

画像処理を使った自走ロボットの研究

宮原 直雄

笠井 陽太

星野 圭吾

(長野県駒ヶ根工業高等学校 情報技術科)

1 研究動機・目的

自動車の自動運転技術に興味があり、画像処理を研究したいと考えた。マシンの製作を通して、1、2 年次に学習した応用技術や、さらに高度な技術を習得すること、画像処理の基礎知識を学ぶ。中間目標として、製作したマシンを使い 9 月 21 日に行われた ROBOCON IN 信州 Camera Class への出場目指した。

2 研究の基礎知識

(1) マイコンカラー Camera Class とは

マシンに搭載されたカメラの映像を用いて、決められたコースを自動走行しタイムを競う競技である。使用部品は【ジャンパンマイコンカラー Camera Class 競技規則】によって定められており、ボードコンピュータである GR-PEACH の他、指定されているカメラモジュール、サーボモータ、モータドライブ基板、その他にも複数の規定部品を使用しなければならない。マシンのサイズは全長 300mm 以内、幅 300mm 以内、高さ 150mm 以内と定められている。

(2) 画像処理について

画像や動画は pixel で構成され、1pixel ごと色情報等が含まれる。一般的に使用される RGB フォーマットとは、1 つの pixel に R(赤)、G(緑)、B(青)それぞれ 8bit 合計 24bit の情報を持つ形式である(図 1)。今回使用したフォーマット YCbCr422 とは Y(輝度)と Cb(青差分)、Cr(赤差分)で記録する方法で、Y(輝度)は 8bit、Cb、Cr(色差)は横方向に交互に 8bit ずつ記録され合計 16bit のデータとなる(図 2)。



図 1.RGB のデータフォーマット

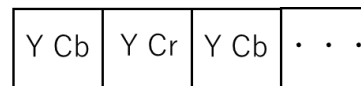


図 2.YCbCr422 のデータフォーマット

YcbCr422 は、人間の目は明るさの変化には敏感だが色の変化には鈍感である視覚特性を利用した圧縮方法で、Y(輝度)は各 pixel に配置されているが、Cb、Cr(色差)は 1pixel ごと間引いてデータ量を圧縮することで圧縮前の画像データと同等の画質を保つことができる。YCbCr は、RGB とは異なる色空間だが、相互に変換することが可能である。RGB から YCbCr への変換には、特定の計算式が用いられる(図 3)。逆に、YCbCr から RGB への変換も同様に行うことができる。

$$\begin{aligned} R &= 1.164(Y-16) + 1.596(Cr-128) \\ G &= 1.164(Y-16) - 0.391(Cb-128) - 0.813(Cr-128) \\ B &= 1.164(Y-16) + 2.018(Cb-128) \end{aligned}$$

図 3. YCbCr から RGB への変換式

カメラから取得したデータは非常に大きいデータになり、処理を行うことが非常に困難なため、データを軽くして処理をやすくするために輝度データ(Y)を取り出しグレースケールで使用したり 2 値化処理を行ったりすることもできる。2 値化とはしきい値を基準に白、黒の値に変換することである。2 値化を行うことでエッジの強調や画像データを軽くすることができる。

3 研究内容

(1) マイコンカーの設計と製作について

Autodesk 社の Fusion という 3DCAD ソフトを使用し、公式大会の規則に沿って車体設計を行った。マイコンカーラリーCamera Class の競技規則には、全長 300mm 以内、幅 300mm 以内、高さ 150mm 以内という規定があり、コースには 10° 以下の坂が設けられている。そのため、車体サイズに関係なく実際に全体の様子を見ることができる 3DCAD を使用し設計した。設定の手順として最初に 2D 画面で設計したい物の平面図を描く。次にそれらを押し出しという機能を用いて立体にしていく(図 4)。基本的にはこれらの動作の繰り返しである。他にも穴あけ、フィレット、面取りなど様々な機能が備わっているため、複雑な形でも作成可能である。

メインシャーシ等のカーボンパーツは 3DCAD で設計時に作成した平面図を DXF 形式に変換し、Autodesk 社の AutoCAD を使用し、図面の細かな修正や CAM ソフトに入れた際に加工経路を作成しやすくする補助線などを追加した。その後 VCarve という CAM ソフトを使用し、切削するための加工経路を作成し、CNC 加工機を用いて加工を行った(図 5)。

