

変形ARマーカの姿勢推定

ER17076 安井 理

内容

- 変形ARマーカを平面状に戻すAAEの作成
- 姿勢推定部分の作成

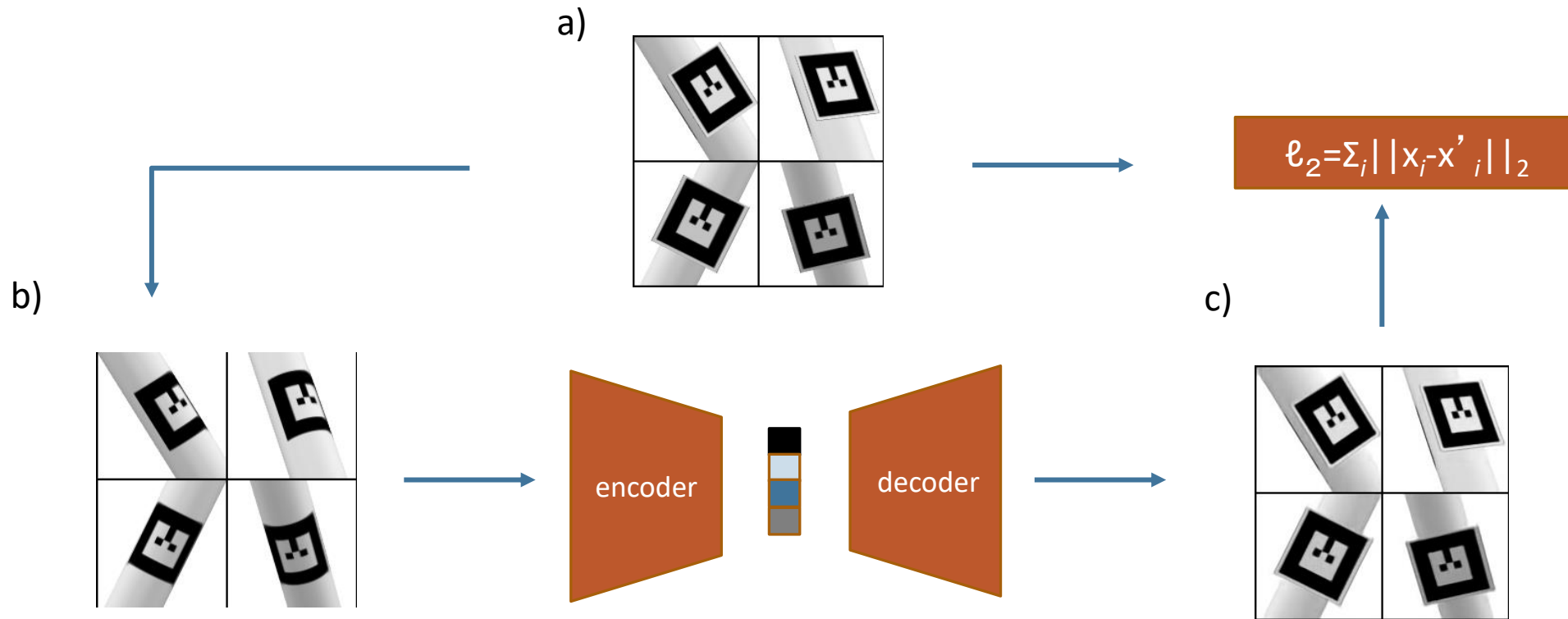
- 評価方法の決定
- 現段階での精度の確認

手順

- AAEを用いて姿勢推定を行うために必要なこと
 - AAEの学習済みの重み
 - 円柱に張られたARマーカを平面状に戻すように学習されたAAEの重み
 - 姿勢情報を持つARマーカモデル画像の作成
 - 円柱の姿勢角度roll,pitch,yawそれぞれ $[-28^{\circ} \sim 28^{\circ}]$ まで 2° 飛ばしで,合計21952枚作成
 - encodeされる潜在変数の書き出し
 - 各姿勢画像の潜在変数をtxtに書き出していく
 - cos類似度による推定姿勢の決定
 - 潜在変数のcos類似度が最も近い姿勢情報を姿勢として推定する

