

# AAEの動作確認と実装について

---

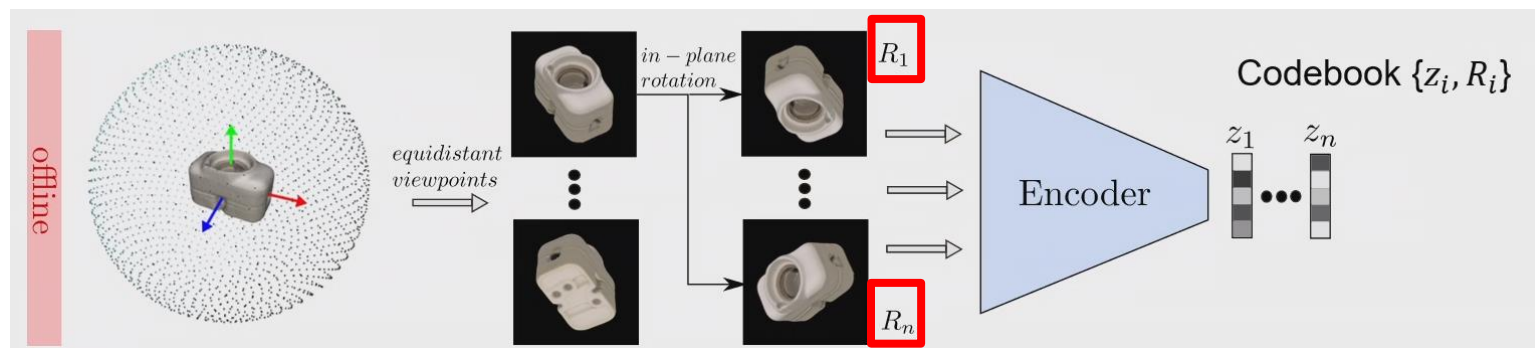
ER17076 安井 理

# 目的

---

- ・リアルタイム検出時における計算量軽減部分の調査
- ・AAEのGit-hubの実装環境を整え動作確認を行うこと
- ・鈴木さんの研究,目標を含め自分の研究の最終目標を考える

# 検出と推定

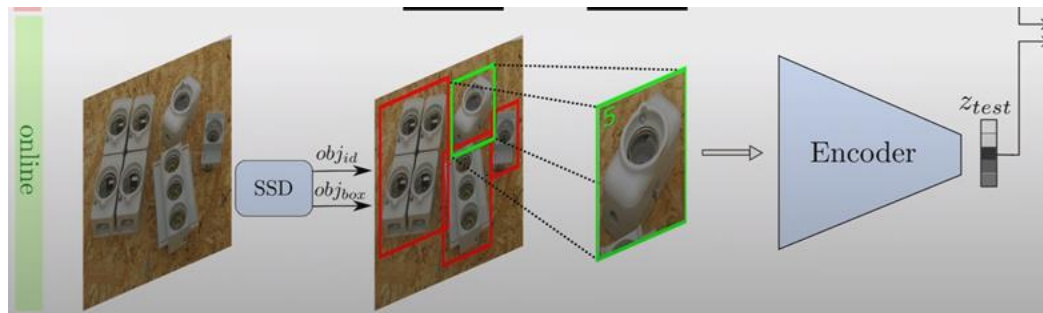


## オフライン: AAEのトレーニング

- ・等間隔に物体を回転させ、すべての角度の視点 $z_i$ (2562点)を用意
- ・物体の各視点につき画内回転 $R_i$ (36)点
- ・AAEにかけ潜在変数 $z \in R^{128}$ を生成して $(z_i, R_i)$ を一つの物体IDに92232個記録

<https://www.youtube.com/watch?v=jgb2eNNIPq4>

# 検出と推定



オンライン: AAEを用いた推定

- ・エンコードされた潜在変数を次の順番でコサイン類似度を求める

1. 視点 $z_i$  (2562) 点
2. 画内回転 $R_i$  (36) 点

- ・一つの物体IDに92232個記録  $\Rightarrow$  約2600個の計算で推定可能

<https://www.youtube.com/watch?v=jgb2eNNIPq4>

# 動作環境

---

- Linux, Python 2.7
  - TensorFlow >= 1.6
  - OpenCV >= 3.1
  - PyOpenGL==3.1.4 PyOpenGL\_accelerate==3.1.4
  - tensorflow-gpu==1.6.0
  - tensorboard ==1.6.0
- 
- TensorFlowのバージョンに合わせ,  
他のバージョンも合わせないといけないものが多い