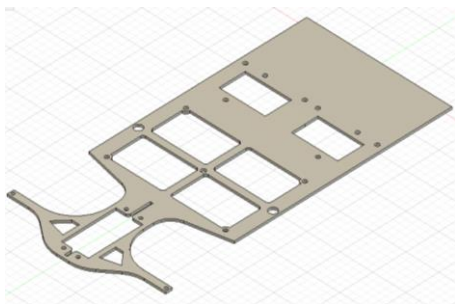


図 4.3DCAD で設計したメインシャーシ

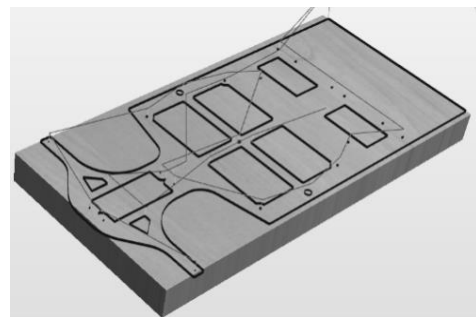


図 5.VCarve を用いた加工経路設

CNC では加工が難しい足回りの部品(図 6)や、カメラスタンド(図 7)は 3DCAD で設計したものを stl 形式に変換し 3D プリンタで造型した。

カメラスタンドは土台をしっかりと固定できるように剛性を高めに設計した。上部はカメラの固定をするだけなのでなるべく細く、H 形に設計し、走行中のねじれを防止した。

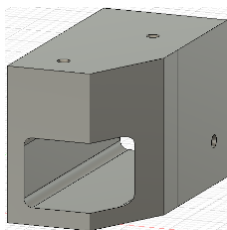


図 6.車軸ユニットの 3D データ

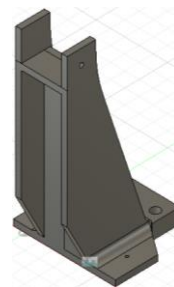


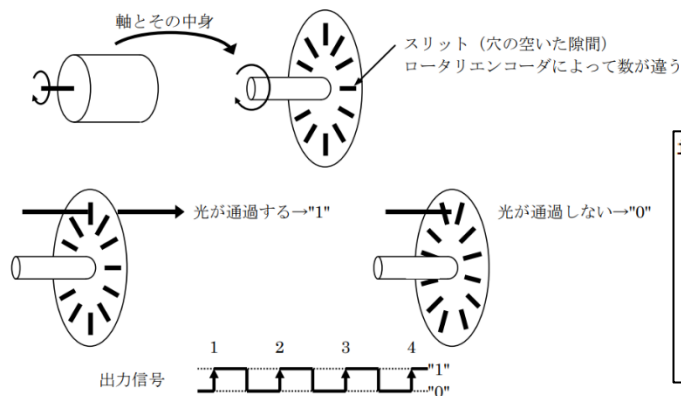
図 7.カメラスタンドの 3D データ

(2) マイコンカーの制御とプログラムについて

本研究では、マシンの制御に PWM 制御、エンコーダ、サーボモータ等を使用している。PWM 制御は、駆動モータの回転速度を制御するものであるが、3 年生の実習にて学んだため今回は説明を割愛する。

エンコーダとは、ロータリ(回転する)エンコーダ(符号化、数値化)という意味で、本研究ではスピード制御をするために使用している。ある一定時間の光の透過回数をカウントして、スピードや距離制御ができる(図 8)。本研究では使用したエンコーダはタイヤ径が 21mm、1 周のカウント数は 72 である。10ms に 1 回処理が行われ、現在のカウント値である MTU2TCNT_1 を変数 i に移し、

最新値とする。iEncoder には i から 10ms 前の値である uEncoderBuff を引き現在のスピードを測っている。lEncoderTotal には iEncoder を加算しており、走行時のトータルの距離を測ることができる(図 9)。



```
iTimer++;
if( iTimer >= 10 ){
    iTimer = 0;
    i = MTU2TCNT_1;
    iEncoder = i - uEncoderBuff;
    lEncoderTotal += iEncoder;
    uEncoderBuff = i;
}
```

図 9 .プログラムの処理

サーボモータは規定で指定されている HI-TEC 製 HS-430BH を使用し、ステアリング制御を行っている。昨年の全国大会のマシンを参考にし、梶切時にカメラとタイヤ両方を同時に動作させることにした。これにより、走行時の安定性が増し、コースアウトもしにくくなった。

