

2013年 自立型ゴルフロボット競技優勝
カトちゃん号の概要と画像処理

所属: 山梨大学工学部コンピュータメディア工学科 丹沢研究室
ロボット名: カトちゃん号
メンバー: 青柳貴弘 三浦範之 加藤信

目次

| | | |
|-----|-----------------------------|----|
| 第1章 | はじめに..... | 1 |
| 第2章 | 作成したロボットの概要..... | 2 |
| | 2.1 作成したロボットの概要..... | 2 |
| | 2.2 ロボットの動作戦略..... | 2 |
| 第3章 | ステレオカメラ..... | 4 |
| | 3.1 ステレオカメラの概要..... | 4 |
| | 3.2 ステレオカメラの原理..... | 4 |
| | 3.3 ステレオマッチング..... | 7 |
| | 3.4 カメラキャリブレーション..... | 9 |
| | 3.4.1 カメラキャリブレーションの必要性..... | 9 |
| | 3.4.2 シェーディング補正..... | 9 |
| | 3.4.3 歪み補正..... | 11 |
| 第4章 | ボールとバンカーの検出手法..... | 16 |
| | 4.1 ボール検出..... | 16 |
| | 4.1.1 ボール検出の手順..... | 16 |
| | 4.1.2 色抽出と HSV 色空間..... | 16 |
| | 4.1.3 ラベリング..... | 18 |
| | 4.1.4 実面積を利用したボール判別..... | 20 |
| | 4.2 バンカー検出..... | 22 |
| | 4.3 光源に対する対処..... | 24 |
| | 4.4 処理時間..... | 25 |
| 第5章 | まとめ..... | 27 |

第1章 はじめに

ロボコンやまなし 大学一般の部 自立型ゴルフロボット競技では各チームがロボットを作成し，完全自立なロボットに図 1.1 のようなコースを用いて，ゴルフを行わせ，カップインまでの打数や時間などで成績を競う．カップのポールからは超音波及び赤外線が送波されており，方向と距離が識別できるようになっている．



図 1.1 2013 年度大会で用いられたコース

我々のカトちゃん号はステレオカメラを用いた画像処理によってボールの検出及び距離の測定とバンカーの検出を行っていることが特徴である．我々はこのロボットによって図 1.1 に示した 2 つのコースでカップインを成功させ，2013 年ロボコンやまなし自立型ゴルフロボット競技で優勝した．

本文では，次年度以降にゴルフロボット競技に参加するチームの参考とするために，2013 年に優勝したカトちゃん号の詳細，及びその画像処理の手順について解説する．第 2 章でロボットの概要について解説し，第 3 章でロボットに用いたステレオカメラの原理とキャリブレーションの方法について解説し，第 4 章でボールとバンカーの検出手法について解説する．ステレオカメラを用いない場合には第 2 章と第 4 章を参考にして欲しい．

第 2 章 作成したロボットの概要

2.1 作成したロボットの概要

作成したロボットの外観を図 2.1 に示す。

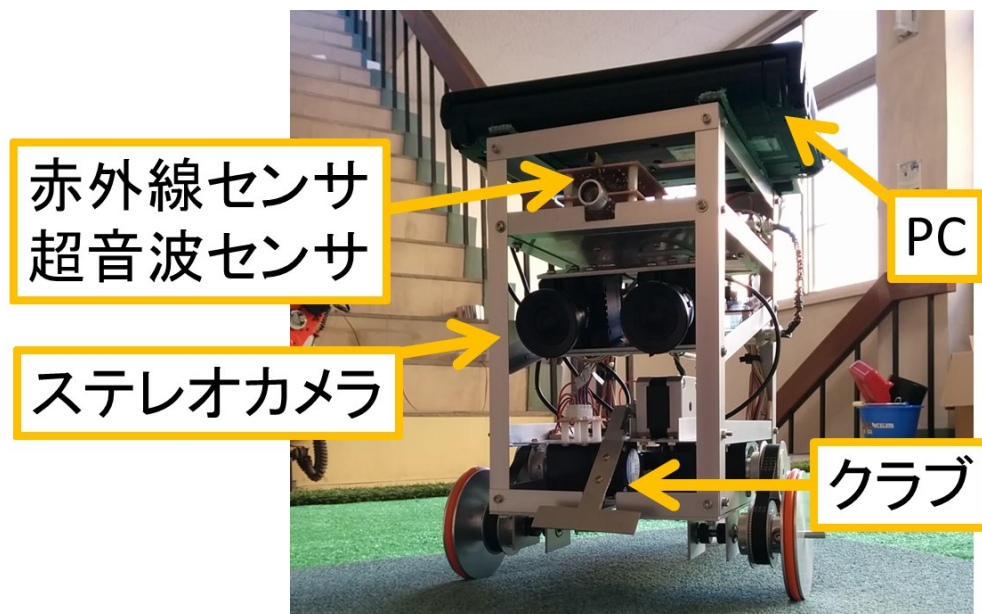


図 2.1 作成したロボット(カトちゃん号)

