

# 深層学習で抽出した 特徴量を用いた 特徴点マッチング

---

2019/2/15

67150120 梶原奏美

# 背景

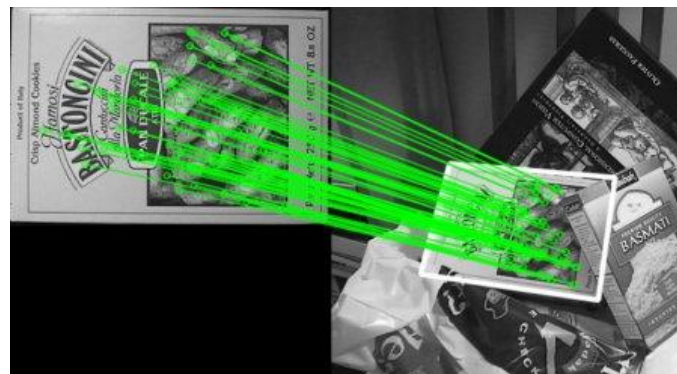
## 【特徴点マッチング】

- 画像からの形状復元



<https://github.com/cdcseacave/openMVS/wiki/Usage>

- 物体認識



など

[http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py\\_tutorials/py\\_feature2d/py\\_feature\\_homography/py\\_feature\\_homography.html](http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_feature2d/py_feature_homography/py_feature_homography.html)

コンピュータビジョンの分野で重要な役割

# 背景

---

AKAZE SIFT SURFなど

改善することで...

コンピュータビジョンの分野で役に立つ

改善の余地がある

# 背景

---

## 【深層学習】

- 機械学習の発展手法
- コンピュータビジョン分野で成果を挙げている



特徴点マッチングに深層学習を用いて改善

# 目的

---

深層学習で抽出した特徴量による  
特徴点マッチングの精度向上

# 特徴点マッチングとは

異なる画像間で、固有の点を対応付けること。



特徴点検出

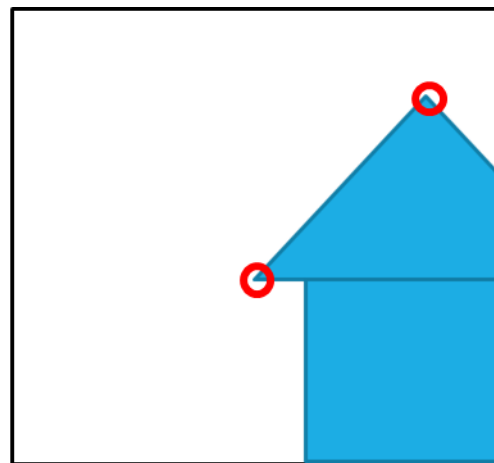
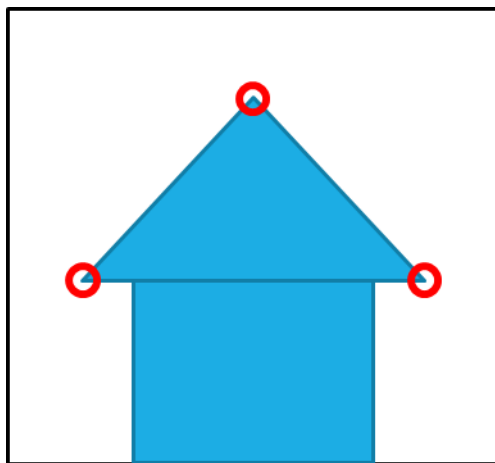
特徴量記述

