

# 深層学習を用いた 特徴点マッチングにおける 回転・拡大縮小不変の実現

---

2020/02/14

67160140 黒河英里子

# 背景

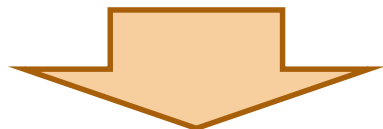
---

## 特徴点マッチング

- 画像からの形状復元
- 物体認識

コンピュータビジョンの分野で重要な役割

AKAZEやSIFTなど多く研究されている



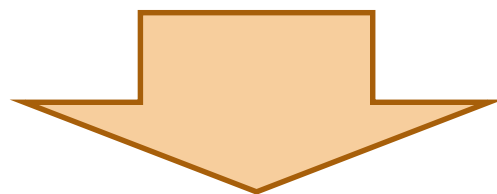
マッチング精度には改善の余地あり

# 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た研究も行われている

[1] Y. Ono, E. Trulls, P. Fua, K. M. Yi, “LF-Net: Learning Local Features from Images”, (2018), arXiv:1805.09662v2[cs.CV].

# 従来手法

---

梶原奏美, H30年度卒業論文[2]

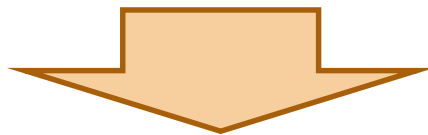
“深層学習で抽出した特徴量を用いた特徴点マッチング”