作成したロボットはセンサとして赤外線センサ，超音波センサ，ステレオカメラを搭載している．赤外線センサ，及び超音波センサはカップのポールから送波される赤外線と超音波を受信し，カップの方向と距離を計測するためのものである．赤外線が受信される方向にロボットを旋回させることでロボットをカップに向けることができる．ステレオカメラはボールの検出と距離の計測，及びバンカーの検出を行う為のものである．ステレオカメラを用いるには画像処理が必要であるため，ロボット上部に PC を搭載している．搭載した PC の詳細は次のとおりである．また，搭載するプログラムはすべて C++にて記述した．開発は Visual C++ 2010 にて行った．OpenCV 等の画像処理ライブラリは用いていない．

- ・ CPU Core i3 2350M 2.3GHz 2 コア
- ・ RAM 4GB
- ・ OS Windows7 64bit

ロボットの移動とクラブのスイングにはステッピングモータを用いた．

2.2 ロボットの動作戦略

作成したロボットは次のような 5 つの動作手順を繰り返す．

- ① ボールを打つ
- ② ステレオカメラを用いてボールを検出し距離，角度を算出

- ③ 算出した距離と角度を用いてボールに移動する
 - ④ 赤外線センサを用いてカップの方向にロボットを向ける
 - ⑤ ボールを打つ方向のバンカーの有無をステレオカメラを用いて判別し、回避する
- 図 2.2 にロボットの動作を示す。

