

# AAEpytorchの準備

---

ER17076 安井 理

# AAEの実装のために

---

- 前回までのミーティングの問題
  - トレーニングモデルを鈴木さんの方法で用意したい(OpenGLを使用しない)
- 解決法として考えた2パターン
  - 1:論文元のコード内のOpenGLで作成される部分を自作で用意した画像のimportするように書き換え
  - 2:Autencoderを用いて自分で既存手法に近づけたものを1から作成
- 前半週で主に1を調査したがコードが複雑で読み取ることができなかった
- 金曜日に三輪さんの作成中のAAEpytorchを共有
- 後半週にAAEpytorchの動作確認を行う準備

# 論文AAEのコードチェック

---

- 目的
  - OpenGLを使用してトレーニング画像生成・import部分のコード書き換え  
自分で作成したトレーニング画像を使用できるように変更
- 調査方法
  - モデルのimporエラーを起こしている場所から調べる  
→imporエラーを起こすモデルを用いてmodelがimporされている部分でエラーを起こさせる
  - メインプログラムのae\_train.pyをたどり、トレーニング画像がimporしているファイルを探す
- 結果として
  - モデルの撮影はmeshrenderファイルの複数のコードで作成されている
  - モデル画像をimporしているコードを見つけられなかった.
  - 自作でAutencoderを実装する方が良いと考えた

# AAEをPytorchでの実装

---

- AAEpytorchは三輪さんが作成済み  
→ [cu-milab/pro-depth-image-interpolation-using-ae](https://github.com/cu-milab/pro-depth-image-interpolation-using-ae)
- 自分で行う事
  - トレーニングデータ(モデル画像)の作成
  - 距離画像をオートエンコーダにかけるコードのため  
→ カラー画像を入力とするものに変更(チャンネル数など)
  - 加えるノイズも違うためノイズ部分の作成(コード内でノイズを加える)  
→ 背景画像・カラーノイズ・光・遮断物などのノイズのランダム配置
  - 姿勢推定部分が未作成のため作成

# トレーニング画像の作成

---

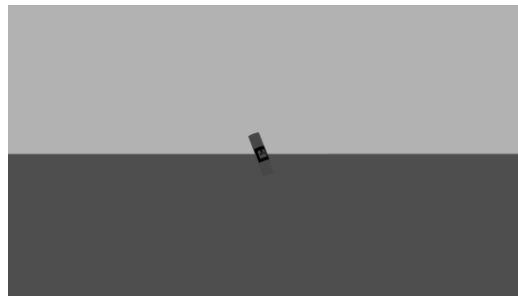
- Blenderで作成した円柱モデルを使用
- 作成方法: 鈴木さんの使用していたgazeboを用いたモデル画像の作成  
→Git-hub ~~cu-milab/pro-detection-of-deformed-AR-markers~~
- 鈴木さんのコードにバグがあったため三輪さんの作成したものを使用  
→Git-hub ~~cu-milab/pro-3D-database~~

距離画像を生成するものになっているためカラー画像を作成するよう変更

# トレーニング画像の作成

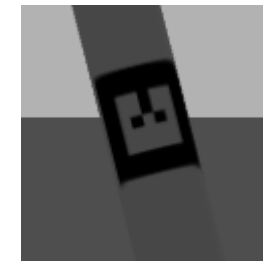
---

- モデル撮影方法
  - gazebo内のkinectで撮影
  - モデルの座標(xyz)は画像中央に来るよう固定
  - モデルの回転範囲をARマーカが移る範囲内でランダムに回転させる
  - 出てきた画像を `convert` コマンドを使用し、モデルのある座標を指定し  $128 \times 128$  ピクセルの画像に変更する



1920 × 1082

convert



128 × 128

# 現在

---

- 行ったこと
  - 円柱モデルで鈴木さんの行っていたgazeboを用いた方法で画像を用意
  - 画像作成後画像をまとめて128×128ピクセルにトリミングを行う
- 行う事
  - 三輪さんのコードを書き換え → トレーニングモデルができ次第動作確認
  - トレーニングモデルを現在は円柱だがより最適なものを考える
  - 姿勢推定部分が未作成

# 次週までにやる事

---

- 用意した画像でAAEpytorchの動作確認・精度の確認  
→ 昨日の時点でgazeboを用いたトレーニング画像を生成まで終了
- ノイズを加えるプログラムの追加  
→ Domain Randomizationを参考に作成
- 姿勢推定部分の再調査  
→ 姿勢推定部分については未実装のため論文を読み返しながら実装方法を考える