

画像からお札を識別する

所属クラス：C3 学籍番号：1290090 氏名：勝又四海

1.導入

- 背景：画像の中のどこに何円札があるのかが分かると紙幣の落とし物を見つけるのに便利だから。
- 問題設計：画像中からお札がどの種類でどこにあるのか判別できるか。
- 目標：それぞれのお札の特徴点を用いて、お札を判別できるようにする。

2.データ

今回は、以下のサイトから画像入手し、お札を判別していく。

<https://upload.wikimedia.org/wikipedia/commons/thumb/3/32/1000_yen_banknote_%28Series_E%29%2C_obverse.png/640px-1000_yen_banknote_%28Series_E%29%2C_obverse.png>

<https://upload.wikimedia.org/wikipedia/commons/thumb/8/87/2000_yen_banknote_%28Series_D%29%2C_obverse.png/640px-2000_yen_banknote_%28Series_D%29%2C_obverse.png>

<https://upload.wikimedia.org/wikipedia/commons/thumb/d/d7/5000_yen_banknote_%28Series_E%29%2C_obverse.png/640px-5000_yen_banknote_%28Series_E%29%2C_obverse.png>

<https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/10000_yen_banknote_%28Series_E%29%2C_obverse.png/640px-10000_yen_banknote_%28Series_E%29%2C_obverse.png>

<<https://livedoor.blogimg.jp/rocolabo/imgs/d/f/df133997-s.jpg>>

3.解析手法

- 手順 1

まず、お札の画像を読み込み、それぞれの特徴点を検出する。特徴点を検出するために、読み込んだ画像をグレースケール画像に変換する。

最初にお札の画像とテスト用の画像を読み込み表示する。また、テストの画像はサイズが小さく、うまく特徴点を検出出来ないおそれがあるためサイズを2倍にする。

読み込んだお札の画像を合体させた画像は特徴点によりお札を種類別に判定できるかの確認用である。

```
image1000=imread('https://upload.wikimedia.org/wikipedia/commons/thumb/3/32/1000_yen_ba
image2000=imread('https://upload.wikimedia.org/wikipedia/commons/thumb/8/87/2000_yen_ba
image5000=imread('https://upload.wikimedia.org/wikipedia/commons/thumb/d/d7/5000_yen_ba
```

```
image10000=imread('https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/10000_yen_'
image=imread('https://livedoor.blogimg.jp/rocolabo/imgs/d/f/df133997-s.jpg');
imagetest=imresize(image,2);
figure;
imageCombine=montage({image1000,image2000,image5000,image10000});
saveas(imageCombine,'Japanese_Bill.png');
```



```
imageCombine=imread('Japanese_Bill.png');
imshow(imageCombine);
title('Combine Image');
```

Combine Image



```
imshow(imagetest);  
title('test image');
```

test image



次に、読み込んだお札の画像、それらを合体させた画像、テスト用の画像をグレースケール画像に変換する。

```
g_image1000=rgb2gray(image1000);
g_image2000=rgb2gray(image2000);
g_image5000=rgb2gray(image5000);
g_image10000=rgb2gray(image10000);
g_imageCombine=rgb2gray(imageCombine);
g_imagetest=rgb2gray(imagetest);
figure;
montage({g_image1000,g_image2000,g_image5000,g_image10000,g_imageCombine,g_imagetest});
title(' grayscale images');
```