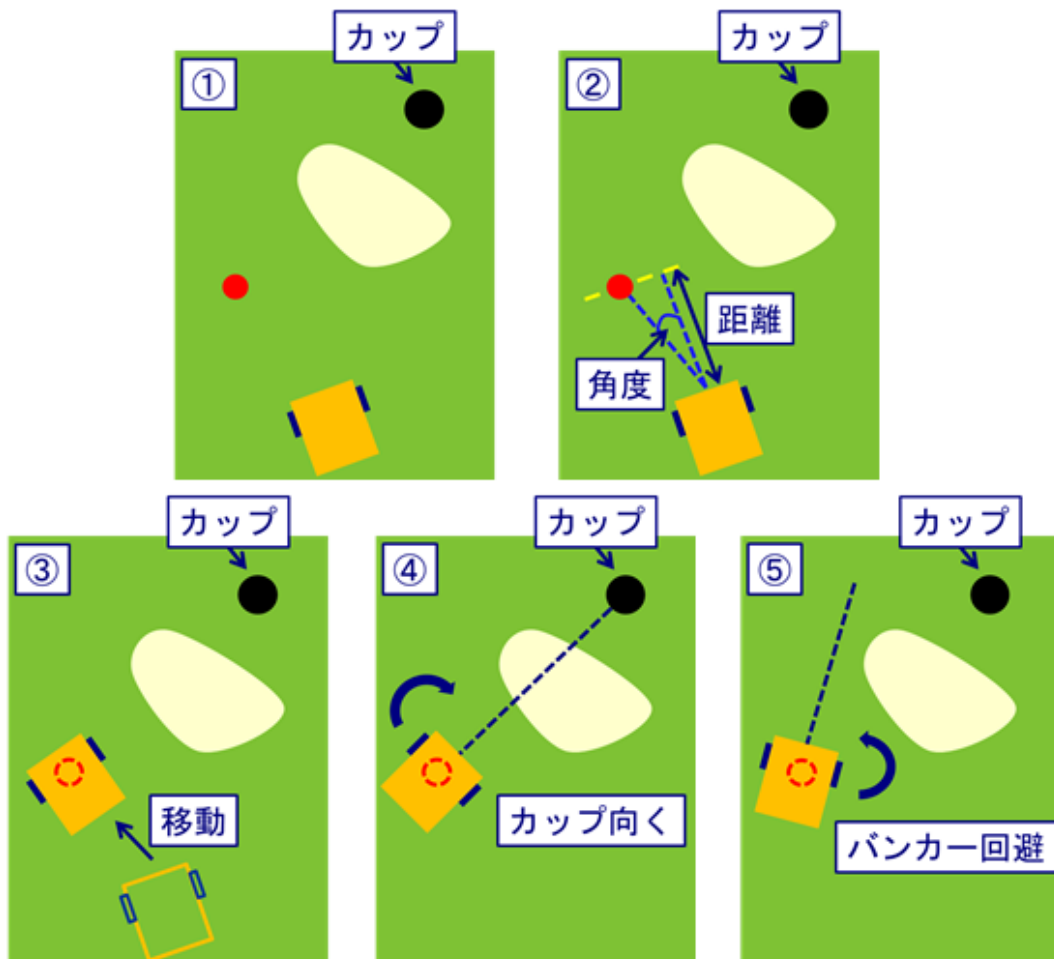


図 2.2 ロボットの動作

ただし、③のボールまでの移動中、ロボットはボールの座標精度向上のため、一定距離ごとに座標を再計算しながら移動する。

第 3 章 ステレオカメラ

3.1 ステレオカメラの概要

本ロボットの最大の特徴はステレオカメラを搭載していることである。ステレオカメラとは図 3.1 のように複数台のカメラを平行に設置したカメラで、物体までの距離を測定することができるカメラである。通常のカメラと同じように、物体の色や形状の情報も通常の画像として得ることができる。



図 3.1 ステレオカメラ

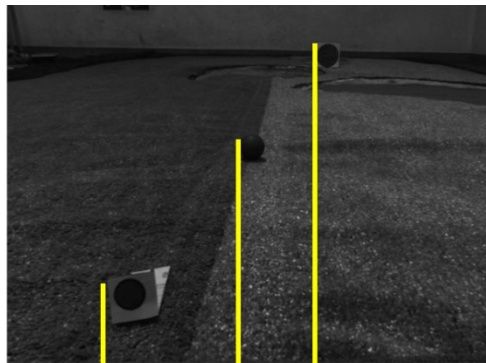
今回ロボットに搭載したステレオカメラは研究用のレンズの交換が可能なセンサーテクノロジー社製の CCD カメラを 2 台使用したものであり、非常に高価である。しかし、ロボコンに必要とされる程度の精度であれば、通常の Web カメラ等でも作成可能である。ただし、距離情報を得るためには、カメラのレンズの焦点距離および、画像素子のサイズと解像度は既知である必要がある。このため、それらのパラメータがわかる製品を選ぶか、パラメータを測定、推定する必要がある。また、ステレオカメラは基線長と呼ばれるカメラ間距離の長さによって観測可能な距離の範囲が変わる。詳しくは後述するが、基線長が短いほど、近方の観測精度が上がり、長いほど遠方の観測精度が上がる。

3.2 ステレオカメラの原理

ステレオカメラとは 2 台以上のカメラを並行に配置した距離情報を得ることのできるカメラである。ステレオカメラでは各カメラの物体の写り方のズレの量から距離情報を取得する。このズレを視差と呼ぶ。視差は近方の物体ほど大きく、遠方の物体ほど小さくなる。実際にステレオカメラで撮影した画像を図 3.2 に示す。右カメラで撮影した画像に比べ、左カメラで撮影した画像は全体に右方向にずれており、視差があ

ることが確認できる．また，遠方の物体ほど視差が小さいことも見て取れる．

右カメラ画像



左カメラ画像

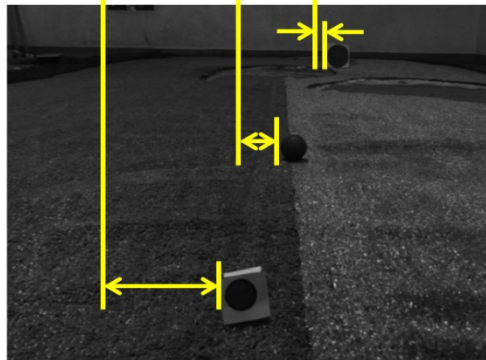


図 3.2 視差の例

ステレオカメラで視差を得るためには基準画像の任意の点が比較画像上のどこにあるかを探索する必要がある．2 次的にこの探索を行うと計算量が莫大になるため，エピポーラ線の考え方を導入し，探索を一次元に限定する．図 3.3 にカメラを 2 台設置した時の光学モデルを示す．

