

物体認識のための画像解析

2年A組 中盛 楓也

望月 草馬

2年B組 小林 望

指導教員 藤野 智美

1. 要約

私たちは現在ロボットに搭載する画像処理システムについて開発をしている。今回はそのテスト段階として、映像の中からピンポン球を探し出し、座標を求めるシステムを開発した。

キーワード processing、OpenCV、AI、webカメラ、VR

2. 研究の目的と背景

現在、人が操作しなくても、AIによる自動判断で目的を達成できるロボットの開発を行っている。そのためにはwebカメラによる画像処理システムが必須になると考え、映像データのみからオブジェクトを認識し、座標を特定するシステムの開発を行った。processingとOpenCVをシステム開発のツールとして用いた。

3. ピンポン球の認識

3. 1 形状の認識

ピンポン玉を認識するために、私たちは色に注目した。ピンポン玉のオレンジという特徴的な色を認識できれば、その物体をピンポン玉として認識出来るのではないかと思い、「webカメラに写るオレンジのピクセルの平均座標=ピンポン玉」として認識させた。しかし、このプログラムではオレンジの物体全てに反応してしまい、ピンポン玉のみの認識が出来なかった。

そこで、ピンポン玉の形を認識することを試みた。ピンポン玉の円形を認識するためのアルゴリズムは以下の通りである。

- ① 画像のピクセルの色を調べ、オレンジが一番長く横に続いた場所を円形の直径とする。
- ② その円が接する正方形の場所ごとのオレンジの割合を調べる。
- ③ その正方形を図1のようにA、Bの領域に分け、Aの領域のオレンジのピクセルが5割以上、Bの領域のオレンジのピクセルが5割未満であれば、円形と認識した。その結果、ほかのオレンジ色の物体と区別して、ピンポン玉のみを認識することに成功した(図2)。

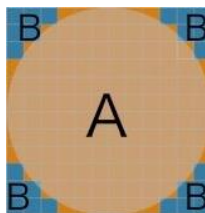


図1



図2

3. 2 webカメラからの距離

次に、認識したピンポン球の座標を求めるプログラムを考えた。最初にwebカメラとピンポン球との距離を求め、遠近法で画面上のピンポン球の大きさから距離を求める方法を考案した。まず、画面上のピンポン球のサイズとwebカメラの距離との関係を見つける。

<結果>



図3

図4

図3ではカメラからの距離が15cmで、画面上のピンポン球の直径は164ピクセルであった。図4ではカメラからの距離が30cmで、画面上のピンポン球の直径は83ピクセルであった。

これらの結果より、ピンポン球との距離(cm)を d 、画面上のピンポン球の半径(ピクセル)を r 、比例定数を a として式を立てると、以下の関係式が得られる。

$$d = \frac{a}{r}$$

写真2の場合、 $a=1230$ であり、写真3の場合 $a=1245$ となる。

<考察>

ピンポン球の直径とwebカメラからの距離はほぼ**反比例**することがわかった。

また、カメラによって比例定数が変わるが、今回のカメラでは何度か確かめた結果、比例定数は1230であった。

3. 3 左右のズレの測定

ピンポン球がwebカメラ(画面の中央)から左右にどれだけズレているかを求めるプログラムを考える。

webカメラから20cmのところに行った場合と30cmの場合で左右に6cmずらし、画面上でのずれたピクセル数を計測した。

<結果>

距離20cmの場合

→192ピクセル

距離30cmの場合

→122ピクセル

<考察>

結果から、

$$g = \frac{dp}{a}$$

d = webカメラからの距離(cm)

p = 画面上での左右のずれ(ピクセル)