## 特徴点検出

- AKAZEを使用

## 特徴量記述

- 深層学習を使用

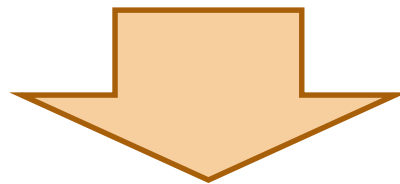


- 拡大縮小変化に対応していない
- AKAZEに依存している

# 本研究の目的

---

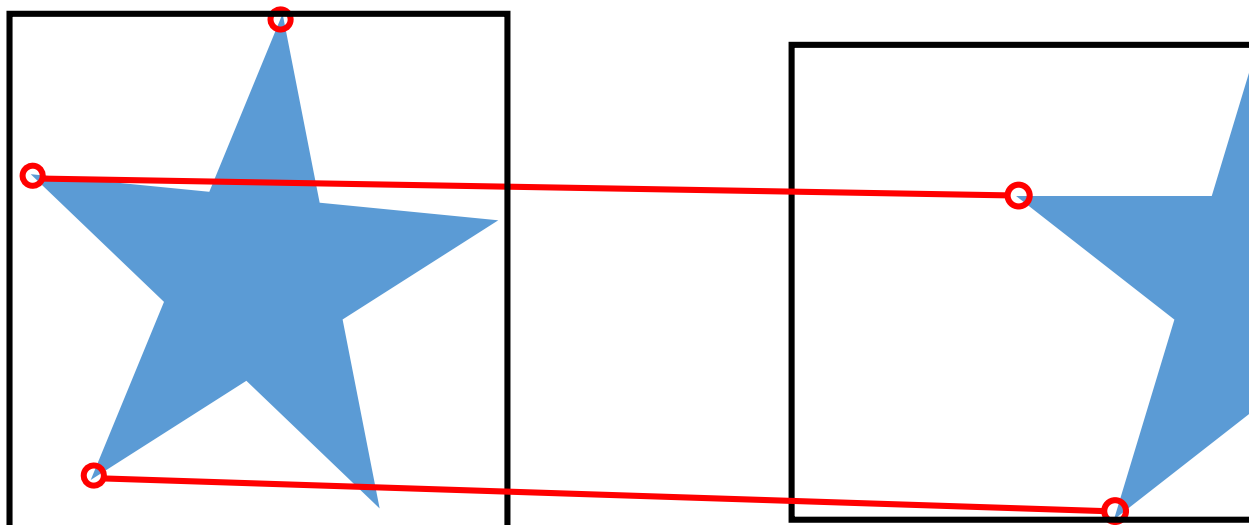
従来手法を改善



- AKAZEに依存することなく特徴点検出
- 回転・拡大縮小不変を実現
- 特徴点マッチングの精度向上

# 特徴点マッチング

異なる画像間で固有の点を対応付けること