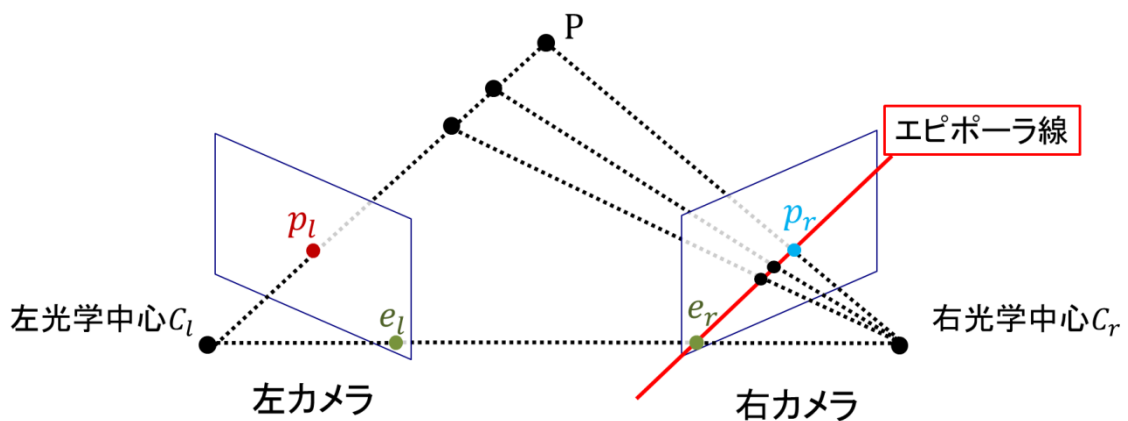


図 3.3 カメラを 2 台設置した時の光学モデル

左右のカメラの光学中心 C_l 、 C_r はそれぞれ e_l 、 e_r に投影される．物体 P を考えるとそれぞれ左右のカメラの p_l 、 p_r に投影される．直線 $P-C_l$ 上の任意の点は左カメラでは全て

点 p_l に投影され、右カメラでは直線 p_r-e_r 上のどこかに必ず投影される。この直線 p_r-e_r をエピポーラ線と呼ぶ。点 p_l の右カメラ上の対応点を探す場合、対応点はエピポーラ線上に必ずあるため探索を行うのはエピポーラ線上の点のみで良い。これにより、視差を求める際の探索を一次元に限定することができる。

並行ステレオカメラでは、2台のカメラの焦点距離、画像平面のサイズなどのパラメータを同一にし、2つのカメラの光軸が並行かつ、両画像平面が同一面上かつ、エピポーラ線が x 軸に平行かつ、両画像平面のエピポーラ線の y 軸座標が一致するようにカメラを並行に配置する。このように配置することで探索の際 x 軸方向に探索を行うだけで良くなるため容易に対応点を求めることができる。また、このように配置した場合、左画像の物体の投影位置は必ず右画像の同じ物体の投影位置よりも右側になる。同一のステレオカメラの光学モデルを図 3.4 に示す。