ステアリングの切角は handle 関数で変更し、引数を -1 とすると左に約 4 度切る。motor2 関数は左右の駆動輪のモータバランスを変更するものである。switch 文を使用して二値化したセンサーパターン(4bit)に応じた条件分岐を行っている(図 10)。

```
switch( (sensor_inp() & 0x0f) ) {
    case 0x00:
        handle( 0 );
        motor2( 20, 20 );
        break;
    case 0x02:
        handle( -1 );
        motor2( 20, 20 );
        break;
}
```

図 10.走行プログラムの中身(一部抜粋)

(2) 画像処理について

学習した基礎知識をもとに YCbCr で取得したカメラの画像データの確認を行った(図 11)。コース上に色のついたラインを上から青、ピンク、緑、赤、黄色の順番に並べカメラで読み込み Tera Term (通信ソフト) を使い CSV 形式のデータを取り出し、Excel に取り込み画像に変換し理論どおりになるのか確認を行った。

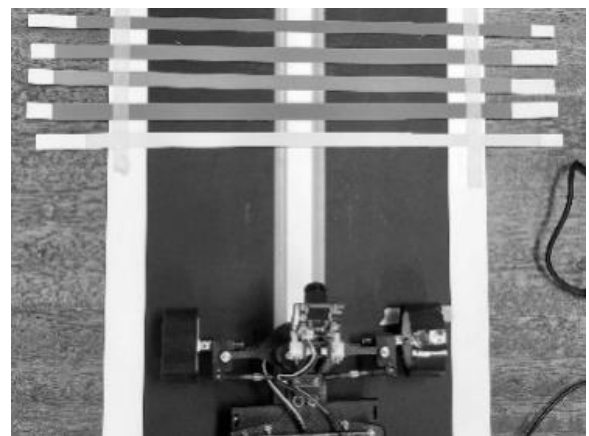


図 11.色情報の確認実験の様子

① YCbCr422 から RGB への変換

実際に Excel に取り込んだ。YCbCr のデータ (D 列～F 列) を図 3 の変換式を参考に RGB データ (H 列～J 列) に変換を行った(図 12)。

	A	B	C	D	E	F	G	H	I	J
18	//	X座標	Y座標	Y	Cb	Cr		R	G	B
19	#YCbCr	0	0	14	129	125		0	0	0
20	#YCbCr	1	0	50	129	125		35	42	42
21	#YCbCr	2	0	255	131	121		255	255	255
22	#YCbCr	3	0	255	131	121		255	255	255
23	#YCbCr									
24	#YCbCr	セル内の YCbCr から RGB への変換式 $=IF(ROUND(1.164*(D20-16)+1.596*(F20-128),0)<0,0,IF(ROUND(1.164*(D20-16)+1.596*(F20-128),0)>255,255,ROUND(1.164*(D20-16)+1.596*(F20-128),0)))$								
25	#YCbCr									
26	#YCbCr									
27	#YCbCr									
28	#YCbCr									

図 12. YCbCr データを RGB に変換したデータ
(注.数値データ一部抜粋)

② Excel のシート上に画像を生成

YCbCr から異なる色空間である RGB 正しく画像と認識できているのか確認するために①で作成したデータを使い Excel のセルを pixel にみ立てて、対応したセルに対応した色を付ける VBA を組み画像を生成した画像サイズは 320(W)×240(H)pixel である。

確認を行い実際に CSV 形式のデータからカラー画像を作成できることと、YCbCr から RGB に正しく変換されていることが分かった。黄色は白飛びが激しかったため、色を取り出すのが難しいことが分かった。

③ 輝度データ取り出し

マイコンカーでは輝度データを取り出して、グレースケール画像にして、そこから平均化を行うため、実際にどのようなグレースケール画像になるのかの確認を行った。Excel VBA を使い Y(輝度)データを Excel の sheet 上に 320(W)×240(H)pixel 並べるソースコードを作成し、実際に輝度データを取り出し並び替えた(図 13)。その後、条件付き書式を使用し、8bit(255 段階)のデータを 10 段階に分けてグレースケール画像に変換した(図 14)。