特徴点検出

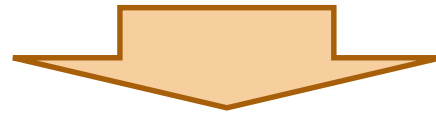
特徴量記述

マッチング

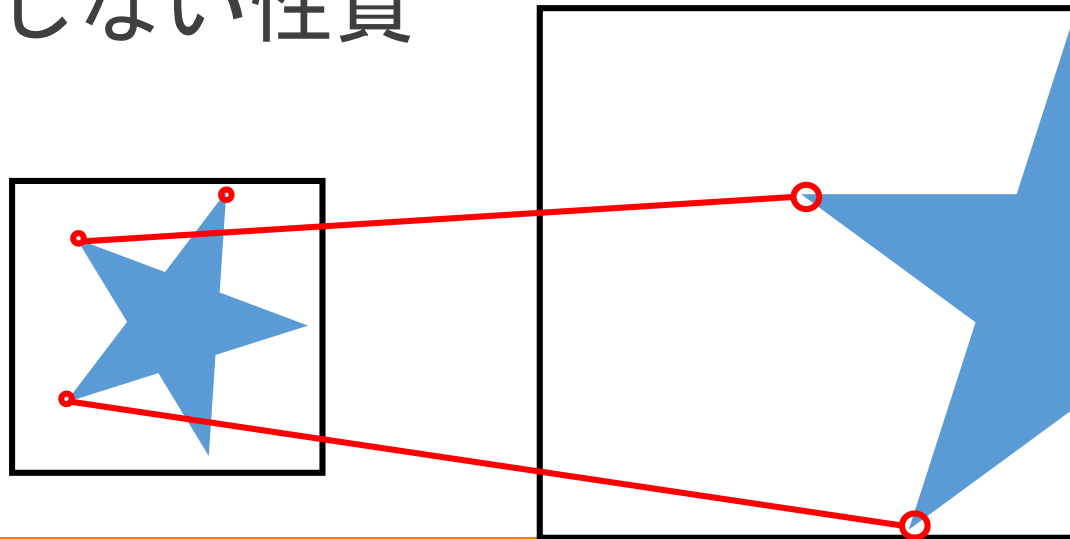
AKAZE

# 回転・拡大縮小不変

特徴点マッチングを行う2枚の画像間では  
対象物体が異なる大きさや角度で写っている

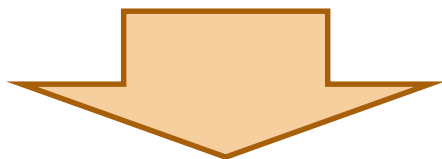


回転、拡大縮小が特徴点マッチングに  
影響しない性質