grayscale images



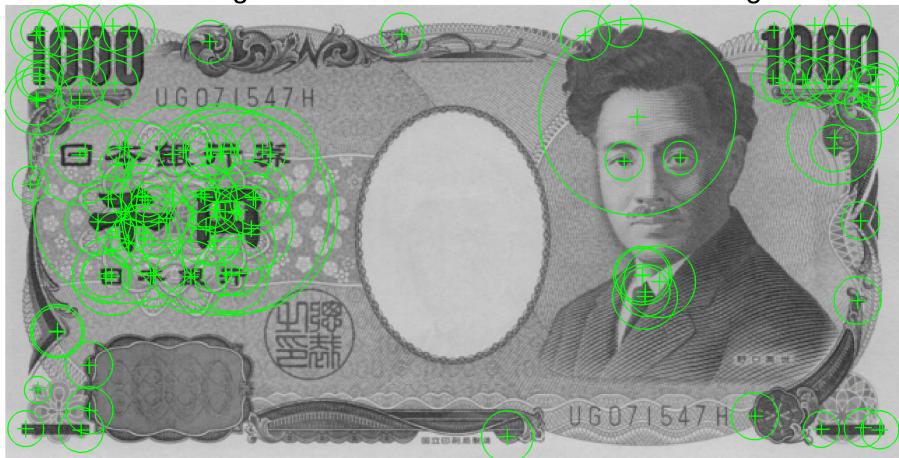
• 手順 2

それぞれの画像の特徴点を検出し表示する。

```
points_1000=detectSURFFeatures(g_image1000);
points_2000=detectSURFFeatures(g_image2000);
points_5000=detectSURFFeatures(g_image5000);
points_10000=detectSURFFeatures(g_image10000);
points_Combine=detectSURFFeatures(g_imageCombine);
points_test=detectSURFFeatures(g_imagetest);

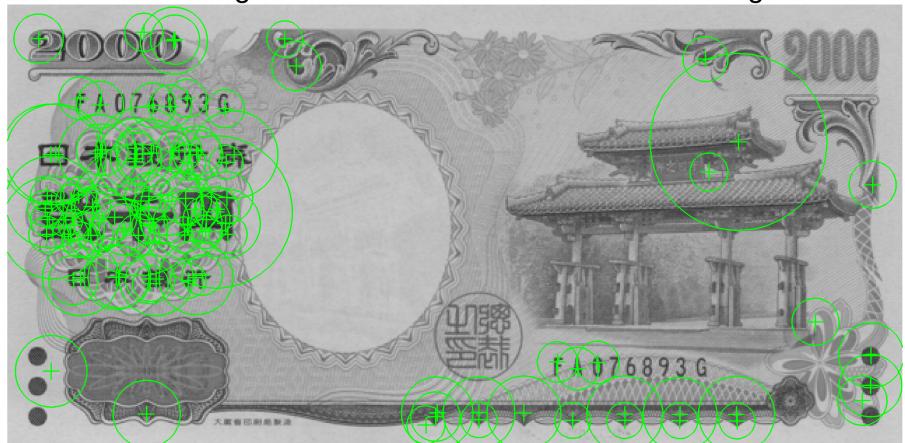
figure;
imshow(g_image1000);
title('100 Strongest Feature Points from 1000Yen Image      ');
hold on;
plot(selectStrongest(points_1000, 100));
```

100 Strongest Feature Points from 1000Yen Image



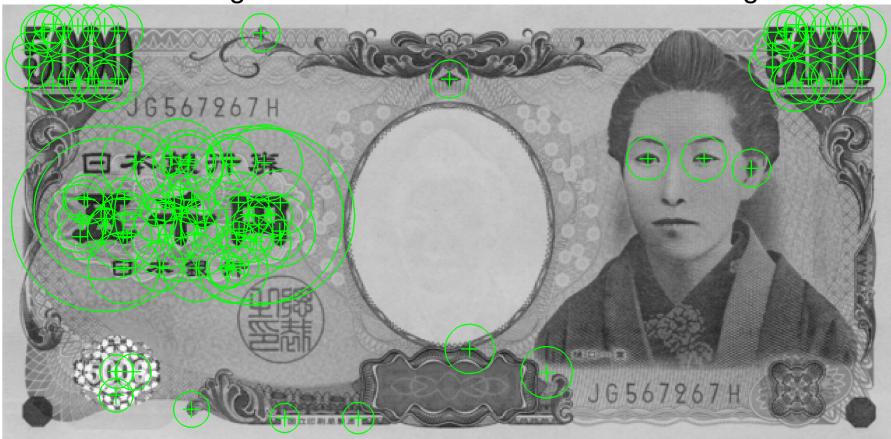
```
figure;
imshow(g_image2000);
title('100 Strongest Feature Points from 2000Yen Image      ');
hold on;
plot(selectStrongest(points_2000, 100));
```

100 Strongest Feature Points from 2000Yen Image



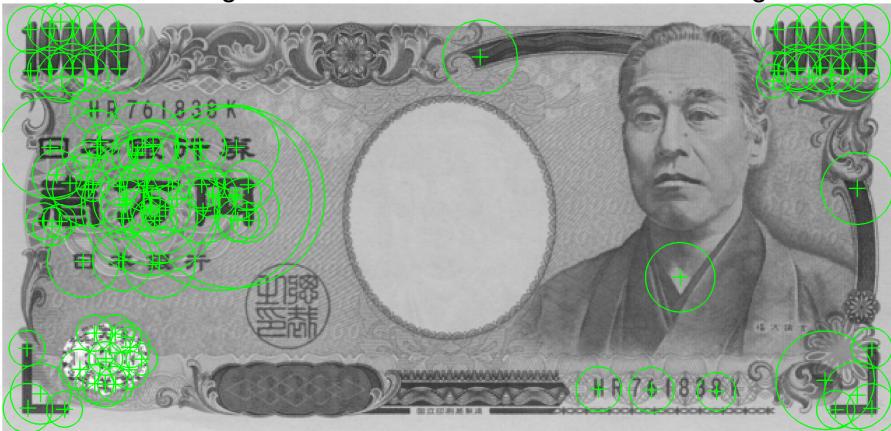
```
figure;
imshow(g_image5000);
title('100 Strongest Feature Points from 5000Yen Image      ');
hold on;
plot(selectStrongest(points_5000, 100));
```

