

関西学院大学
理工学部人間システム工学科
米岡主税

性能評価実験

1. はじめに

1. 1. 最終目標

コンピューターがボードゲームの「イリュージョン」(1. 2で解説)で遊べるように、画像処理を用いた領域抽出を行う。

1. 2. イリュージョンについて

イリュージョンは色カード(図1)と四色(赤、青、黄、緑)の矢印カード(図2)を用いるボードゲームである。ゲームの流れは以下のとおりである。

- (1) 矢印カードを1枚引き、場に出す。このとき、矢印の色を指定色と呼ぶ。
- (2) 各プレイヤーは順番に色カードを1枚引き、絵柄に含まれる指定色の面積の割合を予想する。そして、場に面積の割合が小さい順となるように色カードを並べる。
- (3) 他のプレイヤーが「ダウト」(正しい順番ではない)と宣言するまで(2)を繰り返す。
- (4) (3)の宣言が正しければカードを置いた人に、間違っていれば宣言したプレイヤーに矢印カードを渡して、(1)から再開する。
- (5) 矢印カードを2枚持つプレイヤーが現れたら、そのプレイヤーの負けとしてゲーム終了

当実験では(1)(2)の処理を実現することを目指す。また、実験の精度は実際の面積の割合(色カード裏面に記載)との誤差をいかに小さくできるかで判定する。



図 1. 矢印カード



図 2. 色カード (左:裏面、右:表面)

2. 実験の目標

- (1) 矢印カードの画素値を取得する
- (2) 色カードの矢印カードと同じ色領域を抽出し、面積を求める
- (3) 色カードを領域の面積順に並べ、動画として保存する

3. 処理手順の設計

3.1. 処理手順

- (1) 矢印カードの画素値を取得する

- | |
|---|
| <ol style="list-style-type: none">1. 背景画像を読み込み、濃淡画像に変換2. 矢印カードを置き、カメラ画像を濃淡画像に変換し、背景画像との差分を取る3. 抽出された領域のうち、非常に大きなものの重心における画素値 (HSV) を求める |
|---|

- (2) 色カードの矢印カードと同じ色領域を抽出する

- | |
|--|
| <ol style="list-style-type: none">1. 色カードを置き、背景差分及び矢印の色の領域を抽出する2. クロージング処理とオープニング処理を行う3. フレーム間差分が十分小さければ、領域の面積を求める4. 1〜3 を十回繰り返し、領域の面積が占める割合の平均値を配列 1 に格納5. クロマキー合成を行い、画像を配列 2 に格納 |
|--|

- (3) 色カードを領域の面積順に並べ、動画として保存する

- | |
|---|
| <ol style="list-style-type: none">1. 面積の小さい順になるように配列 1 に基づいてソートを行う2. 配列 2 から画像を取り出し、ビデオに保存する |
|---|

3.2. 処理のねらい

- (1) 矢印カードの画素値を取得する

濃淡画像の背景差分をとり、十分に大きな領域を取り出してやれば、小さな差分は無視されて矢印カードのみを抽出できる。そして、カードの中央に矢印が描かれているため、重心を取ることで矢印上の座標を取得できる。そのため、その点での画素値を調べてやれば、矢印の色が抽出できる。
--

- (2) 色カードの矢印カードと同じ色領域を抽出する

まず、背景差分を取ることで(1)と同様に色カードのみを抽出できる。そして、(1)で調べた画素値と同じ色領域を抽出してやれば、カメラ画像の色領域が抽出される。これらから得られたマスク画像の AND を取ることで、色カード上の矢印と同じ色の領域が抽出できる。この際にオープニング処理とクロージング処理で、切り出された領域にノイズがあれば取り除くことができる。その後、処理を繰り返して平均値をとることで誤差を小さくするのだが、平均値をとる画像間に大きな差がないように、フレーム間差分を取ることで画像間の差が大きければループを抜ける。