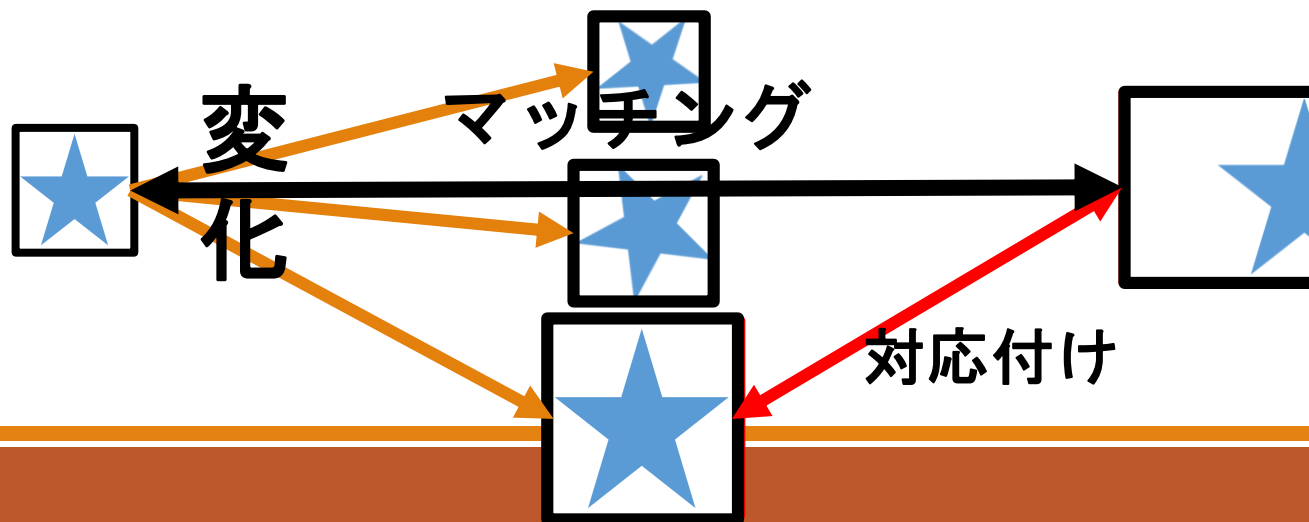


# 回転・拡大縮小不変の実現

数段階に変化した画像のいずれかに近ければ  
正しい対応付けが得られる

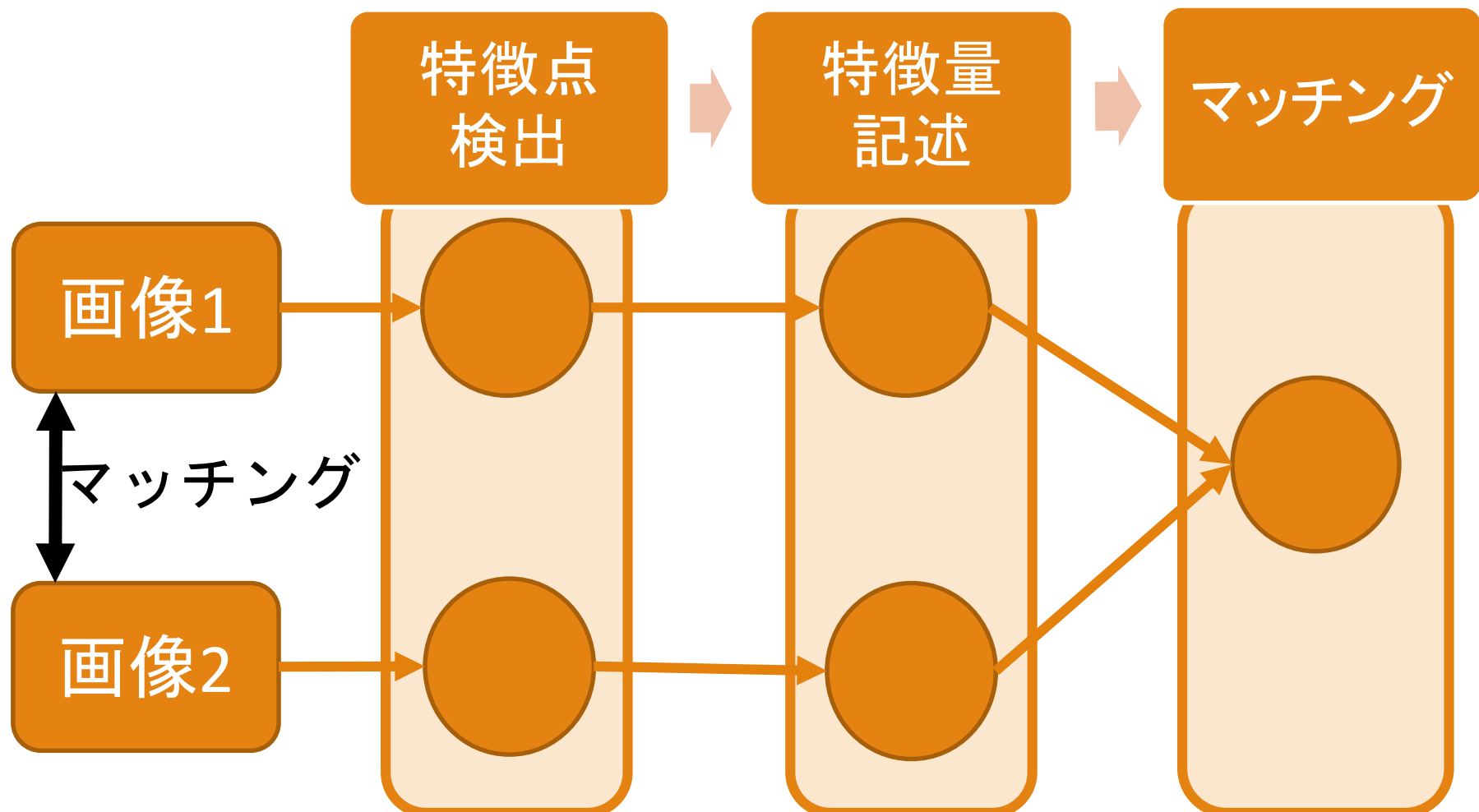


回転・拡大縮小不変を実現



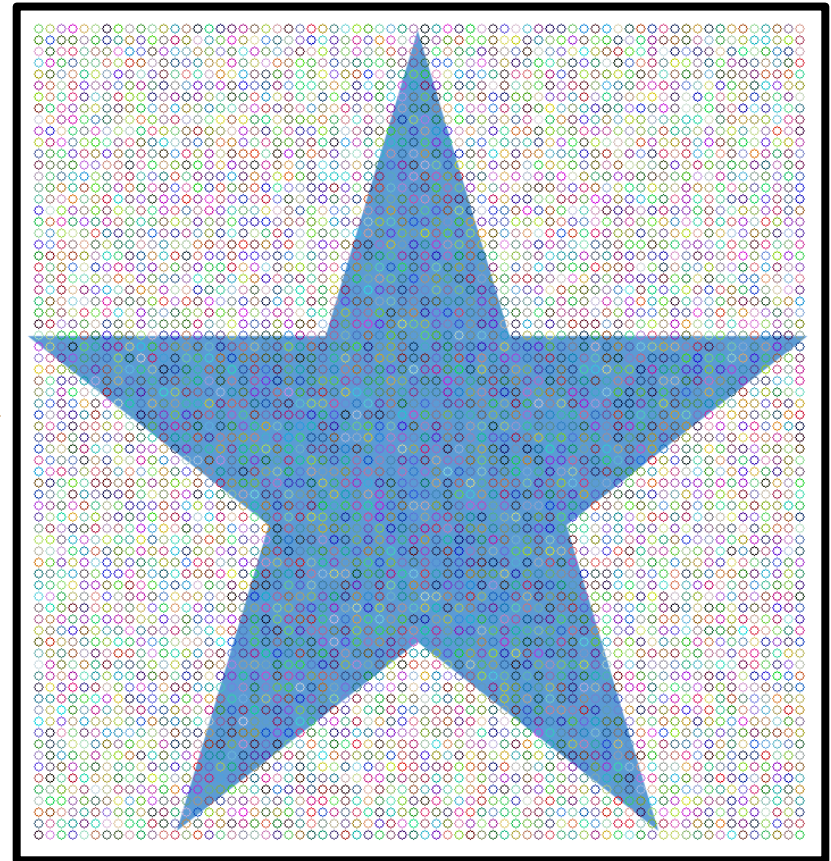
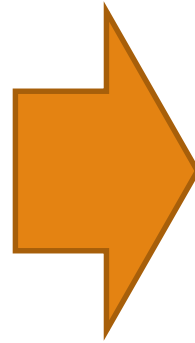
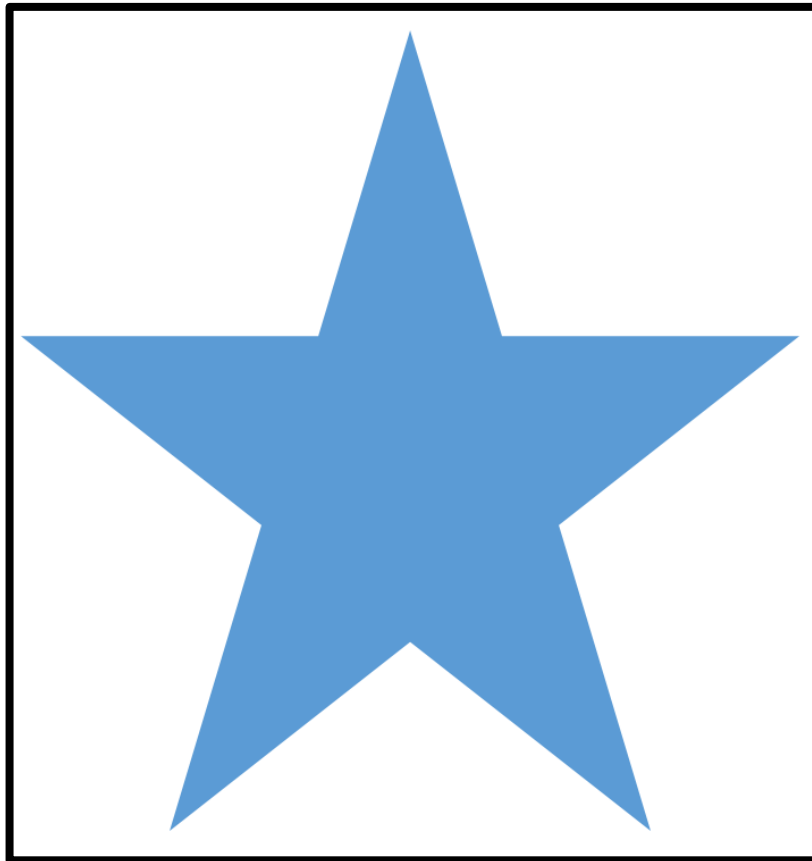