マッチング

# 特徴点検出

画像中から特徴点の座標を検出する。

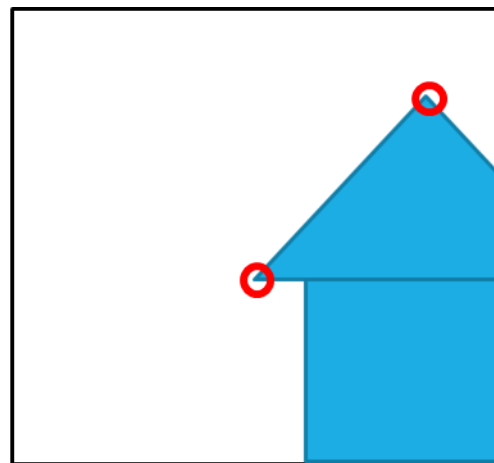
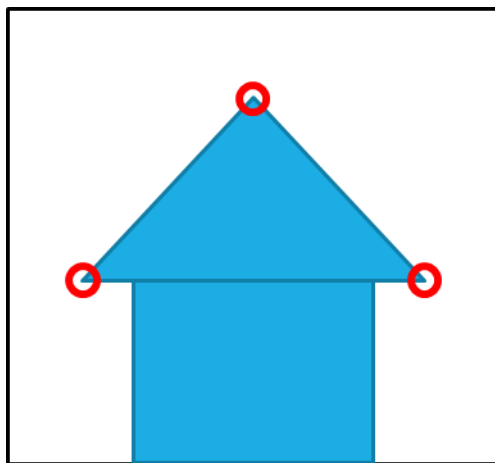
他と区別できるような固有の点  
対象物の角や線の交わり/エッジ(明るさの変わる場所)



# 特徴量記述

特徴点から特徴量を算出する。

特徴点の固有性を  
ベクトルやバイナリコードの数値で表現したもの





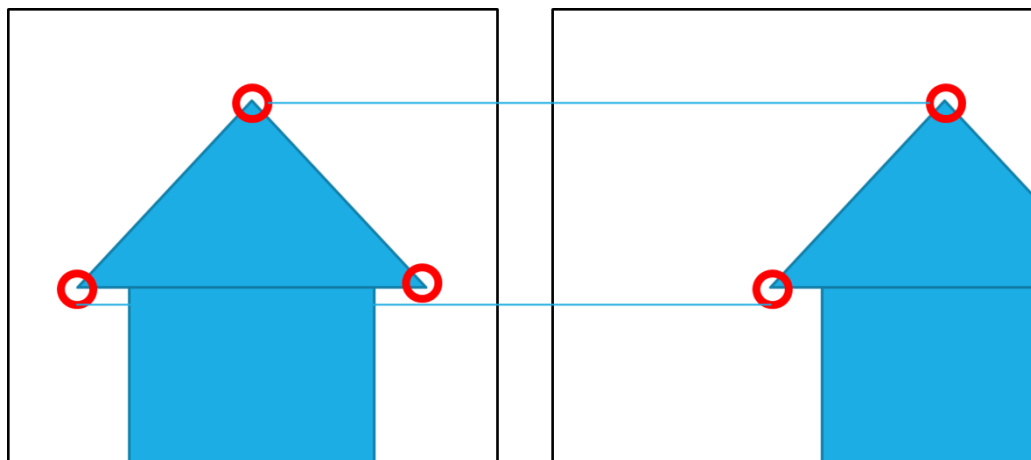
# マッチング

---

類似度の高い特徴点同士を対応付ける。



特徴量間の距離が近いもの



# AKAZE



- 他の手法に比べ、ロバスト性が向上している
- 拡大縮小に強い
- 特徴量：深層学習を用いる。

【提案手法】  
深層学習を用いる。

# 特徴量記述

---

- ① 特徴点の周辺 $D \times D$ 画素で切り取る。  
オリエンテーション を考慮。

特徴点における方向  
勾配の方向、強度により求める

- ② 切り取った画像から特徴量を抽出。  
学習済みの公開モデル(bvlc\_reference\_cafenet)を使用  
学習データ：ILSVRC2012の  
1000カテゴリ、120万画像データ  
次元数：4096次元（fc7層の出力）