- (3) 色カードを領域の面積順に並べ、動画として保存する

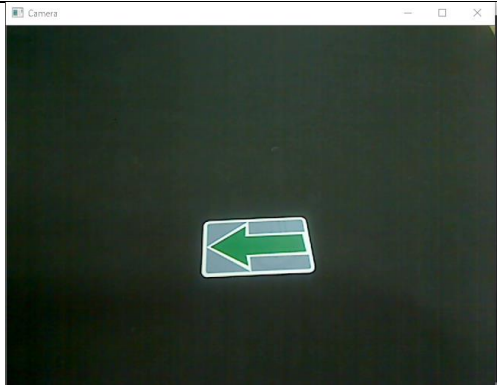
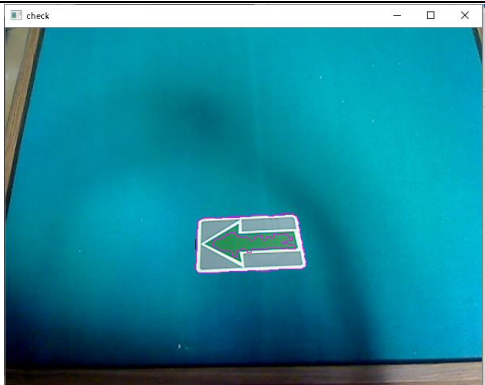
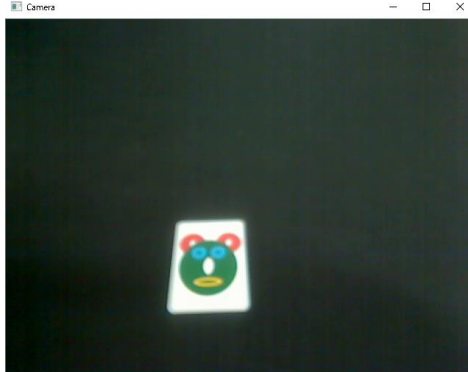
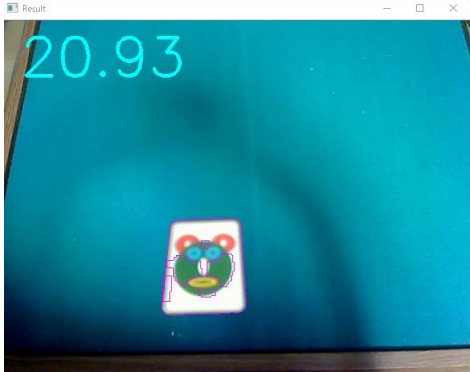
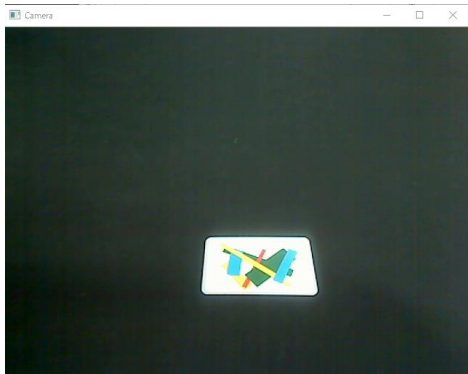
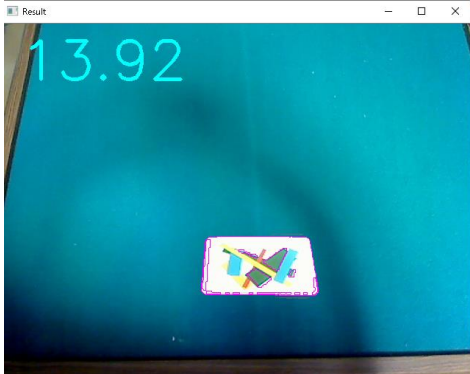
(2)の処理で得られた配列をソートすることで面積順にカードの画像を並べることができる。そして、画像をビデオに格納することで並び替えた結果を確認できる。