# 提案手法



# 特徴点検出

画像の  $N = 10$  画素間隔の点をすべて特徴点とする

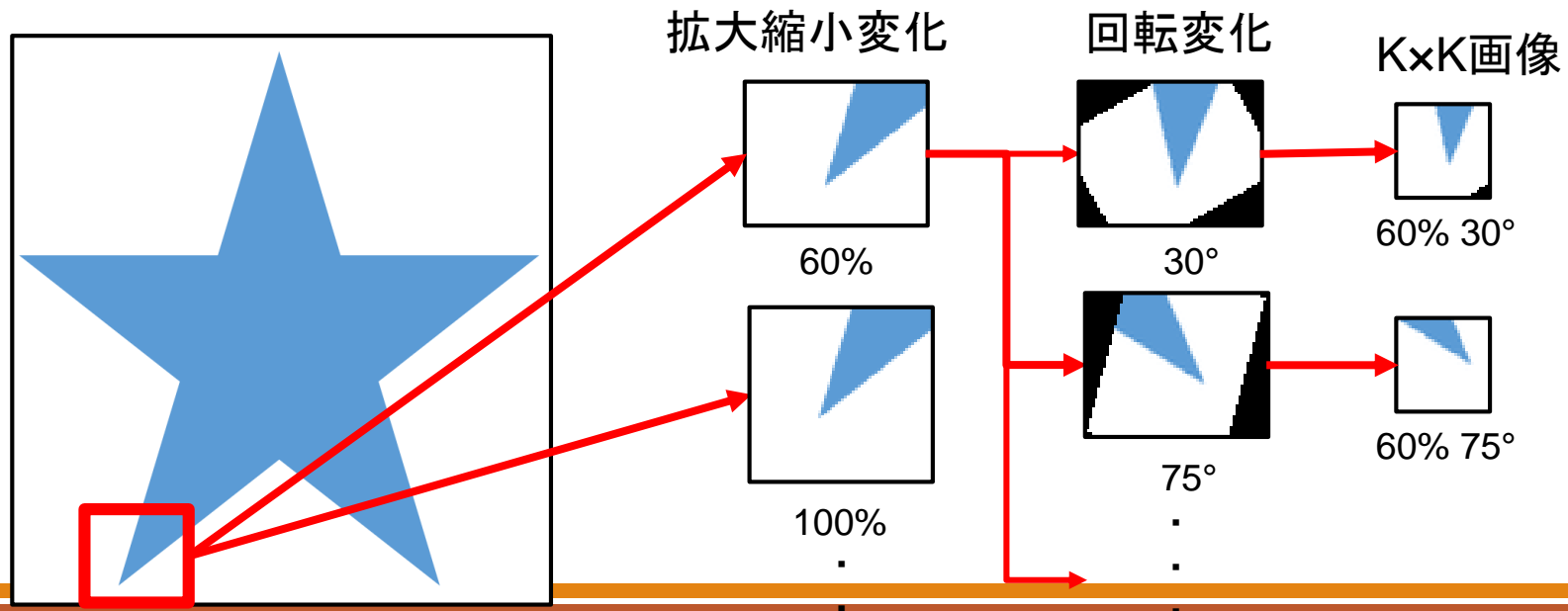


# 特徴量記述

画像1の特徴点を中心として変化（1点で $S \times R$ 通り）

拡大縮小： $S$ 通り  
回転： $R$ 通り

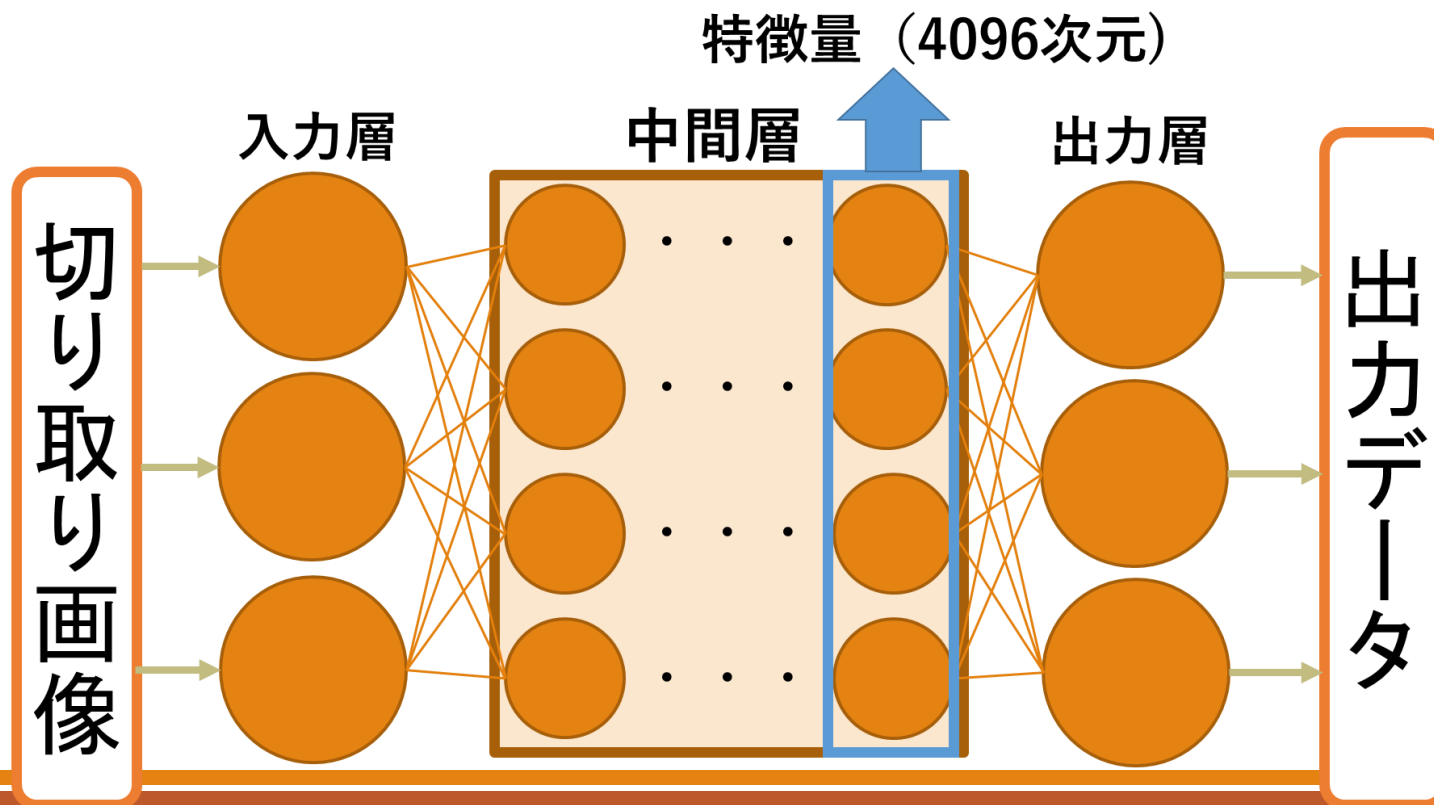
特徴点を中心として $K \times K$ 画素の範囲で切り取る



# 特徴量記述

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

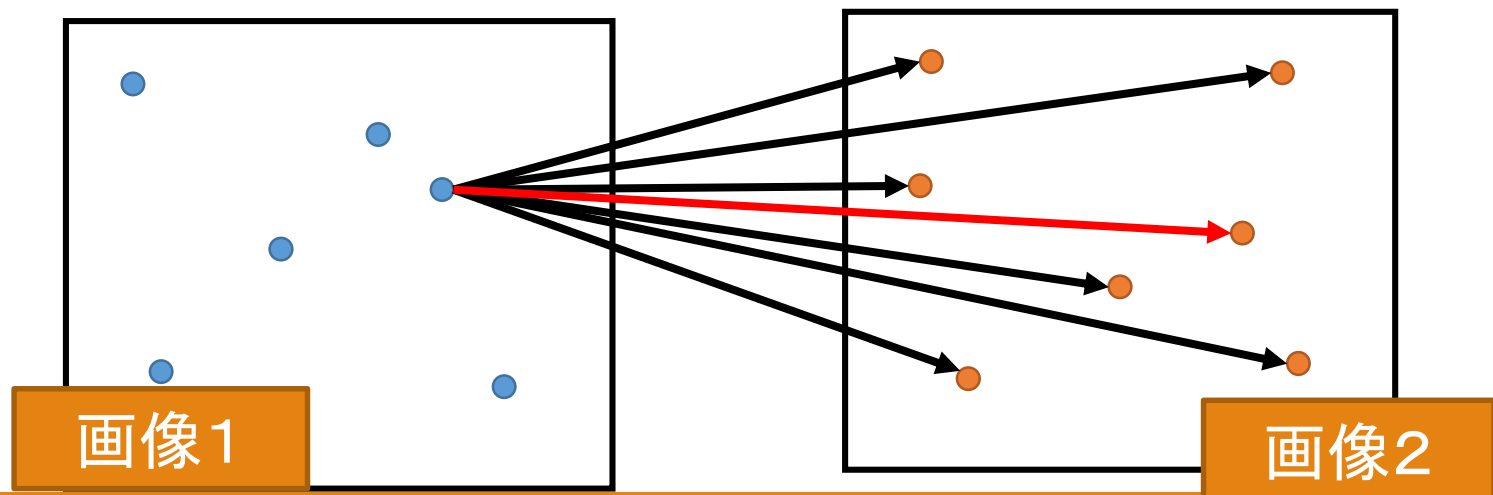


