

深度カメラを活用した RGB 画像からの奥行推定

S19096 小島 龍輝

1. はじめに

昨今、自動運転技術の発展に伴い、画像認識技術をアシストするべく LiDAR やミリ波レーダーの活用が進んでいる。これは画像から人物や車両の位置を推定する技術は進んでいるが、一枚の画像から奥行きを捉える技術が追いついていないからである。これによって物体の速度や距離がつかめず、衝突への時間の推定などが行えないため、LiDAR やミリ波レーダーによるアシストが必要となっている。

本研究では、RGB 画像から得ることができる各種情報を抽出し、深度カメラによる点群データを教師データとして活用した RGB 画像からの奥行きを推定する試みを行う。

2. 研究概要

使用機材は、深度カメラと RGB カメラを搭載した RealSense D435 (図 1) を使用した。

また、学習には敵対的生成ネットワーク (Generative Adversarial Network 以下 GAN) を使用した。これは入力された画像を本物か偽物か分類する識別器と、ノイズから画像を生成する生成器を互いに学習させ、画像の生成を行うというものである。



図 1 RealSense D435

本研究の大まかな流れは、まず RealSense カメラを用い、学習データを作成する。それを GAN ネットワークで学習させ、検証データを評価することで実用性を検証する。また、本研究は GAN ネットワークによる推論が実用的であるかを検証する初期段階であるため、学習データは研究室内の風景を撮影したものとした。

3. 学習データの作成

学習データの作成には、RealSense カメラで撮影した RGB 画像 (図 2 左) と、深度カメラ画像 (図 2 右) を一組の教師・学習データとして使用した。

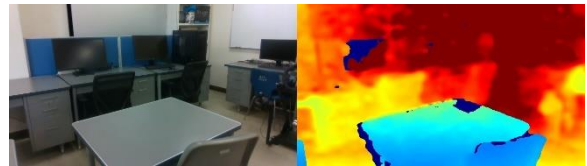


図 2 学習に使用した RealSense カメラ画像

またこの際、深度カメラの性質上、撮影対象の IR レーザーの反射特性によって距離情報の誤りや情報の欠損が生ずることがわかった。

そこで、色深度画像には OpenCV ライブラリを用いて情報の修正と欠損の補填を行った。

手順は以下の通りである。

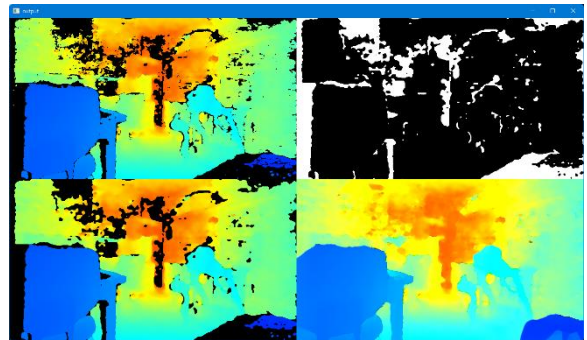


図 3 OpenCV による画像修正

1. メディアンフィルタを適用し、乱反射した小さな誤り深度情報をノイズとして除去(図3左上に対して左下が適用後)
2. 図3左下のフィルタ適用後画像の欠損部分のマスク画像を作成(図3右上)
3. 図3のマスク画像を使用し、Fast Marchingのアルゴリズムによる色情報の補完を行う(図3右下)

図2の学習データにはこの修正を適用した深度画像を使用している。

4. GANによる学習

GANによる学習には、Pix2Pix手法を使用した。本手法は入力画像を別の画像へ変換するというものであり、RGB画像を深度画像へ変換する本研究の目的にあったものであったからである。

初期段階として、GitHubにて公開されているTensorFlowを利用したpix2pixライブラリ⁴⁾を使用して学習を行った。検証のため学習データは200枚程度の小規模にとどめた。

結果は、図4のように物体を認識し、深度特徴を捉えたような画像が生成されることがあったが、図5のような複雑な情報を持つ入力画像に対しては、特徴量を捉えきれなかったような印象を受ける。

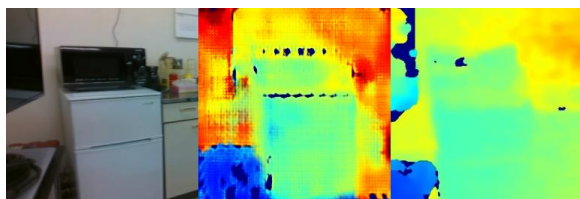


図4 成功例(左から入力,出力,正解画像)

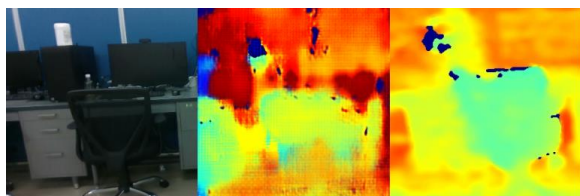


図5 失敗例(左から入力,出力,正解画像)

また、この結果から画像中の冷蔵庫に注目し、学習したオブジェクトに対し着色を行っているの

ではないかという点に疑問が生じた。その検証が以下の画像となる。図6の場合は近距離のため図4と同様の出力結果となったため正解と言える。また、図7の遠距離においては遠くにあるはずの冷蔵庫に対し、近距離を表す水色で着色しようとする動きが見られた。しかし領域の半分ほどを黄色で着色しているため、中・遠距離とみなす動きも見えるため、学習データを増やした場合にどういった振る舞いをするのか注目したい。



図6 冷蔵庫注目近距離画像

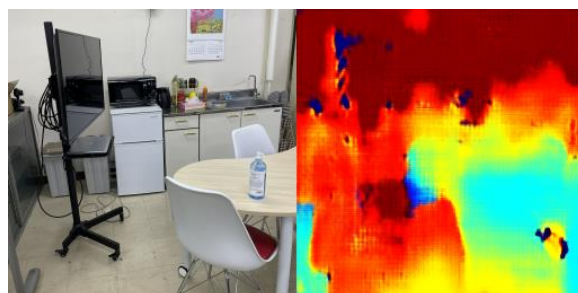


図7 冷蔵庫注目遠距離画像

5. おわりに

実験の結果によって、GANを用いた深度画像へ変換には一定の成果があった。

今後は、GANにAttention機構を組み込むことで、画像内の物体の前後関係を補填するような情報の追加を行い、精度向上が見込めるかの検証を行うつもりである。

6. 参考文献

- 1) pix2pix-tensorflow,
<https://github.com/affinelayer/pix2pix-tensorflow>