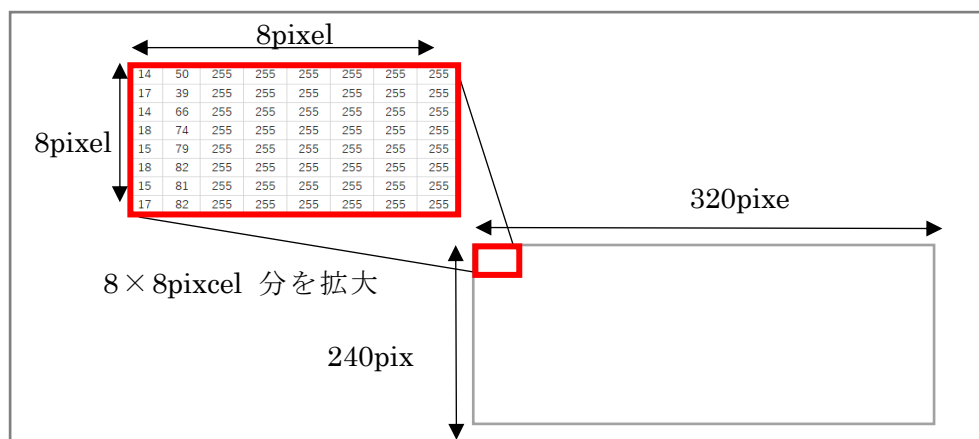


図 13.画像縮小 (注.数値データ一部抜粋)

① 解像度変更による画像縮小と 2 値化

マイコンカーでは③で記した 320(W)×240(H) pixel の画像を 8×8pixel を 1pixel として平均化を行い、20(W)×15(H)pixel にサイズを変更している。具体的には、図 13④の拡大図部分の値を平均化したものを 1pixel に置き換えてこの作業を Excel VBA を使い行った。その後シート上の条件付き書式を使い白黒データへの変換(2 値化画像の作成)をおこなった(図 15)。

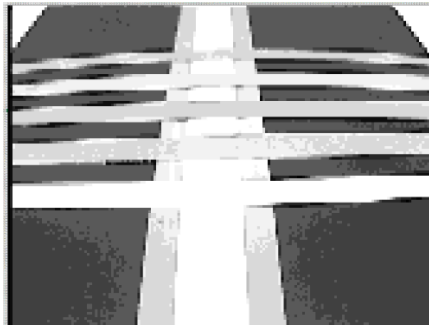


図 14. 10 段階グレースケール画像

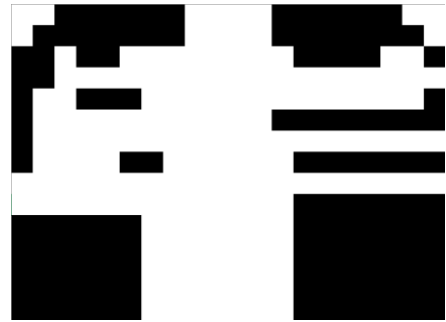


図 15. 2 値化画

(3) 色判別プログラム

上記の実験を経て、赤色は光の影響をあまり受けず制御しやすいことが分かったため、赤色を認識し停止するプログラムを作成した。赤色が視界から外れると走行を再開するようにプログラムした。実験から Cr(赤色)の値が 190~200 前後であることが分かったので、閾値を 200 に設定した。1pixel では誤検知が多発すると考えたため、5pixel の平均値で判断するようにした。縦方向(y)は中心で固定、横方向(x)は中心の 1pixel と左右に 2pixel ずつの値を取り出し平均化する関数 CrAverage を作成した。

色判断には if 文を使用し、CrAverage と閾値である 200 を比較し、平均値が上回った場合に停止させるプログラムを作成した。赤色が外れた時の判断は if 文条件式の不等号を反転させることで動作している。外乱が入ることによって赤色が外れたと誤検知し、走行を再開してしまうことが多々あったため、プログラム上で 5 回連続でカウントした後に走行を再開するようにした。

4 研究結果

(1) 車体の製作

Fusion を使用し、公式大会のレギュレーションに沿って車体設計ができた(図 16)。また、中間目標であった ROBOCON IN 信州については、夏休みにも研究に取り組むことにより、ハードウェアの完成と基礎の走行プログラムの知識を身に付け出場することができた。完走をすることはできなかったが 3 位という成績を修めることができた。製作したマシンはサーボモータによるステアリングの操作や、エンコーダによる距離やスピードの制御を行い、より高度な制御技術も身につけることができた。