# マッチング

## 総当たりマッチング+クロスチェック

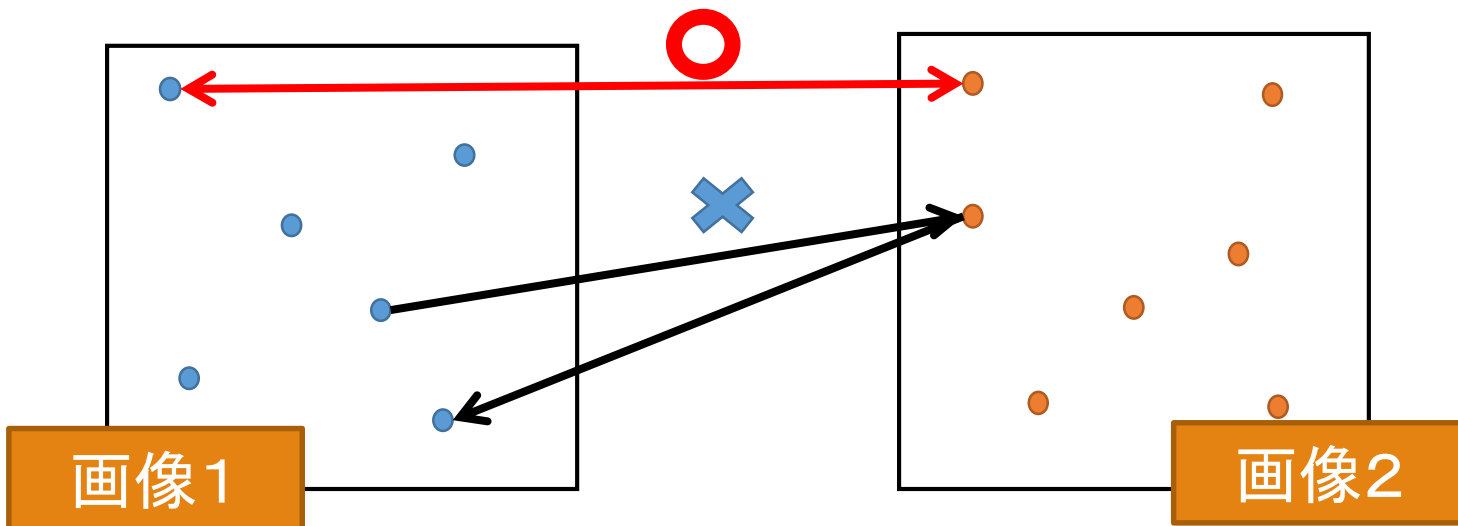
- 総当たりマッチング

- 全ての点を対象に総当たりで特徴点類似度を計算
- 最も類似度が高い特徴点と対応付ける



# マッチング

- クロスチェック
  - 画像1と画像2を入れ替えて総当たりマッチング
  - 同じ特徴点の組が対応したもののみマッチング結果



# 実験

## 提案手法とAKAZEの比較実験

回転、拡大縮小変化の度合いに対する  
マッチング成功率の変化を評価

画像1



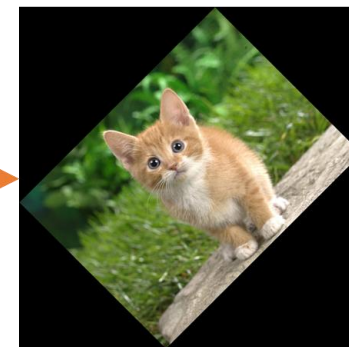
画像2



画像1



画像2



# 使用画像

画像2: 画像1をアフィン変換して変化したもの  
スケール:50%~150% 回転:0°~90°



480 x 360[3]