4. 実験方法と実験結果

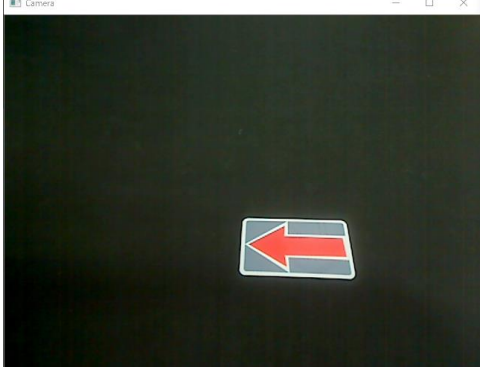
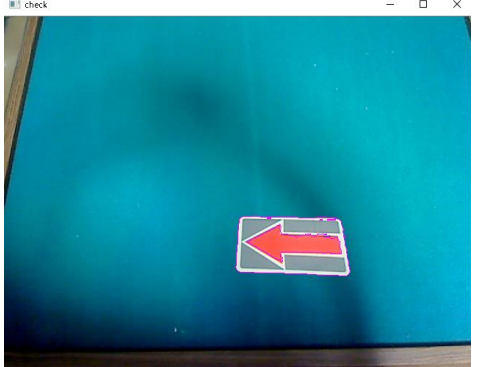
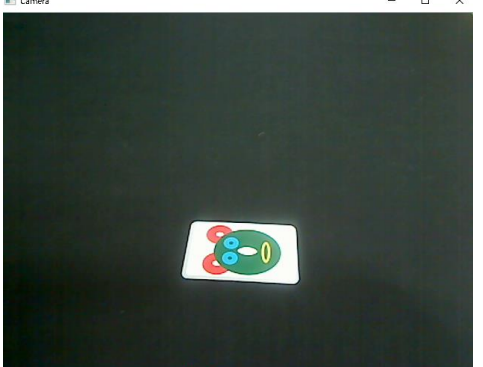
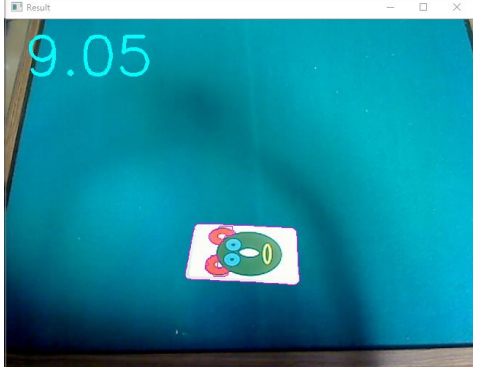
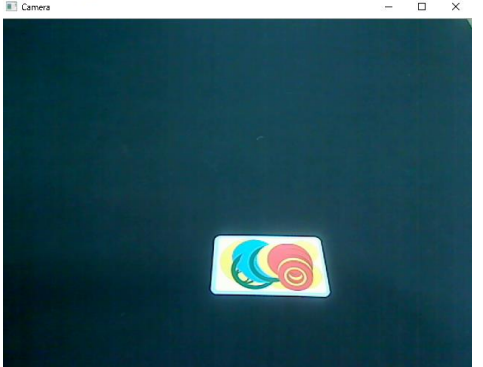
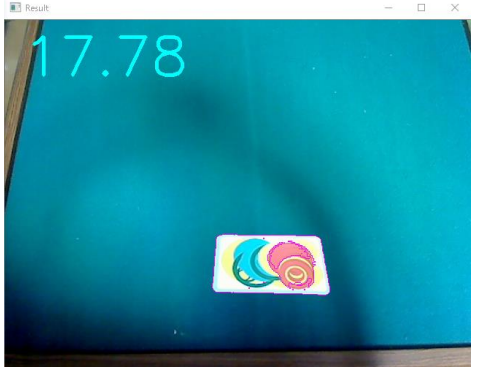
4.1. 入力画像