# マッチング

---

- 特徴量間の距離計算：cos距離

$$\frac{\vec{q}_i \cdot \vec{p}_j}{|\vec{q}_i| |\vec{p}_j|}$$

$\vec{q}_i$  : 1つ目の画像の特徴点  $i$  の特徴量  
 $\vec{p}_j$  : 2つ目の画像の特徴点  $j$  の特徴量

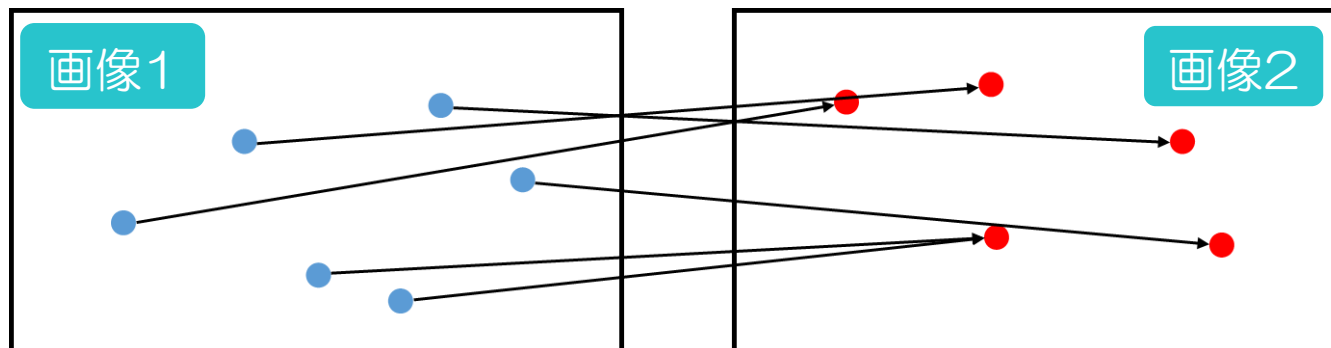
→ 値が1に近いほど類似している。

- 『総当たりマッチング』 + 『クロスチェック』

# マッチング

## 『総当たりマッチング』

画像1から抽出したそれぞれの特徴点について  
画像2中で最も特徴量間距離が小さい特徴点と対応付ける。



## 『クロスチェック』

画像1→画像2の総当たりマッチングと  
画像2→画像1の総当たりマッチングの双方の結果で  
一致するもののみをマッチング結果

# 実験目的

---

特徴量記述能力の評価のため  
提案手法とAKAZEでの比較実験

# 使用画像

画像の中心を原点とし、  
5, 15, 35, 45, 90度で回転させ、  
それぞれ50%~150%で拡大縮小した画像



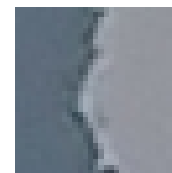
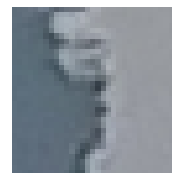
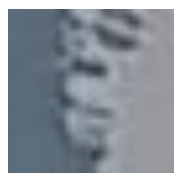
# 特徴量記述