35° 150%

画像1



90° 110%



5° 70%

画像2



# 実験条件

---

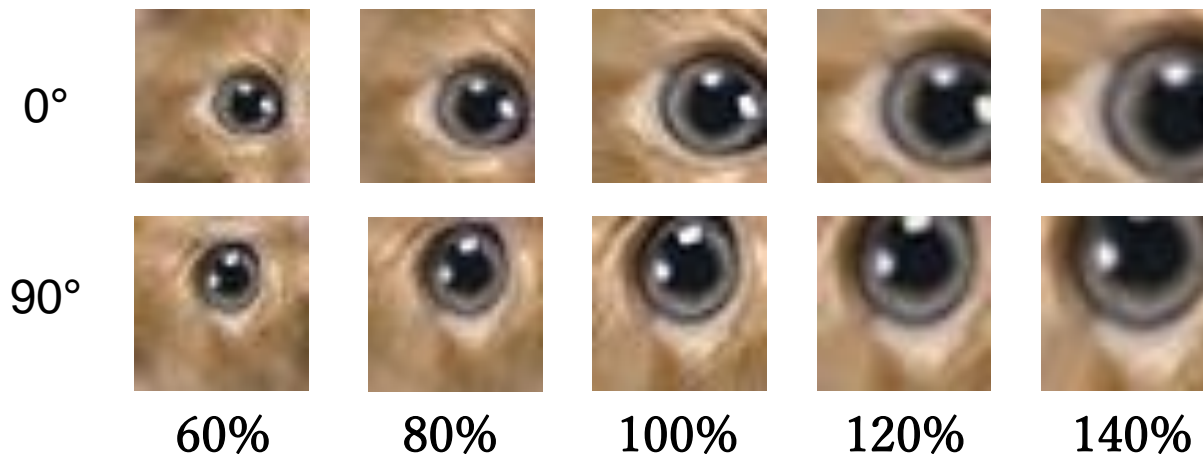
切り取り範囲

$K = 32$ 画素( $32 \times 32$ )

画像1の切り取り時の変化

拡大縮小変化 :  $S = 5$ 通り

回転変化 :  $R = 7$ 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:  $|Q_i - P'_j| < T$

$Q_i$  : 画像1の*i*番目の特徴点座標

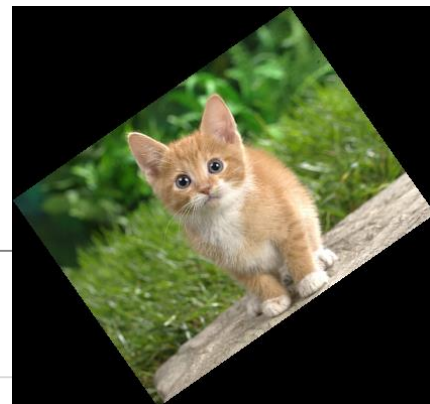
$P_j$  : 画像2の*j*番目の特徴点座標

$P'_j$  : 正解座標( $P_j$ に対して逆変換で算出)

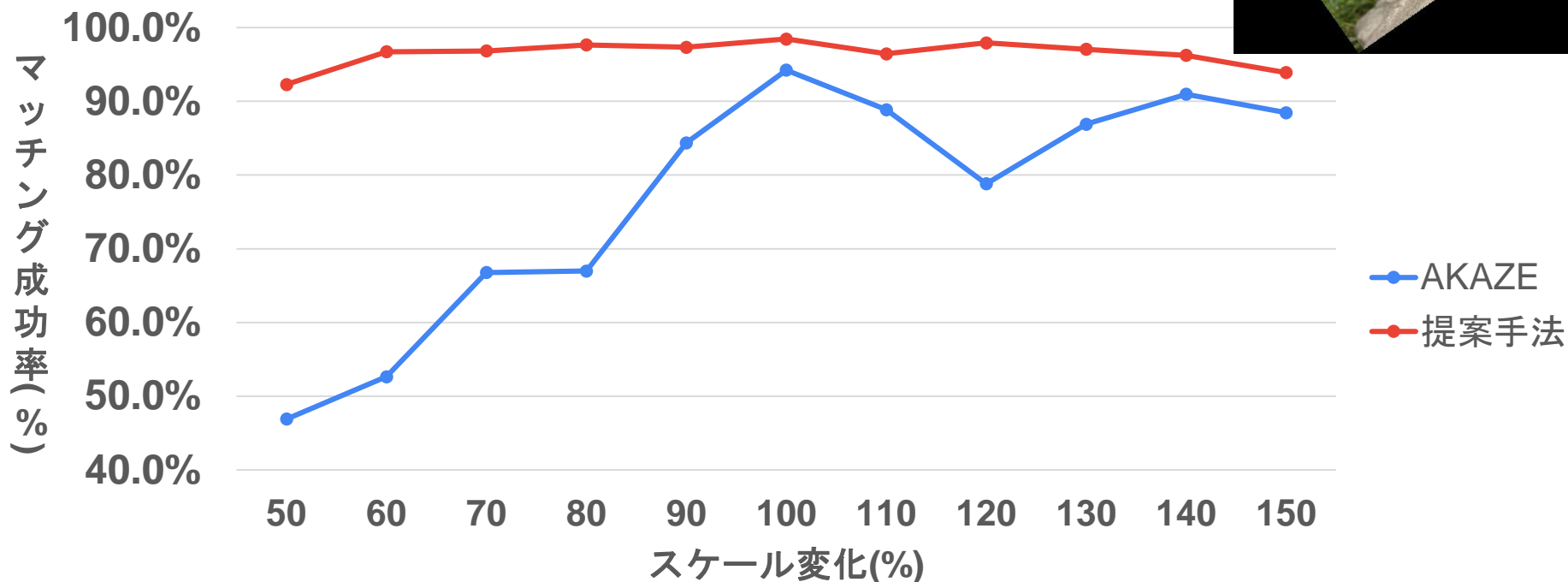
$T$  : 許容座標誤差

( $T = 10$ 画素)