実際は複数の静止画像の入力に対して動画像が出力されるが、簡略化のために各入力に対して静止画像を出力として提示している。




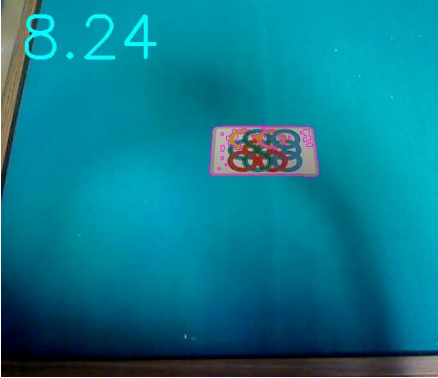
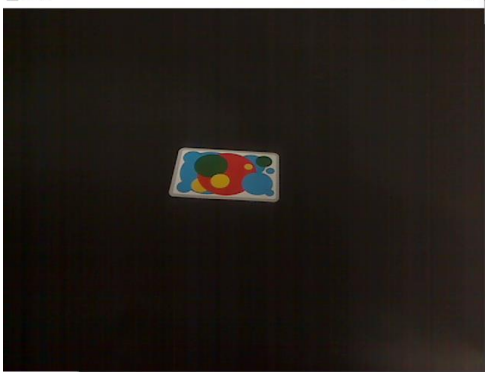
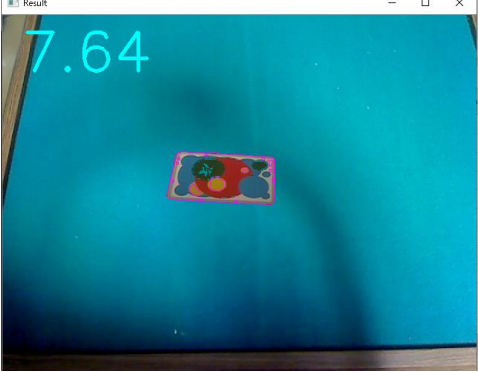
(1) 矢印が緑色のとき

真の割合	入力画像	出力画像
矢印		
30%		
14%		


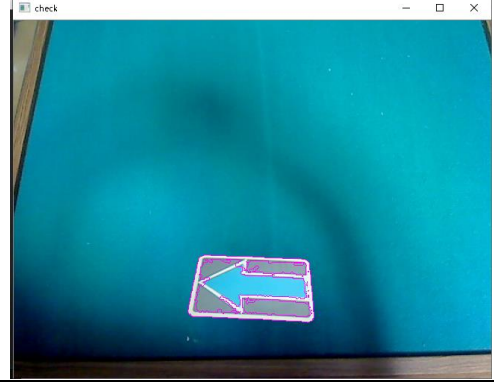



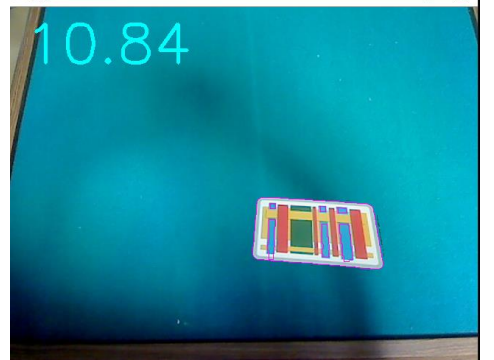
(2) 矢印が赤色のとき

真の割合	入力画像	出力画像
矢印		
10%		
25%		

(3) 矢印が黄色のとき

真の割合	入力画像	出力画像
矢印		
10%		
8%		

(4) 矢印が青色のとき

真の割合	入力画像	出力画像
矢印		
11%		
15%		



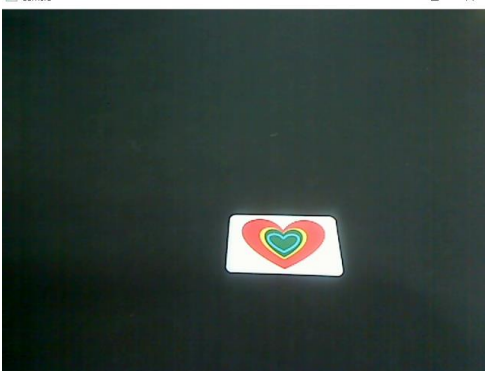
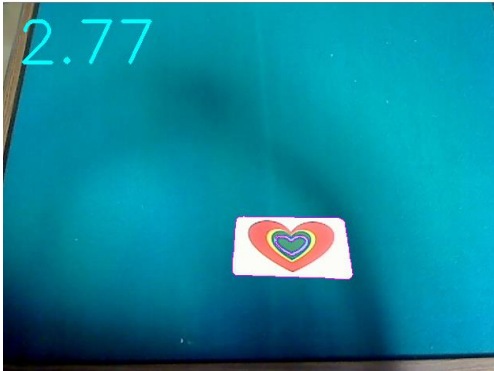