---

【特徴点検出】 AKAZE

【切り取り画像】

16×16, 32×32 の2通り





# マッチング成功率

---

$$\text{マッチング成功率} = \frac{\text{正しい対応点の数}}{\text{得られた対応点の数}}$$

※ 正しい対応点  $|Q_i - P'_j| < T$

$Q_i$  : 元画像の  $i$  番目の特徴点座標

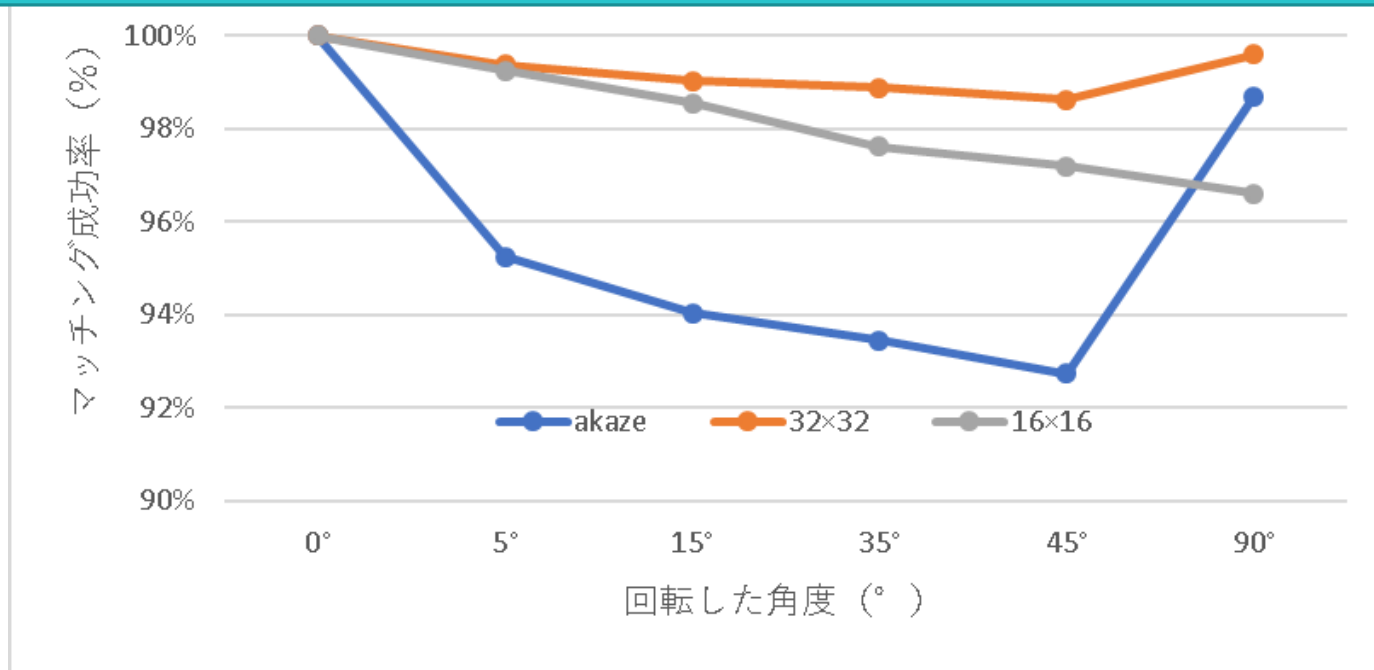
$P_j$  : 回転画像の  $j$  番目の特徴点座標

$P'_j$  : 回転分、時計回りに回転させた座標

$T$  : 許容座標誤差

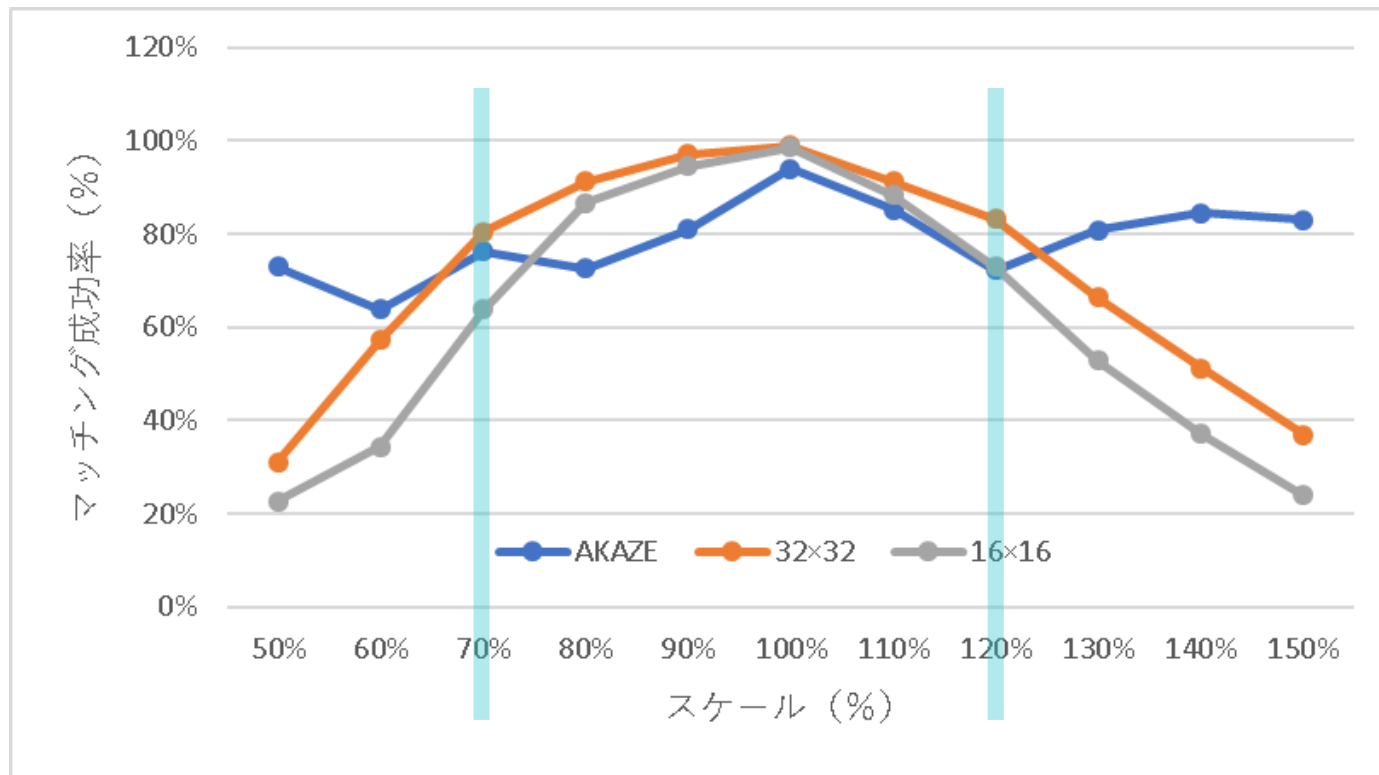
【許容範囲誤差  $T$ 】  $T = 10$

- $32 \times 32$ が $16 \times 16$ より上回っている