# 実験結果



## 画像2が35°回転のマッチング成功率

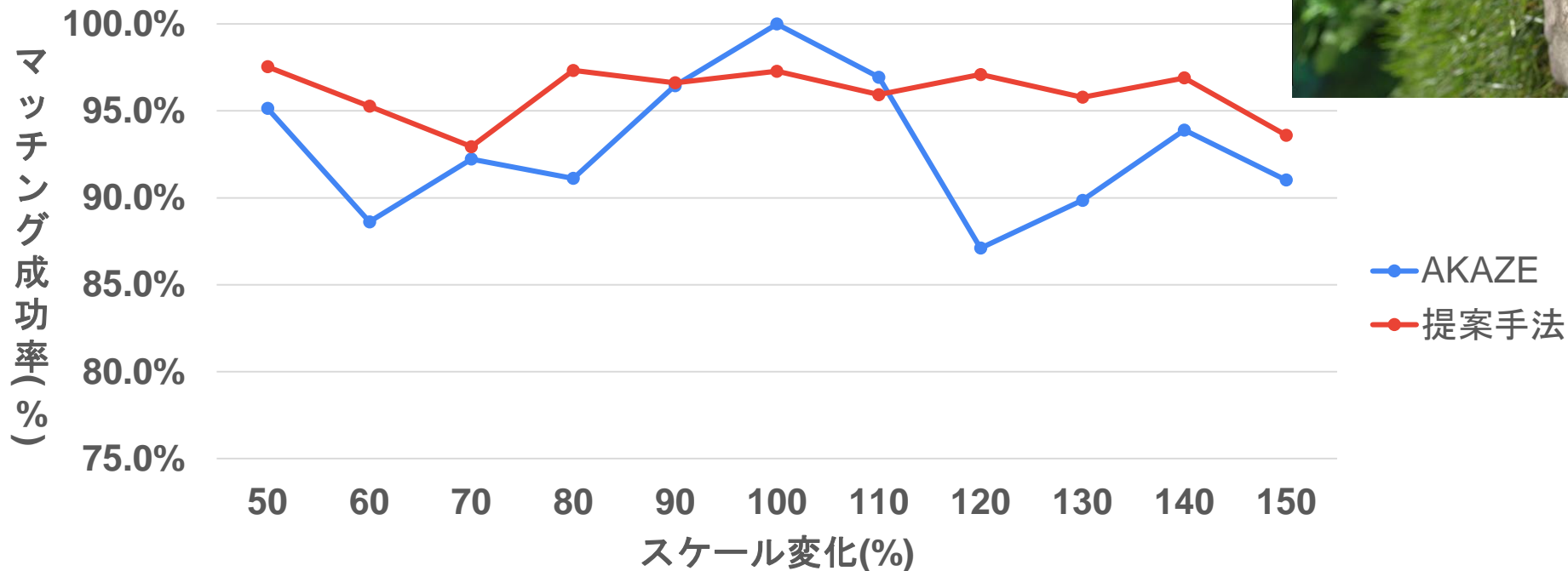


- 画像1の回転変化に含まれていない場合でも90%以上
- AKAZEを上回ることができている

# 実験結果



## 画像2が90°回転のマッチング成功率



90°はAKAZEのマッチング率が高いこともあり、  
スケール100,110%はAKAZEの値を上回ることができていない

# 実験結果

## 提案手法

角度 スケール	0°	5°	15°	35°	45°	90°
50%	98.4	94.6	94.6	92.3	93.2	97.5
60%	97.7	96.9	96.9	96.7	97.2	95.3
70%	93.2	96.4	96.4	96.8	96.8	92.9
80%	98.3	97.7	97.7	97.6	96.3	97.3
90%	95.7	97.0	97.0	97.3	97.1	96.6
100%	100.0	98.1	98.1	98.4	96.2	97.3
110%	96.7	97.3	97.3	96.4	96.4	95.9
120%	97.1	97.3	97.3	97.9	96.9	97.1
130%	97.2	96.9	96.9	97.0	96.7	95.8
140%	97.1	96.2	96.2	96.2	98.7	96.9
150%	94.3	95.0	95.0	93.9	93.5	93.6

## AKAZE

角度 スケール	0°	5°	15°	35°	45°	90°
50%	93.3	80.5	54.5	46.9	45.7	95.1
60%	90.4	75.0	57.4	52.7	51.8	88.6
70%	93.3	80.8	67.7	66.8	66.4	92.2
80%	90.8	81.6	69.5	67.0	69.7	91.1
90%	96.4	90.9	84.1	84.4	86.3	96.5
100%	100.0	93.0	94.4	94.2	93.7	100.0
110%	97.0	84.6	90.6	88.9	87.9	96.9
120%	88.3	90.3	79.9	78.8	80.3	87.1
130%	89.9	93.0	89.4	86.9	85.4	89.9
140%	94.2	91.8	89.9	91.0	88.2	93.9
150%	93.8	54.5	87.7	88.4	87.6	91.0

- 提案手法はすべて90%を上回っている
- ほとんどの場合で提案手法がAKAZEを上回る

# 処理時間

画像2：回転90°、スケール150%

処理時間

提案手法の処理時間内訳

AKAZE	0.579s
提案手法	10430s

	特徴量記述	マッチング
画像1	9633s	247s
画像2	614s	

- 提案手法は174mの時間を必要としている。
- 特に時間がかかるのは特徴量抽出である。

# まとめ

---

- AKAZEに依存することなく回転・拡大縮小不変を実現することができた。
- 提案手法はすべて90%を上回っている。
- ほとんどの場合で提案手法がAKAZEを上回る。

## 今後の課題

処理時間の短縮