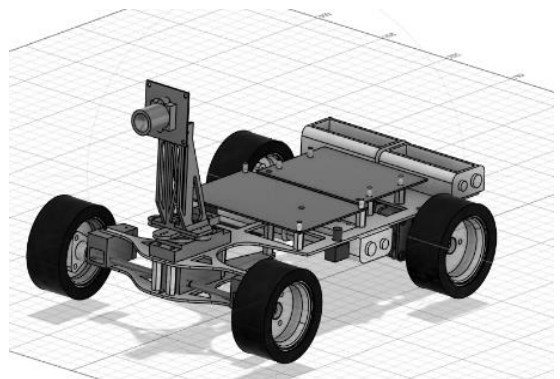


図 16. Fusion を用いて設計したアセンブル図

(2) 画像処理について

確認実験の画像と Excel VBA を使い YCbCr から RGB 変換を行い、画像を見比べてみたが、一番手前の黄色の線の白飛びが激しく黄色なのか白なのか区別が付きにくくなってしまった。実験の画像より鮮やかさは少し劣ったが、赤、緑、ピンク、青の色を YCbCr 形式の CSV ファイルから取り出せることを確認できた。また、データ容量は RGB の時の 2/3 に圧縮されていることが確認できた。

マイコンカーでは、取得した 320(W)×240(H) pixel のデータの中から Y(輝度)を取り出し 2 値化を行い 8(W)×8(H)pixel の平均値をとって 20(W)×15(H) pixel にサイズ変更を行って画像データを少なくして処理を効率よく行っていることが分かった。実験のデータでは色情報を含んでいてコースを判別することが難しいので、クランクのコースも Excel VBA を使い変換を行った。

320(W)×240(H) pixel のデータ(図 17)から Y(輝度)を取り出し 8(W) ×8(H)pixel の平均値をとって 20(W)×15(H) pixel にサイズ変更し、2 値化を行った。結果、2 値化後もクランクがあることが画像から読み取ることができた(図 18)。

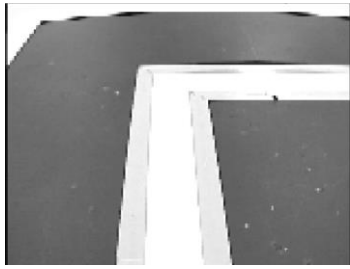


図 17. クランクグレースケール画像



図 18. 2 値化したクランク

5 考察・まとめ

画像処理マイコンカーの製作と研究を行いこれまで学んできた基礎を基に車体の設計、組み立てを行うことができた。3D プリンタ、CNC 加工機の使い方などの新しい技術も学ぶことができた。

3D プリンタを使い印刷をしている際に印刷がずれてしまい思うような部品を作成することができず苦労したが、部品を太くしたり印刷の向きなどを変えたりすることでうまく印刷することができた。

実習で学んできたマイコンカーをより正確に動かす制御方法や、画像の取り込み方やフォーマットなどの原理について学習することができた。

最初は色判断のほかにも、形による判別をしようと考えていたが、画像データは複雑で膨大なデータ量を使うためとても難しく、画像から色データを取り出すだけでも大変で形による判別まで行うことができなかった。画像データの処理だけでも大きなデータ容量を使うことが分かり、動画データになるとより大きなデータを使うことが分かった。また、実験を行い VBA で変換した画像では黄色の白飛びが激しく白との区別がつきにくくなってしまった。このことからカメラは外部の影響をととても受けやすいことが実験を通して知ることができた。OpenCV などの画像処理、画像解析などを行えるライブラリを使い取得した画像の補正を行い外部の影響を減らすことで今回白飛びしてしまっていた黄色の線もより正確に読み取ることができると考えた。

現在人工知能や自動運転技術で重要な役割を果たしている画像処理を支える基礎的な知識を知ることができ、改めて今使われている技術のすごさを体感した。

6 謝辞

本研究にあたりましては、長野県産業教育振興会様より研究助成をいただき実施することができました。また、3 年間私たちの指導をしてくださった駒ヶ根工業高等学校の先生方をはじめ、活動を支えてくれた家族、大会運営にご尽力くださった皆様等のおかげで研究を進めることができました。本研究を支えてくださった多くの皆様に感謝申し上げます。

7 参考文献

- Fusion360 マスターガイド ベーシック編 : ソーテック社 小原照記・藤村祐爾[著]
「GR-PACH」で始める電子工作 : 工学社 GADGET RENESAS プロジェクト[著]
OpenCV による画像処理入門 : 講談社 小枝正直・上田悦子・中村恭之[著]
JMCR 2024 講習会資料 : (株)日立ドキュメントソリューションズ
ロータリエンコーダ kit12_38a プログラム 解説マニュアル :
https://www2.himdx.net/mcr/product/download/encoder_kit12_kaisetsu_r8c38a.pdf