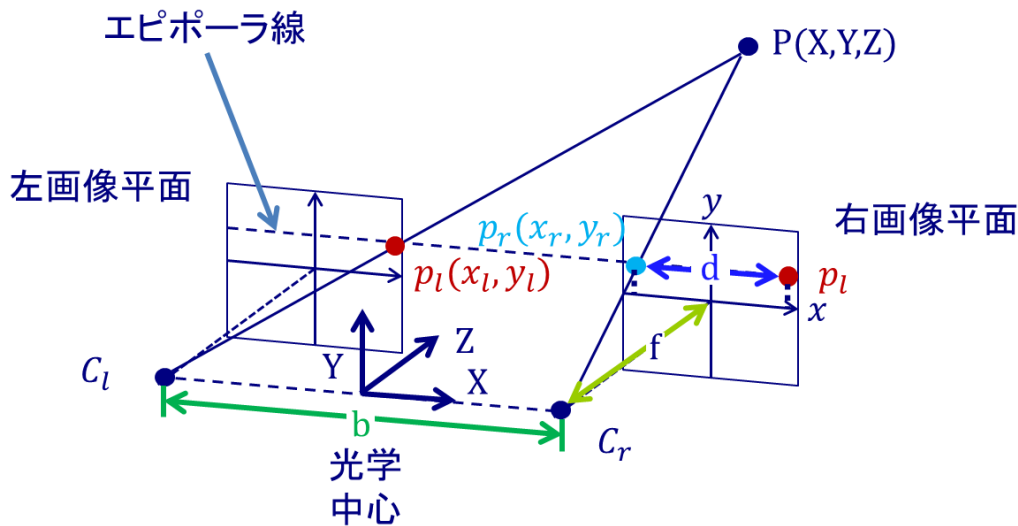


図 3.4 ステレオカメラの光学モデル

図中の f はレンズの焦点距離であり、図中の b は左右の光学中心間の距離、すなわち基線長である。任意の点 $P(X, Y, Z)$ を考えると、点 P は左右の画像平面の $p_l(x_l, y_l)$ 、 $p_r(x_r, y_r)$ に投影される。左右のエピポーラ線の y 軸座標は一致しているため $y_l = y_r$ である。ステレオカメラの光学中心を原点とした点 P の実空間上の座標 (X, Y, Z) を求めるのに必要な視差量 d は式 3.1 より求めることができる。

$$d = |x_l - x_r| \dots (3.1)$$

ステレオカメラの光学中心を原点とした点 P の Z 座標は $\triangle P, C_l, C_r$ と $\triangle C_r, p_r, p_l$ の相似関係から、基線長 b 、焦点距離 f 、視差量 d を用いて式 3.2 より求めることができる。

$$Z = \frac{b \cdot f}{d} \dots (3.2)$$

同様に他の三角形の相似関係から実空間上の X , Y 座標はそれぞれ式 3.3, 3.4 で求めることができる.

$$X = \frac{b(x_l + x_r)}{2d} \dots (3.3)$$

$$Y = \frac{b \cdot y_r}{d} \dots (3.4)$$

以上の式より任意の点 P のステレオカメラを原点とする実空間上の座標 (X, Y, Z) を求めることができる.

式 3.2 を見ると視差量 d は分母にあり, 視差 d に対する距離 Z のグラフは反比例するような形になる. このことは, 視差量が小さいほど距離に対する分解能が悪くなることを示す. 観測したい距離に対応する視差量が小さくなり過ぎないように, 観測したい範囲にあわせて基線長 b と焦点距離 f を決める必要がある.

3.3 ステレオマッチング

前述したとおりステレオカメラでは実空間上の座標を計算するのに視差量が必要である. この視差量を求める処理をステレオマッチングと呼ぶ. ステレオマッチングでは基準画像の各画素に対応する画素を比較画像上で見つけ, 画素単位のズレを求めることで視差量を求める. 対応画素の探索は注目画素に小さな窓領域を設け, 基準画像の注目画素と比較画像の注目画素の窓領域のパターンマッチングを行うことで行われる. 比較画像の注目画素を x 軸方向にずらしながらパターンマッチングを繰り返し, 最も類似度の高い画素を基準画像の注目画素に対応する画素として視差量を算出する. この処理を基準画像の各画素に対して行い, 各画素の視差量を求める. 基準画像を右画像, 比較画像を左画像とし, 注目画素を (x_1, y_1) , 対応点を x_2 とした時の処理の概要を図 3.5 に示す. 並行ステレオの条件から対応点は必ず基準画像の投影点よりも右側に存在するため比較画像の探索は基準画像の注目画素の画像座標から右方向に向かって行えば良い.