g = 実際のwebカメラからの左右のずれとなり、定数 a は結果から、

$$6 = \frac{20 \times 192}{a}$$

というように求めることができ、複数回の実験の結果から a の平均は646となった。

よって次の式が成り立つ。

$$g = \frac{dp}{646}$$

3.2と同じように、比例定数はwebカメラによって変わる。何度か試した結果、今回のwebカメラでは比例定数が約646ということがわかった。これらの結果をもとに

実際にロボットが行くべき道筋を求めることができる。その関係図が図5である。

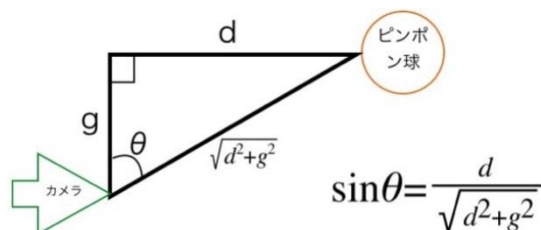


図5

4. VRへの応用

先述した内容を応用し、ヴァーチャル空間で現実の自分の動きを適用するVRコンテンツの開発を試みた。

4. 1 概要

webカメラデータから、プレイヤーの現在座標を求め、VR内のキャラクターに適用させるというものである。開発はUnityで行った。

4. 2 背景差分法

背景を事前に登録し、それをもとに人の位置を特定するという方法を試した。

アルゴリズムは以下の通りである。

- ① まず、動くものがない状態で画面上の全てのピクセルの色を記憶する(図6左上)。
- ② ①で記憶したそれぞれのピクセルで違う色を探し出し(図6中央)、そのピクセルの平均座標をプレイヤーの位置とする。

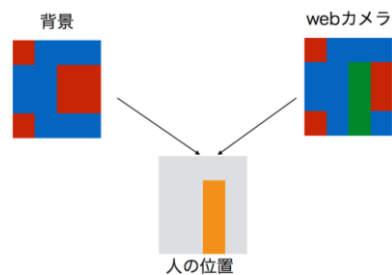


図6

<結果>

フルピクセルで調べると、ラグが多発してしまった。そのため、画素数を1/64のみ調べるといった方法をとると、多くの誤差が発生した。また、画面が少しでもずれると全てのピクセルが反応してしまい、正しく認識することができなかった。

4. 3 特徴検出法

体に色の付いたマーカを取り付け、そのマーカ的位置をwebカメラで見つけるという方法をとった。写真4はwebカメラの映像である。メカニズムは以下の通りである。

図7のようにピクセルを縦向きに調べていき、赤、緑、赤の順に並んでいるところをplayerの位置とする。

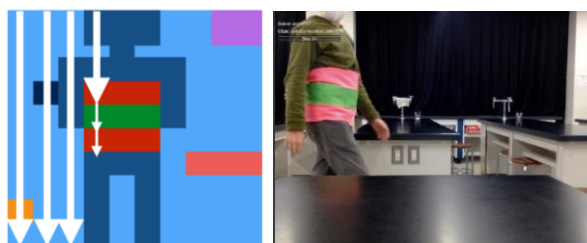


図7

図8

<結果>

今までで、一番スムーズかつ正確に反応し、この方法が一番実用的だと判断した。

5. コンテンツの仕組み

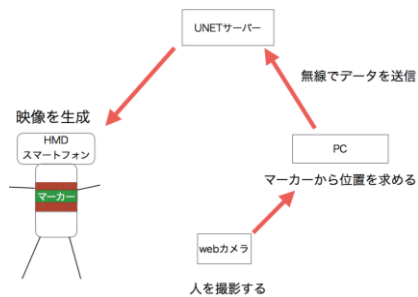


図 9

Unity を用いて、PC 上で上記の方法からプレイヤーの位置を求め、そのデータを無線で HMD 内のスマートフォンにインストールした App に送信する。そのデータをもとに、App でヴァーチャルワールド内の 1 人称映像を生成し、画面に表示する。また、スマートフォンのジャイロデータをもとに、視点の移動をする(図 9)。

6. 今後の課題

今後 VR で Kinect を使用し、人の手や足の動きも再現したり、時間のズレを解消していきたい。画像認識では、光の加減や、影のつき方で認識精度が大きく変わってしまうため、光加減を自動で認識するシステムが必要となる。今後もそのような視点を持って研究を進めたい。

また、今回はピンポン球という単純な形のオブジェクトを認識したが、もっと複雑な形のものを認識する方法も考えていきたい。加えて、複雑な形状の物体を学習するプログラムをも作っていきたい。

6. 謝辞

今回の研究を行うにあたり、顧問の藤野先生には多大なご指導をいただきました。この場をお借りして深く御礼申し上げます。