# 動作確認(セットアップ)

---

- .bashrcにワークパスの設定の記述

```
AE_WORKSPACE_PATH=/path/to/autoencoder_ws
```

- ディレクトリの作成(ホームに作成)

```
mkdir ae_workspace_path
```

- 作成したディレクトリに移動しコマンドで入力

```
ae_init_workspace
```

- ディレクトリにファイルが生成される

# 動作確認(トレーニング)

---

- ・作成されたファイルに移動し実行

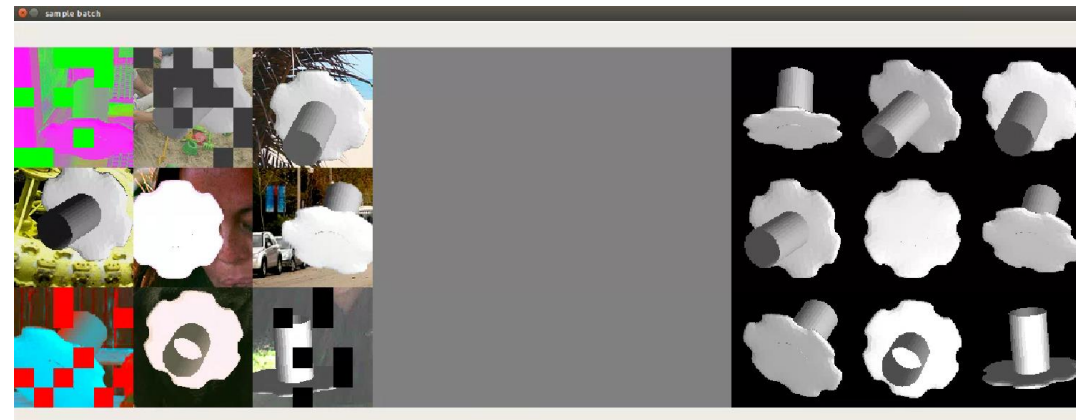
```
ae_train exp_group/my_autoencoder -d
```

- ・ノイズを加えた画像の生成

```
ae_train exp_group / my_autoencoder -d
```

- ・別のターミナルでトレーニングを開始

```
ae_train exp_group / my_autoencoder
```



# 動作確認

---

- ・現在トレーニング中(現在,リモートデスクトップオフライン:PCが落ちている可能性)
- ・bash部分の記述・実行プログラムなどのpathの設定が必要なものが多くある
- ・3dのデータセットがないため他のGit-hubから持ってくる必要がある



# 実装に向けて

---

- ・テスト(推定)部分がまだ行えていないため,確認を行う
- ・SSDでの動作確認ができればFaster-RCNとの互換性を調べる
- ・ARマーカを四角の形のみの姿勢推定だとデータ量が軽くなるのでは.  
⇒ID検出と四角形の推定を分けて考える
- ・PCが落ちてしまう原因を調査する必要がある.

# 今後の予定

---

- ・PCが落ちる原因の究明と解決を行う
- ・テスト(検出推定)の流れの動作確認を行う
- ・ARマーカのモデルでトレーニング・テストを行う.
- ・トレーニング時にARマーカだとノイズに弱くなるのではないかと考えられる  
⇒姿勢推定は輪郭で行った方がいい..?

# 参考文献

---

- 6次元物体検出の論文

[http://openaccess.thecvf.com/content\\_ECCV\\_2018/papers/Martin\\_Sundermeyer\\_Implicit\\_3D\\_Orientation\\_ECCV\\_2018\\_paper.pdf](http://openaccess.thecvf.com/content_ECCV_2018/papers/Martin_Sundermeyer_Implicit_3D_Orientation_ECCV_2018_paper.pdf)

- git-hub     <https://github.com/DLR-RM/AugmentedAutoencoder#testing>