- どの回転角度でも提案手法が同等以上



スケール100%のときのマッチング成功率

# スケール70%~120%：提案手法 スケール変化が大きい：AKAZE



15° 回転させたときのマッチング成功率

# 考察

---

学習済みモデルが物体認識用  
小さい画像に向いていない

マッチング精度向上した  
特徴量記述の能力が高い

スケール変化について考慮していない  
改善が必要

# まとめ

---

深層学習を特徴量記述に用いた  
→ マッチング精度が向上

## 【今後の課題】

- スケール変化への対応
- 深層学習で特徴点検出

---



# 予備実験 [AKAZEの精度評価]

---

## 【使用画像】

画像の中心を原点とし、  
反時計回りに45° 回転させた画像



## 【マッチング方法】

距離：ハミング距離

『総当たりマッチング』 + 『クロスチェック』

# 予備実験 [AKAZEの精度評価]

【マッチング率】

$$\text{マッチング率} = \frac{\text{正しい対応点の数}}{\text{得られた対応点の数}}$$

※ 正しい対応点  $|Q_i - P'_j| < T$

$Q_i$  : 元画像の  $i$  番目の特徴点座標

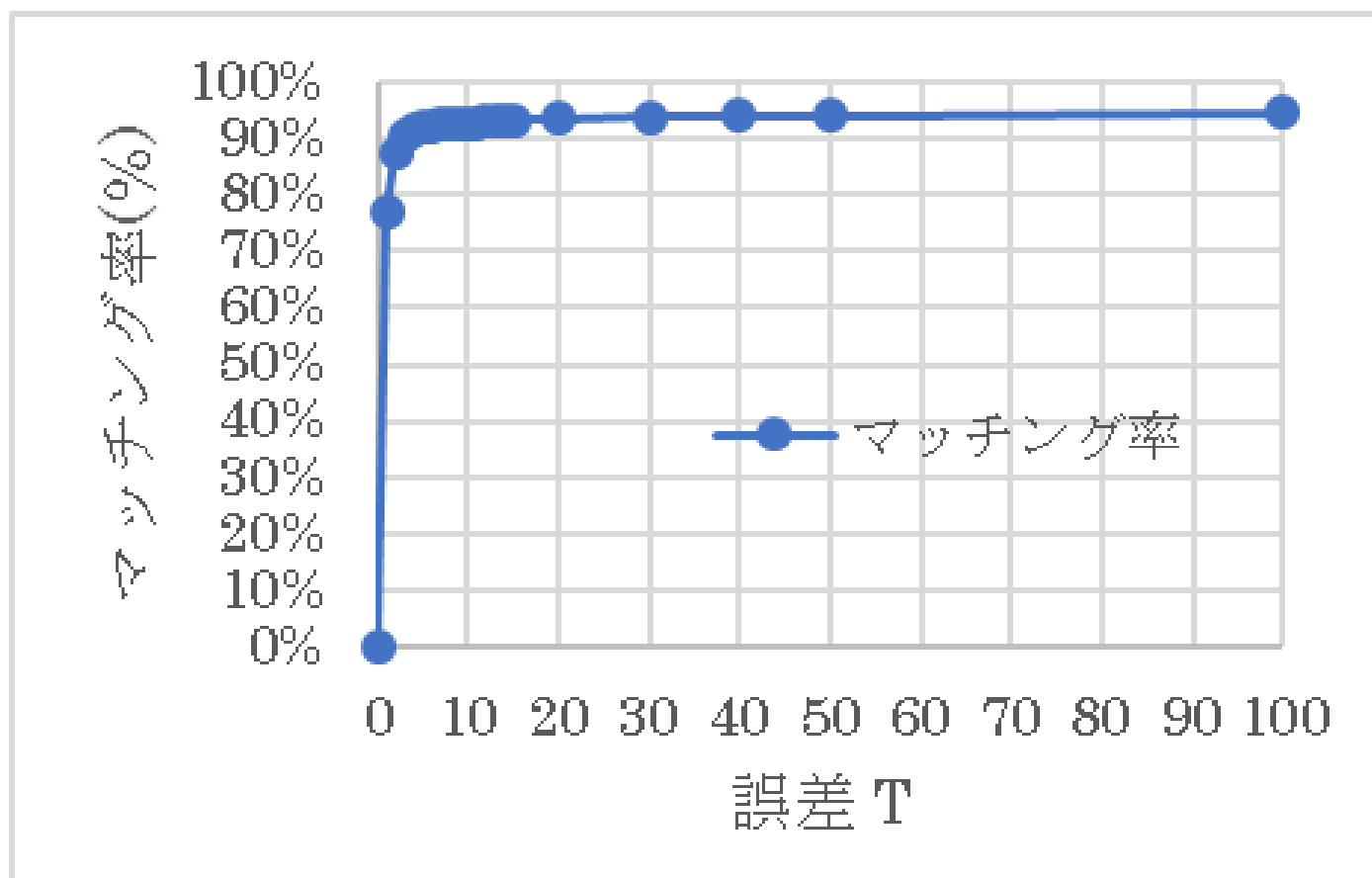
$P_j$  : 回転画像の  $j$  番目の特徴点座標

$P'_j$  : 時計回りに  $45^\circ$  回転させた画像座標

$T$  : 許容座標誤差

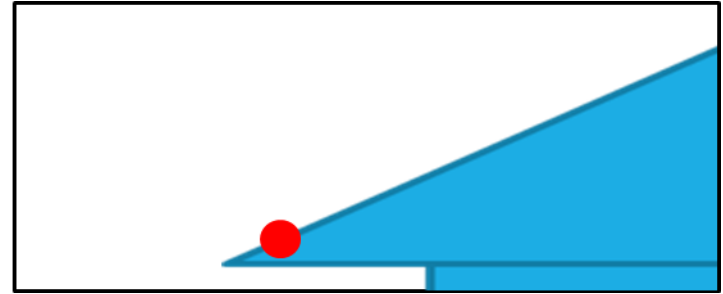
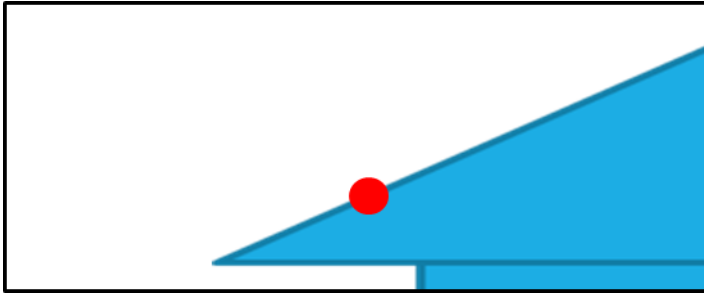