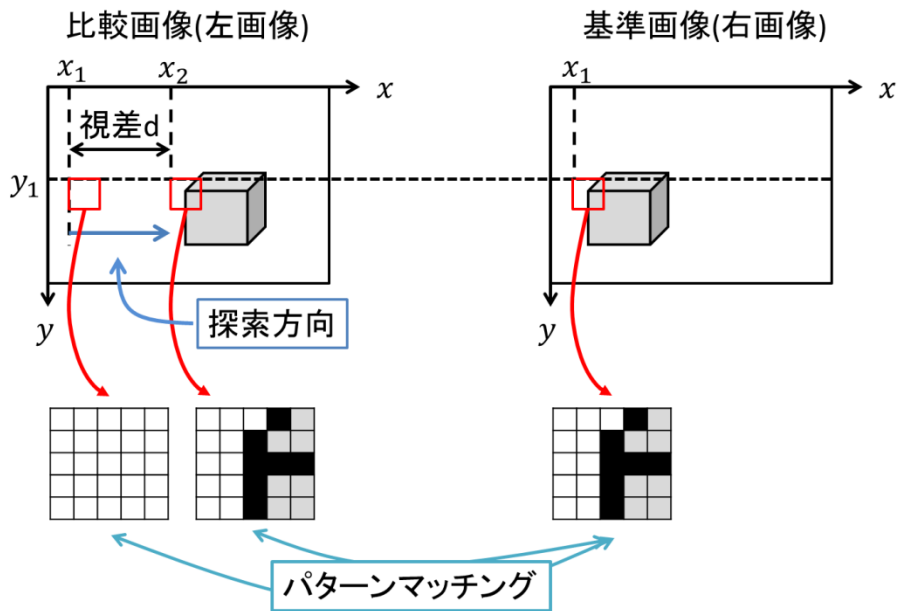


図 3.5 ステレオマッチングの概要

パターンマッチングには SSD(Sum of Squared Difference)や SAD(Sum of Absolute Difference), 正規化相互相関など輝度値を用いる方法の他, フーリエ変換を用いて周波数領域で比較を行う方法など様々な方法がある. 今回は単純で Intel MMX の PSADBW 命令による SIMD 実行が可能な SAD を用いる. SIMD 命令を用いることで大幅な高速化が可能である. SAD では類似度の量基準として輝度値の絶対値差分の合計を用いる. SAD では計算結果の値が小さいほど類似度が高いといえる. SAD の式を式 3.5 に示す. ここで, N, M は窓領域の幅と高さ, $I(i, j), T(i, j)$, はそれぞれ基準画像と比較画像の窓の輝度値である.

$$R = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} |I(i, j) - T(i, j)| \quad \dots(3.5)$$

実際にステレオ画像をステレオマッチングした結果を図 3.6 に示す. ステレオマッチングの結果として出力される視差量を記録した画像を視差画像と呼ぶ. 図 3.6(a)のようなステレオ画像をステレオマッチングすることで図 3.6(b)のような視差画像を得ることができる. 図 3.6(b)では輝度が高いほど視差が大きく近い物体である.

ステレオマッチングによって求めることができる視差量はピクセル単位だが, 実際の座標の算出に用いる視差量は長さである. このため, 座標の算出を行う際はピクセル単位の視差量に実際のカメラの画像素子の 1 ピクセルのサイズをかけたものを用いることになる.