100 Strongest Feature Points from 5000Yen Image



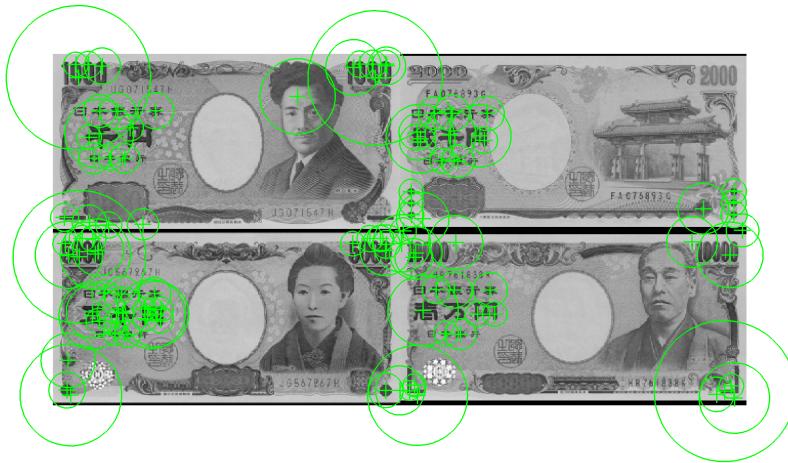
```
figure;
imshow(g_image10000);
title('100 Strongest Feature Points from 10000Yen Image    ');
hold on;
plot(selectStrongest(points_10000, 100));
```

100 Strongest Feature Points from 10000Yen Image



```
figure;
imshow(g_imageCombine);
title('100 Strongest Feature Points from Combine Image      ');
hold on;
plot(selectStrongest(points_Combine, 100));
```

100 Strongest Feature Points from Combine Image



```
figure;
imshow(g_imagetest);
title('100 Strongest Feature Points from test Image      ');
hold on;
plot(selectStrongest(points_test, 100));
```

100 Strongest Feature Points from test Image



• 手順3

すべてのイメージ内の関心点における特徴記述子を抽出する。

```
[features_1000, points_1000] = extractFeatures(g_image1000, points_1000);
[features_2000, points_2000] = extractFeatures(g_image2000, points_2000);
[features_5000, points_5000] = extractFeatures(g_image5000, points_5000);
[features_10000, points_10000] = extractFeatures(g_image10000, points_10000);
[features_Combine, points_Combine] = extractFeatures(g_imageCombine, points_Combine);
[features_test, points_test] = extractFeatures(g_imagetest, points_test);
```

• 手順4

特徴記述子を使用して特徴をマッチングし、マッチであると推定される特徴を表示する。

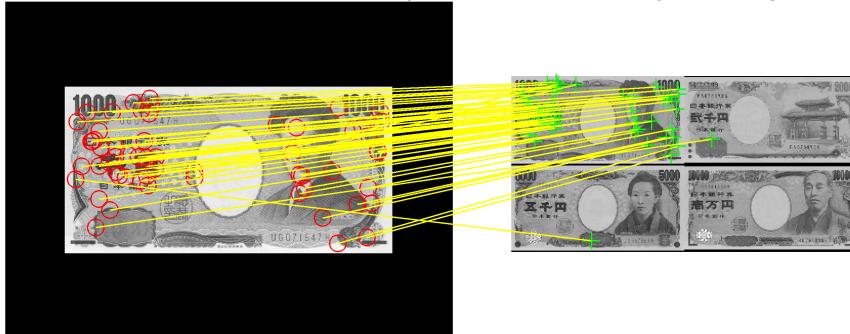
最初に 1000 円、2000 円、5000 円、10000 円それぞれの画像とそれらを結合した画像のマッチングをする。

```
pairs_1000_Combine = matchFeatures(features_1000, features_Combine);
pairs_2000_Combine = matchFeatures(features_2000, features_Combine);
pairs_5000_Combine = matchFeatures(features_5000, features_Combine);
pairs_10000_Combine = matchFeatures(features_10000, features_Combine);

matched_1000Points = points_1000(pairs_1000_Combine(:, 1), :);
matched_1000_CombinePoints = points_Combine(pairs_1000_Combine(:, 2), :);
figure;
```