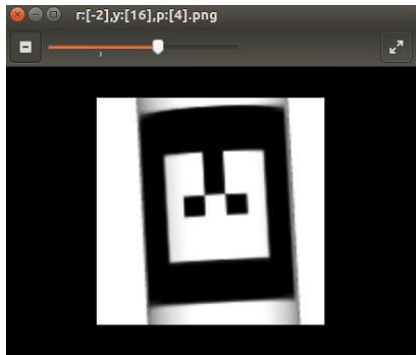
AAEの学習

- 円柱に貼られたARマーカを平面状に直すように学習を行う
 - b)の入力を変形ARマーカとして、平面状のARマーカc)を出力するよう学習
 - 学習された重み(500epoch)を推定時に使用



姿勢情報を持つARマーカの作成

- 円柱に貼り付けられたARマーカのモデルの姿勢を用意
- gazebo上で円柱の姿勢(roll,pitch,yaw)
それぞれ $[-28^{\circ} \sim 28^{\circ}]$ までの範囲で 2° ずつ進めて撮影
→合計21952($28*28*28$)枚作成
- 画像の保存名が「r:[roll],p:[pitch],y[yaw].png」となるように設定しておく



r:[2],p:[16],y[4].png

encodeされた潜在変数の書き出し

- 姿勢情報を持つ各姿勢の画像をAAEに入力
- encoderから得られる潜在変数 z_n (128)を128行「画像名.txt」に書き出す
- txtファイルは姿勢情報画像分の21952個分生成
- 保存名「 r:[roll],p:[pitch],y[yaw].txt」とする

cos類似度による姿勢の決定

- 推定したいARマーカの画像をAAEに入力
潜在変数 z_{test} をtxtに書き出す
- 潜在変数 z_{test} と z_n のcos類似度を計算し最も近い z_n を姿勢とする
- cos類似度式(1)は, 要素の近さを求める計算

$$cos_i = \frac{z_{test} z_n}{||z_{test}|| ||z_n||} \quad (1)$$

評価方法の決定

現在考えている評価方法

- ・ARマーカモデル, 半径20,30,40,50[mm] の円柱を各モデル姿勢10 枚ずつ用意.
 - ・撮影されたモデル画像をSSDを用いて検出, バウンディングボックスの画像を使用
 - ・画像を(128*128)にリサイズし, 入力. 提案手法により姿勢推定を行う.
 - ・①評価用画像のモデル姿勢と提案手法により推定された姿勢の[roll,pitch,yaw]を比較し誤差を評価.
 - ・②ar_track_alvarを用いた姿勢推定との推定可能な変形度合いの比較の評価.
- ・どのような評価方法が最適か, 考え中.

姿勢推定の検証

- cos類似度を求めるコードの精度の検証を行った.
- 用意した評価用データはgazebo上にモデルの姿勢を指定して表示, 撮影
- 検証用の姿勢画像は5枚用意
- 画像サイズを 128×128 に切り取り, AAEに入力
- cos類似度計算を行い姿勢推定を行った.

姿勢推定の検証

検証用画像(正解)

r:[0.0],y:[0.0],p:[0.0]

r:[17.18],y:[12.60],p:[-2.86]

r:[20.05],y:[20.05],p:[20.05]

r:[-4.01],y:[7.44],p:[-9.16]

r:[-5.72],y:[-8.59],p:[-11.45]

推定された姿勢

r:[8],y:[-20],p:[-20]

r:[16],y:[4],p:[22]

r:[8],y:[0],p:[26]

r:[0],y:[-12],p:[-26]

r:[8],y:[-24],p:[-24]

姿勢推定の検証

- 検証結果の考察
 - 姿勢推定が正しく行えていない結果となった
- 原因として、姿勢情報用に用意した画像に問題がある可能性
→ 姿勢情報用画像を作り直し、再度検証

参考文献

- 6次元物体検出(AAE)の論文

http://openaccess.thecvf.com/content_ECCV_2018/papers/Martin_Sundermeyer_Implicit_3D_Orientation_ECCV_2018_paper.pdf

- AAEを実装しているgit-hub

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