(1)～(4) どの色に関しても似たような結果が得られている。

まず、背景差分を取ったことでカードの領域が抽出された。そして、クロマキー合成を行ったことで、元々の背景は黒い画用紙だが、机の背景に置き換わっている。また、重心を求めて、そこでの画素値を取得したことでそれぞれの矢印と同じ色（緑、赤、黄、青）が色相として得られた。さらにこの色相をもとに色カードから矢印と同じ色の領域を抽出してやれば、領域の面積の和が求まる。この面積の和を背景差分によって求めたカードの領域の面積で割ってやれば、全体に占める割合を求めることができ、10 回実験した平均値が数値として画像の左上に表示されている。実際にはこの後割合順に並び替えて、ビデオに格納していくことでカードが正しい順番に並んだ動画画像が表示された。

4.2. 懸念される入力

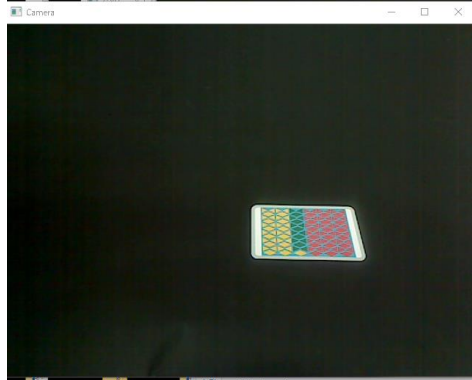
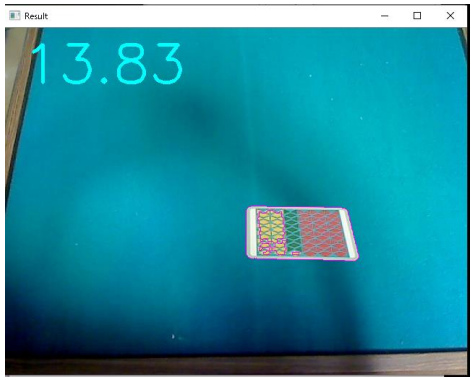
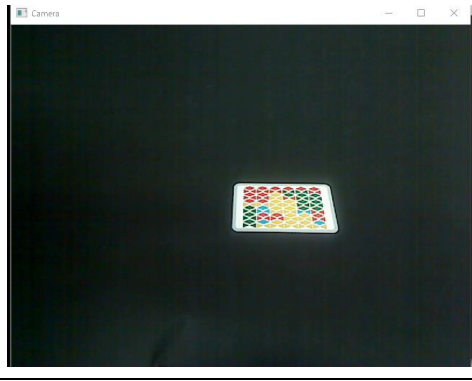
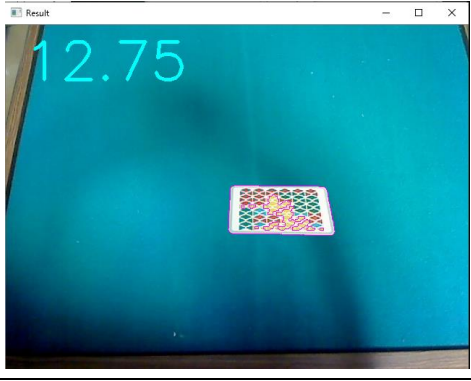
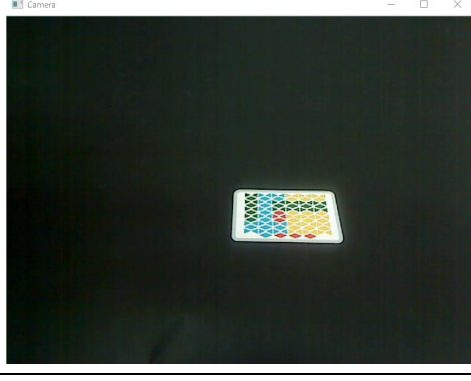
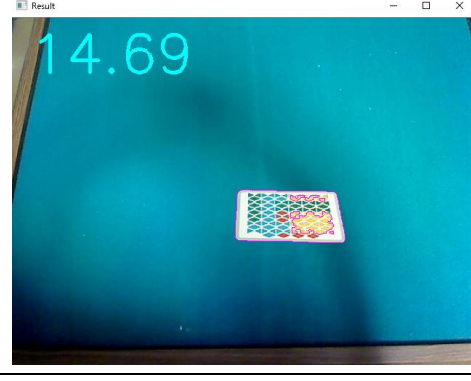
以下のような例ではうまくいかない可能性があると考えた。