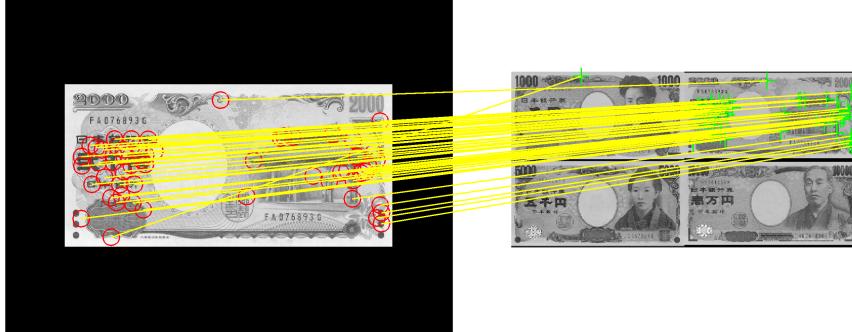
```
showMatchedFeatures(g_image1000, g_imageCombine, matched_1000Points, matched_1000_Combi  
title('Estimated match between 1000 yen and Combine (Including Outliers)');
```

Estimated match between 1000 yen and Combine (Including Outliers)



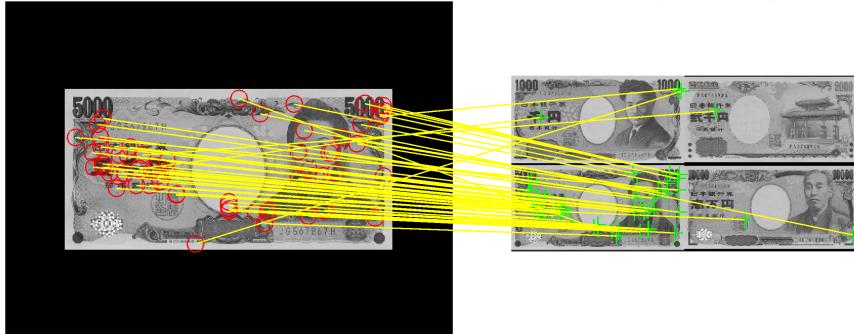
```
matched_2000Points = points_2000(pairs_2000_Combine(:, 1), :);  
matched_2000_CombinePoints = points_Combine(pairs_2000_Combine(:, 2), :);  
figure;  
showMatchedFeatures(g_image2000, g_imageCombine, matched_2000Points, matched_2000_Combi  
title('Estimated match between 2000 yen and Combine (Including Outliers)');
```

Estimated match between 2000 yen and Combine (Including Outliers)



```
matched_5000Points = points_5000(pairs_5000_Combine(:, 1), :);
matched_5000_CombinePoints = points_Combine(pairs_5000_Combine(:, 2), :);
figure;
showMatchedFeatures(g_image5000, g_imageCombine, matched_5000Points, matched_5000_CombinePoints);
title('Estimated match between 5000 yen and Combine (Including Outliers)');
```

Estimated match between 5000 yen and Combine (Including Outliers)



```
matched_10000Points = points_10000(pairs_10000_Combine(:, 1), :);
matched_10000_CombinePoints = points_Combine(pairs_10000_Combine(:, 2), :);
figure;
showMatchedFeatures(g_image10000, g_imageCombine, matched_10000Points, matched_10000_C
title('Estimated match between 10000 yen and Combine (Including Outliers)    ');
```

Estimated match between 10000 yen and Combine (Including Outliers)



- 手順 5

手順 4 から外れ値を排除しながら、マッチした点を関連付ける変換を計算する。この変換によってシーン内のオブジェクトの位置を決定し、イメージ内の対象オブジェクトを多角形で囲む。

まず、外れ値を排除し、マッチした点を関連付ける。

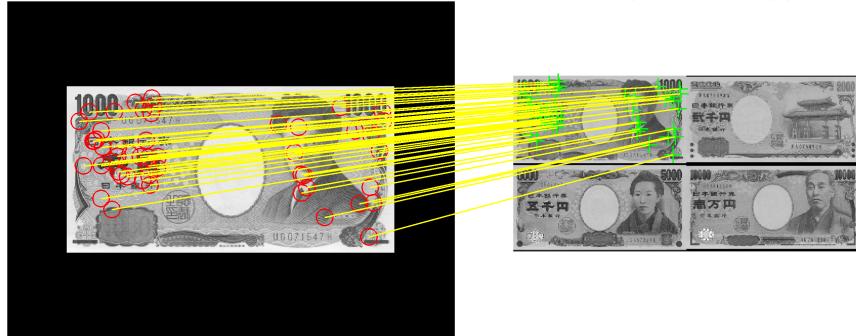
1000 円

```
[tform, inlierIdx] = estimateGeometricTransform2D(matched_1000Points, matched_1000_CombinePoints);
inlier_1000Points = matched_1000Points(inlierIdx, :);
inlier_1000_CombinePoints = matched_1000_CombinePoints(inlierIdx, :);
```

外れ値を排除したマッチング画像を表示する。

```
figure;
showMatchedFeatures(g_image1000, g_imageCombine, inlier_1000Points, inlier_1000_CombinePoints,
title('1000 Yen and Combine Matched Points (Inliers Only)'));
```

