが，これは物体が実際に動いている場合にも影響し，位置が不自然に構成される問題がある．そこで本プロジェクトでは最頻値フィルタによって，あるピクセルに対して複数フレームの中で最も頻出する値を選び，その値と最新の深度値が一致した場合のみ RGB 情報を更新する手法をとった。この最頻値計算を安定的に行うため，予め深度値を 16bit で量子化した．

この手法により静止背景の安定性が向上したが，処理時間が増すため，人物などの動きの大きい前景についてはフィルタを通さず点群として描画する．

2.3 点群のアラインメント

複数のカメラで取得した点群を共通の空間上で統合するには，各カメラの外部パラメータを世界座標系上で推定する必要がある．本研究では，既知の位置に特徴点を設置して撮影し，PnP 問題を解いて外部パラメータを求めた．特徴点には 1 辺 4cm の着色した発泡スチロールを使用し，アルミフレームで構成した直方体の頂点に取り付けた．その配置を図 2.3 に示す.

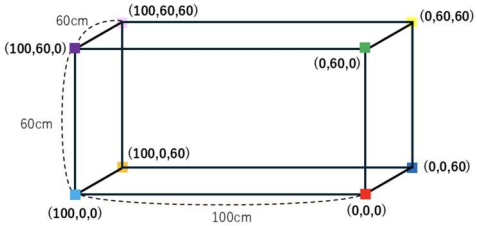

図 2.3: アルミフレームの構成と
実空間における特徴点の座標

PnP 問題を解くために用いた OpenCV の solvePnP 関数は右手系ロドリゲスの回転ベクトルを出力するが，幾つかのステップを経て Unity で用いる左手系クオータニオン表現に変換し用いた．

3. 結果

3 台の単眼カメラから取得した RGB 画像に対して Depth Pro による深度推定を行い，仮想空間上でリアルタイムに再構成した空間を図 3.1 に示す.

図 3.1: 再構成した空間

再構成された空間では，図 3.2 で示すように，中央の机など複数視点から観測された領域が概ね正確に合成された．

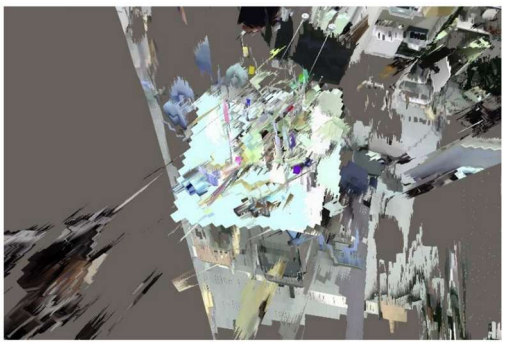

図 3.2: 中央の机を上から見た様子

一方で，カメラ間の距離と角度の差が大きい場合には，再構成にズレが生じ，対象物が二重に描画される例も見られた．このとき，現実空間の壁は一つであるのに対し，仮想空間上では 50cm ほどの差で二つの壁が描画されている．

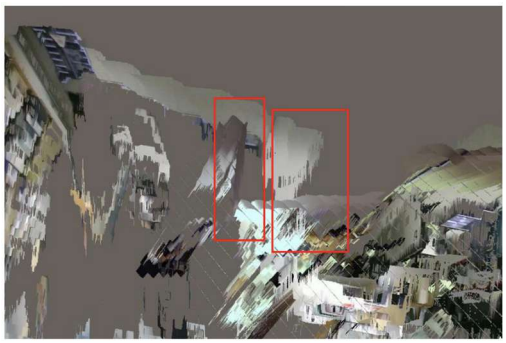

図 3.3: 二つの壁が描画される様子

4. 結論

本稿では，複数の単眼カメラと深度推定モデル Depth Pro を用いてリアルタイム実空間再構成を試みた．Depth Pro の推定誤差はほぼ同じ範囲に収まっており，人物セグメンテーションや最頻値フィルタなどの工夫は結果を大きく改善したが，根本的な深度精度の不足により物体の二重描画や破綻が発生し，実用的な意味では不十分なものであると結論した．

参考文献

[1] Yuxin Wang, Golnaz Ghiasi, et al. : Depth Pro: Sharp Monocular Metric Depth in Less Than a Second, 2024

[2] Lihe Yang, Bingyi Kang, et al. : Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data, CVPR 2024

[3] David Eigen et al : Depth Map Prediction from a Single Image using a Multi-Scale Deep Network, NeurIPS 2014.