(1) 領域が小さい

真の割合	入力画像	出力画像
4% (緑)		
3% (青)		
2% (赤)		

これらの結果が得られた理由は4.1. とほぼ同じである。違いとしてはクロージング処理によって、穴の開いてしまっていた領域も埋めることができたため、対象となる色領域のみが抽出されている。そのため、当初懸念していた面積の小さな領域が無視されることは阻止できた。

(2) 同程度の割合の画像同士でも、大きな差が出てしまう

真の割合	入力画像	出力画像
15% (黄)		
16% (黄)		
16% (黄)		

これらの結果が得られた理由も 4.1 とほぼ同じである。

ただし、これらのカードは真の割合が近いために面積の割合順に並び替えた場合には順番が正しく並ばない。クロージング処理やオープニング処理を行なっても、完全に面積を一致させるほどの精度ではないために結果が変わってしまっている。

5. 考察

5.1. 期待値に対する結果の評価

矢印カードから画素値を取得するのは期待通りの結果になっており、特に問題は見受けられなかった。しかし、色カードの処理に関しては、正確な結果が出たものも多かったがカメラから画像が遠ざかるとカードが歪んで認識されてしまうために正しい面積を算出できないということが問題となっていた。また、このとき向きを変えるとそれによっても面積が変わってしまった。特にこの傾向は面積が大きいものに見られ、ひどいものであれば10%ほどの誤差があるものも見受けられた。そのため、位置によって結果が変わらないように、カードを置く位置を固定してやるなどの対策が必要である。また、抽出する色によって精度に差が見られた。最も精度が高かったのは黄色で、最も精度が低かったのが緑であった。緑は背景差分が小さくなりやすく、状況によって閾値を調整してやらなければマスクされてしまう場合が多々あった。また、赤は白色、青は灰色、緑は黒と色相の値が似ているために関係のない領域も抽出されてしまった。そのため、今回はHSVのうち色相(Hue)にのみ注目したが、彩度と明度にも注目する必要がある。ただし、彩度と明度を絞ると本来の色領域が正しく抽出できない可能性もあるため、この点に関しても実験を行うべきであろう。

5.2. 懸念点に対する結果の評価

領域が小さい場合は予想に反してクロージング処理を行なってやれば領域をうまく抽出できていた。ただし画像によっては白色の領域がノイズとして出てしまい、クロージング処理によってこのようなノイズが除去できない状況もあった。そのため、現状ではクロージング処理が必ずオープニング処理よりも先に行われる仕様となっているが、オープニング処理とクロージング処理のどちらを先に行うか選択できるようにすべきである。また、面積がほぼ同じカード同士では、当初懸念していた通り、誤差などの影響が大きく期待通りの結果にはならなかった。特に大小関係が反転してしまう例もあり、この点がゲームを行う際に致命的になると思われた。

5.3. 実験全体への評価

今回の実験では目標となる処理自体は行えたが、プログラムだけではなく実験を行う環境など様々な要因で精度はそれほど高くはならなかった。しかし、面積の割合にある程度差のある者同士を比較した場合にはしっかりと順番通りに並べることができたため、結果的にはゲームを行える程度には精度を確保できた。