1000 Yen and Combine Matched Points (Inliers Only)



イメージの境界多角形を取得する。この多角形(ボックス)をターゲット イメージの座標系に変換し、イメージ内のオブジェクトの位置を示す。

1000 円は黄色のボックスで表す。

```
Polygon_1000 = [1, 1;size(g_image1000, 2), 1;size(g_image1000, 2), size(g_image1000, 1);  
newCombine1000Polygon = transformPointsForward(tform, Polygon_1000);  
figure;  
imshow(g_imageCombine);  
hold on;  
line(newCombine1000Polygon(:, 1), newCombine1000Polygon(:, 2), 'Color', 'y');  
title('Box where 1000 yen was detected ');
```

Box where 1000 yen was detected

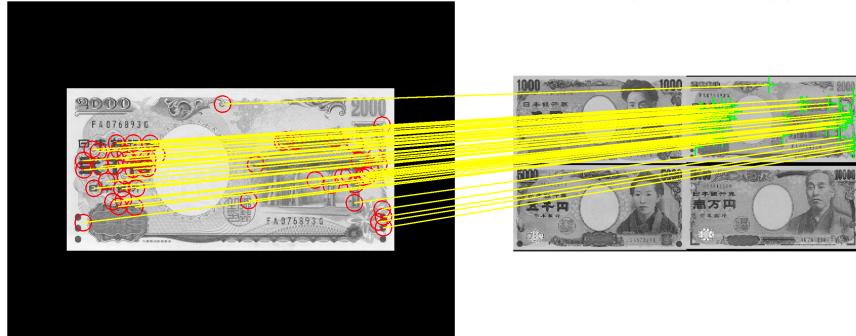


2000 円、5000 円、10000 円にも同様の操作をする。2000 円は青、5000 円は緑、10000 円は赤のボックスで囲まれる。

2000 円

```
[tform, inlierIdx] = estimateGeometricTransform2D(matched_2000Points, matched_2000_CombinePoints);
inlier_2000Points = matched_2000Points(inlierIdx, :);
inlier_2000_CombinePoints = matched_2000_CombinePoints(inlierIdx, :);
figure;
showMatchedFeatures(g_image2000, g_imageCombine, inlier_2000Points, inlier_2000_CombinePoints);
title('2000 Yen and Combine Matched Points (Inliers Only)');
```

2000 Yen and Combine Matched Points (Inliers Only)



```
Polygon_2000 = [1, 1:size(g_image2000, 2), 1:size(g_image2000, 2), size(g_image2000, 1)]
newCombine2000Polygon = transformPointsForward(tform, Polygon_2000);
figure;
imshow(g_imageCombine);
hold on;
line(newCombine2000Polygon(:, 1), newCombine2000Polygon(:, 2), 'Color', 'b');
title('Box where 2000 yen was detected '');
```

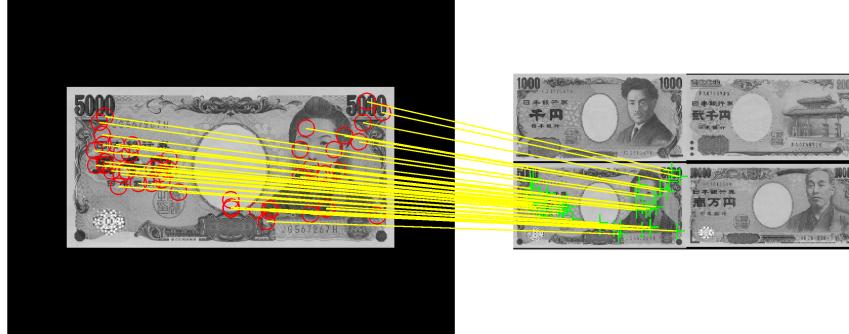
Box where 2000 yen was detected



5000 円

```
[tform, inlierIdx] = estimateGeometricTransform2D(matched_5000Points, matched_5000_CombinePoints);
inlier_5000Points = matched_5000Points(inlierIdx, :);
inlier_5000_CombinePoints = matched_5000_CombinePoints(inlierIdx, :);
figure;
showMatchedFeatures(g_image5000, g_imageCombine, inlier_5000Points, inlier_5000_CombinePoints);
title('5000 Yen and Combine Matched Points (Inliers Only)');
```

5000 Yen and Combine Matched Points (Inliers Only)



```
Polygon_5000 = [1, 1:size(g_image5000, 2), 1:size(g_image5000, 2), size(g_image5000, 1)]
newCombine5000Polygon = transformPointsForward(tform, Polygon_5000);
figure;
imshow(g_imageCombine);
hold on;
line(newCombine5000Polygon(:, 1), newCombine5000Polygon(:, 2), 'Color', 'g');
title('Box where 5000 yen was detected '');
```