# 結果

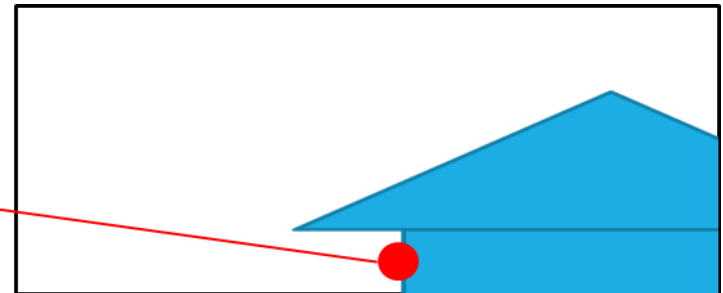
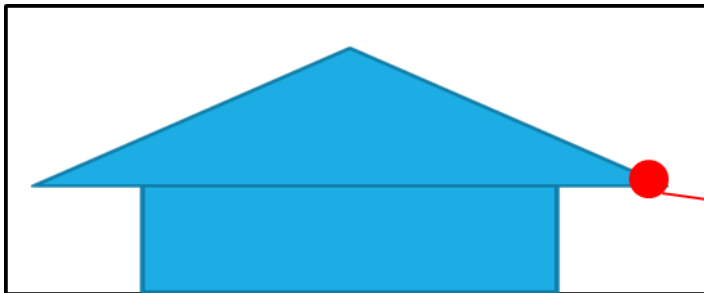


# 結果

- $T < 10$  マッチング率の変化が大きい



- $T \geq 10$  変化が小さい



# 結果

100%

マッチング率 約 93%

正しい対応点の数 6598  
得られた対応点の数 7115 (T = 10のとき)

0 10 20 30 40 50 60 70 80 90 100

誤差 T

# AKAZE



- 他の手法に比べロバスト性や処理速度で優れている
- 拡大縮小不変(スケール)、回転不変(オリエンテーション)
- 特徴量：バイナリコード

# 結果

---

	0°	5°	15°	35°	45°	90°
50%	25.23%	21.10%	22.60%	20.32%	17.81%	22.65%
60%	39.59%	37.54%	34.41%	33.81%	31.55%	37.64%
70%	66.13%	65.40%	63.87%	61.20%	61.45%	62.21%
80%	88.72%	86.91%	86.75%	84.85%	83.33%	83.63%
90%	97.11%	95.53%	94.57%	94.28%	93.60%	92.71%
100%	100.00%	99.24%	98.56%	97.62%	97.20%	96.60%
110%	95.86%	90.52%	88.27%	86.27%	84.44%	85.98%
120%	85.16%	74.76%	72.90%	70.77%	70.02%	73.82%
130%	67.35%	55.76%	52.77%	50.97%	50.15%	55.04%
140%	48.87%	38.45%	37.04%	35.90%	35.01%	39.58%
150%	35.61%	25.37%	24.03%	20.71%	22.17%	28.67%

# 結果

---

	0°	5°	15°	35°	45°	90°
50%	35.05%	35.35%	31.15%	33.49%	30.65%	37.44%
60%	60.75%	61.65%	57.25%	53.75%	51.72%	59.86%
70%	81.13%	78.88%	80.51%	76.91%	77.64%	80.32%
80%	91.83%	91.40%	91.32%	91.21%	90.89%	91.48%
90%	98.61%	97.57%	97.21%	96.69%	97.12%	98.34%
100%	100.00%	99.37%	99.02%	98.89%	98.62%	99.60%
110%	95.71%	91.19%	91.26%	90.37%	89.69%	93.69%
120%	88.90%	83.04%	83.21%	81.03%	81.47%	86.11%
130%	76.23%	66.04%	66.38%	65.97%	66.28%	74.48%
140%	62.47%	52.62%	51.13%	48.91%	48.92%	59.07%
150%	47.18%	34.66%	36.91%	33.89%	35.03%	45.16%

# 結果

---

	0°	5°	15°	35°	45°	90°
50%	86.80%	72.96%	72.96%	64.25%	64.85%	86.97%
60%	80.58%	68.66%	63.92%	63.87%	63.39%	79.95%
70%	88.52%	79.59%	76.20%	77.29%	75.99%	88.26%
80%	82.38%	75.79%	72.67%	73.06%	72.62%	83.23%
90%	88.39%	83.01%	81.04%	81.34%	82.15%	87.46%
100%	100.00%	95.25%	94.03%	93.45%	92.73%	98.70%
110%	88.95%	86.62%	85.32%	84.95%	84.76%	89.10%
120%	76.85%	73.84%	72.16%	72.35%	72.98%	77.07%
130%	83.91%	82.14%	80.92%	80.53%	81.42%	82.69%
140%	86.40%	85.71%	84.46%	84.34%	84.18%	85.94%
150%	84.96%	83.36%	83.07%	81.76%	82.05%	84.10%