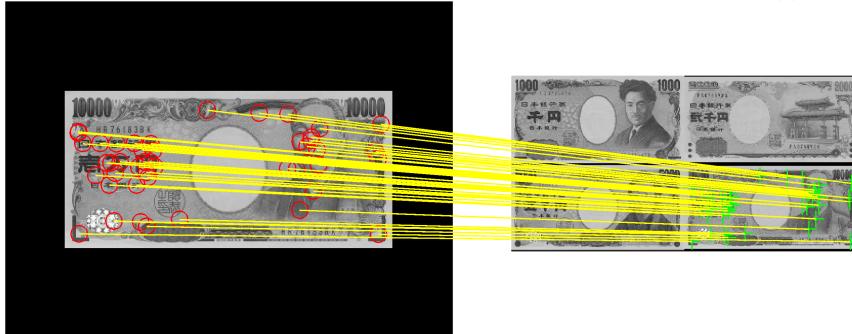
Box where 5000 yen was detected



10000 円

```
[tform, inlierIdx] = estimateGeometricTransform2D(matched_10000Points, matched_10000_Co  
inlier_10000Points = matched_10000Points(inlierIdx, :);  
inlier_10000_CombinePoints = matched_10000_CombinePoints(inlierIdx, :);  
figure;  
showMatchedFeatures(g_image10000, g_imageCombine, inlier_10000Points, inlier_10000_CombinePoints,  
title('10000 Yen and Combine Matched Points (Inliers Only)      ');
```

10000 Yen and Combine Matched Points (Inliers Only)



```
Polygon_10000 = [1, 1:size(g_image10000, 2), 1:size(g_image10000, 2), size(g_image10000, 2)];
newCombine10000Polygon = transformPointsForward(tform, Polygon_10000);
figure;
imshow(g_imageCombine);
hold on;
line(newCombine10000Polygon(:, 1), newCombine10000Polygon(:, 2), 'Color', 'r');
title('Box where 2000 yen was detected');
```

Box where 2000 yen was detected



- テスト

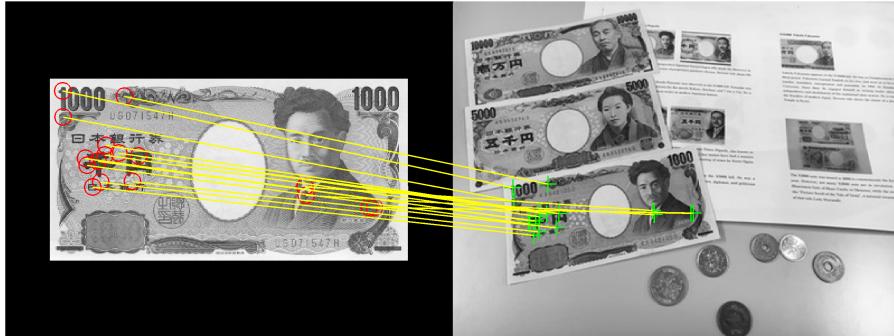
結合した画像だけではお札を種類ごとに判別できているのか判断するには不十分であると考えたため、別の画像をテスト用に用意した。以上の手順をお札の画像とテストの画像のペアでも行う。なお、テスト画像には2000円札がないが、どのような結果になるか確認するために2000円札とも行う。

1000 円

```
pairs_1000_test = matchFeatures(features_1000, features_test);
pairs_2000_test = matchFeatures(features_2000, features_test);
pairs_5000_test = matchFeatures(features_5000, features_test);
pairs_10000_test = matchFeatures(features_10000, features_test);

matched_1000Points = points_1000(pairs_1000_test(:, 1), :);
matched_1000_testPoints = points_test(pairs_1000_test(:, 2), :);
[tform, inlierIdx] = estimateGeometricTransform2D(matched_1000Points, matched_1000_testPoints);
inlier_1000Points = matched_1000Points(inlierIdx, :);
inlier_1000_testPoints = matched_1000_testPoints(inlierIdx, :);
figure;
showMatchedFeatures(g_image1000, g_imagetest, inlier_1000Points, inlier_1000_testPoints);
title('1000 Yen and test Matched Points (Inliers Only)      '');
```

1000 Yen and test Matched Points (Inliers Only)



```
Polygon_1000 = [1, 1:size(g_imagine1000, 2), 1:size(g_imagine1000, 2), size(g_imagine1000, 1)
newtest1000Polygon = transformPointsForward(tform, Polygon_1000);
figure;
imshow(g_imagetest);
hold on;
line(newtest1000Polygon(:, 1), newtest1000Polygon(:, 2), 'Color', 'y');
title('Box where 1000 yen was detected in test');
```

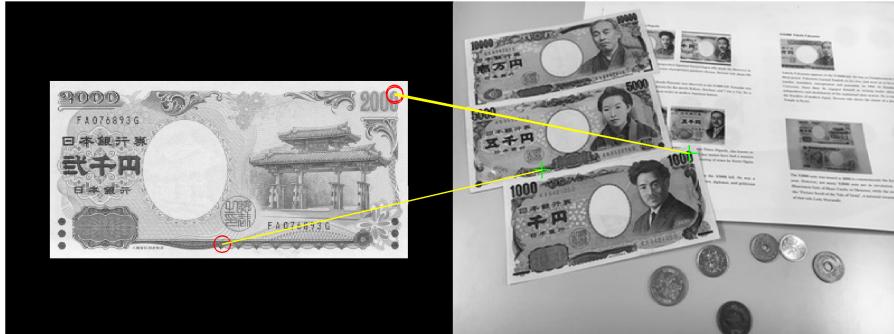
Box where 1000 yen was detected in test



2000 円

```
matched_2000Points = points_2000(pairs_2000_test(:, 1), :);
matched_2000_testPoints = points_test(pairs_2000_test(:, 2), :);
[tform, inlierIdx] = estimateGeometricTransform2D(matched_2000Points, matched_2000_testPoints);
inlier_2000Points = matched_2000Points(inlierIdx, :);
inlier_2000_testPoints = matched_2000_testPoints(inlierIdx, :);
figure;
showMatchedFeatures(g_image2000, g_imagetest, inlier_2000Points, inlier_2000_testPoints);
title('2000 Yen and test Matched Points (Inliers Only)      ')
```

2000 Yen and test Matched Points (Inliers Only)



```
Polygon_2000 = [1, 1:size(g_image2000, 2), 1:size(g_image2000, 2), size(g_image2000, 1)
newtest2000Polygon = transformPointsForward(tform, Polygon_2000);
figure;
imshow(g_imagetest);
hold on;
line(newtest2000Polygon(:, 1), newtest2000Polygon(:, 2), 'Color', 'b');
title('Box where 2000 yen was detected in test    ');
```

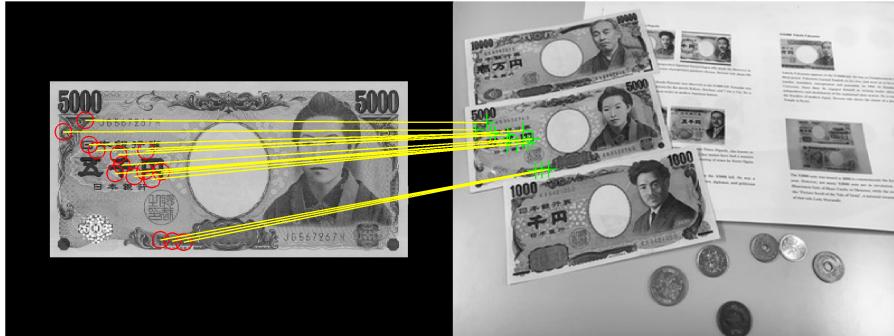
Box where 2000 yen was detected in test



5000 円

```
matched_5000Points = points_5000(pairs_5000_test(:, 1), :);
matched_5000_testPoints = points_test(pairs_5000_test(:, 2), :);
[tform, inlierIdx] = estimateGeometricTransform2D(matched_5000Points, matched_5000_testPoints);
inlier_5000Points = matched_5000Points(inlierIdx, :);
inlier_5000_testPoints = matched_5000_testPoints(inlierIdx, :);
figure;
showMatchedFeatures(g_image5000, g_imagetest, inlier_5000Points, inlier_5000_testPoints);
title('5000 Yen and test Matched Points (Inliers Only)      ')
```

5000 Yen and test Matched Points (Inliers Only)



```
Polygon_5000 = [1, 1:size(g_imagetest, 2), 1:size(g_imagetest, 2), size(g_imagetest, 1)]  
newtest5000Polygon = transformPointsForward(tform, Polygon_5000);  
figure;  
imshow(g_imagetest);  
hold on;  
line(newtest5000Polygon(:, 1), newtest5000Polygon(:, 2), 'Color', 'g');  
title('Box where 5000 yen was detected in test')
```

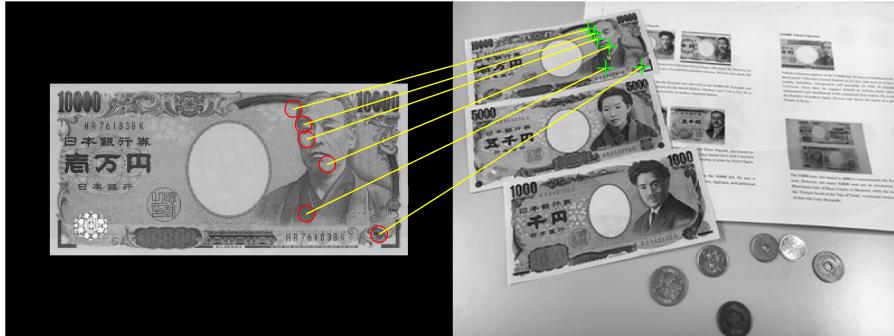
Box where 5000 yen was detected in test



10000 円

```
matched_10000Points = points_10000(pairs_10000_test(:, 1), :);
matched_10000_testPoints = points_test(pairs_10000_test(:, 2), :);
[tform, inlierIdx] = estimateGeometricTransform2D(matched_10000Points, matched_10000_te
inlier_10000Points = matched_10000Points(inlierIdx, :);
inlier_10000_testPoints = matched_10000_testPoints(inlierIdx, :);
figure;
showMatchedFeatures(g_image10000, g_imagetest, inlier_10000Points, inlier_10000_testPo
title('10000 Yen and test Matched Points (Inliers Only)    ');
```

10000 Yen and test Matched Points (Inliers Only)



```
Polygon_10000 = [1, 1:size(g_imaget10000, 2), 1:size(g_imaget10000, 2), size(g_imaget10000, 2)];
newtest10000Polygon = transformPointsForward(tform, Polygon_10000);
figure;
imshow(g_imagetest);
hold on;
line(newtest10000Polygon(:, 1), newtest10000Polygon(:, 2), 'Color', 'r');
title('Box where 10000 yen was detected in test');
```

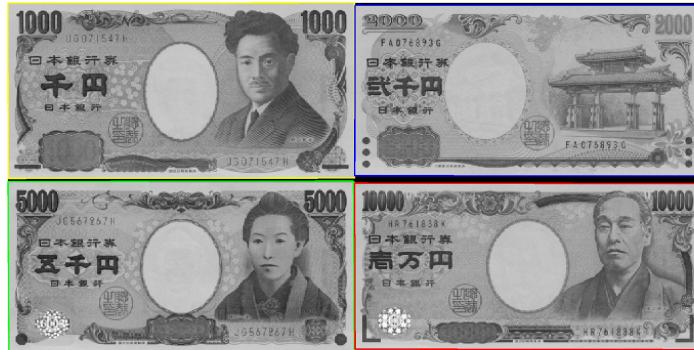
Box where 10000 yen was detected in test



4.結果

```
figure;
imshow(g_imageCombine);
hold on;
line(newCombine1000Polygon(:, 1), newCombine1000Polygon(:, 2), 'Color', 'y');
line(newCombine2000Polygon(:, 1), newCombine2000Polygon(:, 2), 'Color', 'b');
line(newCombine5000Polygon(:, 1), newCombine5000Polygon(:, 2), 'Color', 'g');
line(newCombine10000Polygon(:, 1), newCombine10000Polygon(:, 2), 'Color', 'r');
title('Identified Bills');
```

Identified Bills



3の手法で結合した画像からお札を判別したものを一つにまとめると上のようにになり、すべての種類のお札を検出、判別できたとわかる。

```
figure;
imshow(g_imagetest);
hold on;
line(newtest1000Polygon(:, 1), newtest1000Polygon(:, 2), 'Color', 'y');
line(newtest2000Polygon(:, 1), newtest2000Polygon(:, 2), 'Color', 'b');
line(newtest5000Polygon(:, 1), newtest5000Polygon(:, 2), 'Color', 'g');
line(newtest10000Polygon(:, 1), newtest10000Polygon(:, 2), 'Color', 'r');
title('result of test');
```

result of test



上のテスト用の画像から 1000 円、5000 円、10000 円は種類、位置ともに検出、判別できた。

また、2000 円は検出できていないが、テスト用画像には元より 2000 円がないので正常に機能している。

5.議論

今回の検証では、日本の紙幣の表面の特徴点から紙幣を種類別に判別することができた。しかし、紙幣の裏の特徴点の検出は行っていないので表面と同じようにうまくいくかは分からぬ。また、今回はある程度紙幣の全体像が見える画像で判別したので、どの程度紙幣が映っていれば検出できるのかも不明である。従って、本検証の手法はあくまでも紙幣の表面の全体像が見える画像でのみ有効な判別方法であると言える。

さらに、今回画像を集める上で気づいたことだが、紙幣の画像はあまり出回っておらず、画像が大量に必要になるような手法は紙幣に対してはあまり有効でないよう感じた。

6.結論

大量にある画像から紙幣を発見判別するのは難しい。しかし、今回の検証では MATLAB によりオブジェクトの特徴点から画像の中の紙幣を発見判別することができた。紙幣の表面の全体像が見える画像であるという条件は付くものの、検証に使った画像以外でも判別は可能であると思われる。しかし、画像に対して紙幣がある程度の大きさでないと特徴点をうまく検出できない可能性があるため、どの画像を使うかの選択が重要になってくる。

7.参考文献

特徴点のマッチングを使用した要素の多いシーンでのオブジェクト検出

<<https://jp.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-matching.html>>