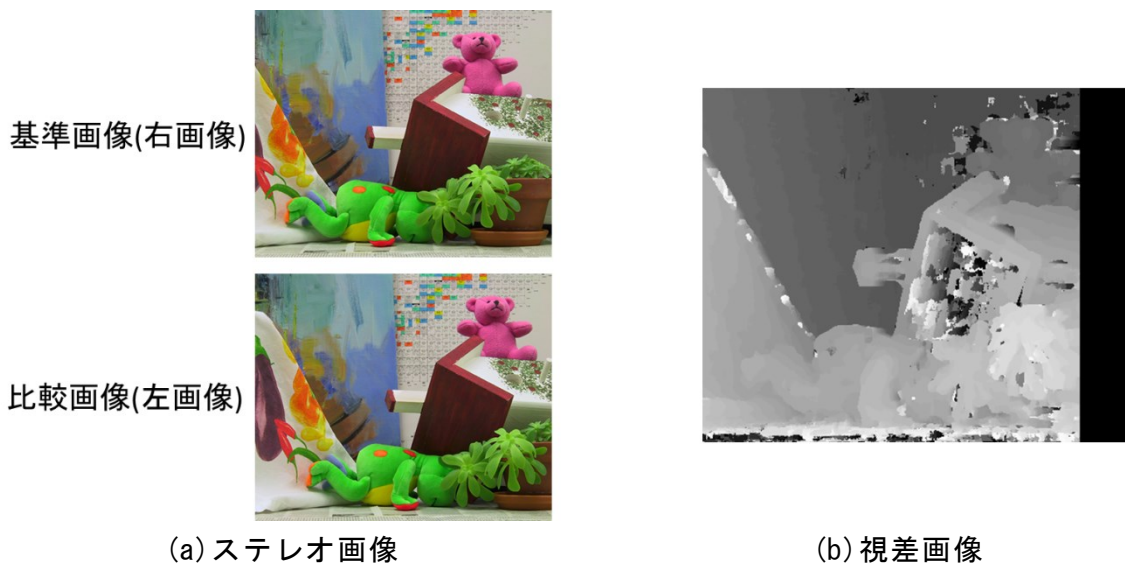


図 3.6 ステレオマッチングの結果の例

3.4 ステレオカメラのキャリブレーション

3.4.1 カメラキャリブレーションの必要性

理想的なステレオカメラであれば撮影した画像をそのままステレオマッチングして視差画像を得ることが可能だが、実際にはただ並行にカメラを配置しただけではレンズやカメラの特性や個体差の問題が原因でステレオマッチングが可能なステレオ画像は得られない。このため、カメラキャリブレーションを行って画像をステレオマッチング可能なステレオ画像に補正する必要がある。

ステレオ画像を得るためには次の 2 つの補正を行う必要がある

- 1.シェーディング補正
- 2.歪み補正

シェーディング補正はレンズの特性による減光やカメラのセンサ感度の個体差によって発生する輝度ムラなどを補正する処理である。ステレオマッチングでは左右の画像の輝度値を基準に比較するため同じ物体が同じ輝度値で写るように補正する必要がある。

歪み補正はレンズの光学特性によって生じる周辺部の歪みを補正する処理である。ステレオマッチングでは左右画像の比較を行うため同じ物体は同じ形状に写っている必要があるためこの補正を行う必要がある。また、平行にカメラを設置してもセンサの位置の個体差などでエピポーラ線が一致しない。歪み補正にはエピポーラ線を一致させる役割もある。

3.4.2 シェーディング補正

ステレオマッチングを輝度値を基準にして行う場合、同じ物体は左右の画像で同じ輝度値に写る必要がある。しかし、カメラでは一般にレンズの絞りの影響から画像周辺部の光量が少なくなる周辺減光と呼ばれる現象が発生する。周辺減光が発生すると

画像周辺部の輝度値が本来よりも下がってしまい、左右の画像で物体が同じ輝度に映らなくなる場合がある。周辺減光は画像周辺部、つまりカメラに対して横方向から入射する光の量がカメラ正面から入射する光の量に比べて相対的に少なくなるために発生する。また、カメラの映像センサには個体ごとに異なる画素毎の輝度ムラが存在する。この輝度ムラも左右画像の輝度の違いの原因になる。図 3.7 に輝度一様になるように撮影し、コントラストを強調した画像を示す。周辺部の輝度値が落ちていることや、画素毎に輝度ムラがあることが見て取れる。輝度一様になるように撮影した画像をフラットフレームと呼ぶ。

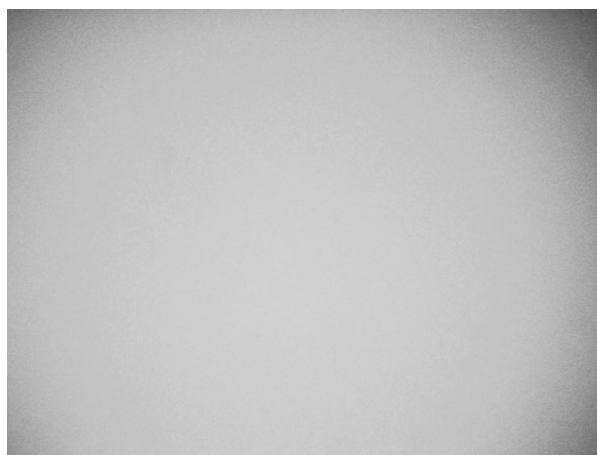


図 3.7 コントラストを強調したフラットフレーム

周辺減光や輝度ムラの影響を無くすための補正処理をシェーディング補正と呼ぶ。減光量や輝度ムラに応じて補正量を求め、テーブルとして保存しておく。撮影された画像の輝度値に補正値をかけることで周辺減光や輝度ムラを補正する。補正量の算出は先程のようなフラットフレームを用いて行う。フラットフレームは本来輝度一様に写るはずであるため、画像全体が中心付近の輝度値になるように補正値を求めれば良い。補正値は式 3.6 を用いて、基準輝度と各画素の輝度値の比率で求める。ここで、 P は基準輝度、 $I(i,j)$ は各画素の輝度値、 $T(i,j)$ は各画素の補正量である。

$$T(i,j) = \frac{P}{I(i,j)} \quad \dots(3.6)$$

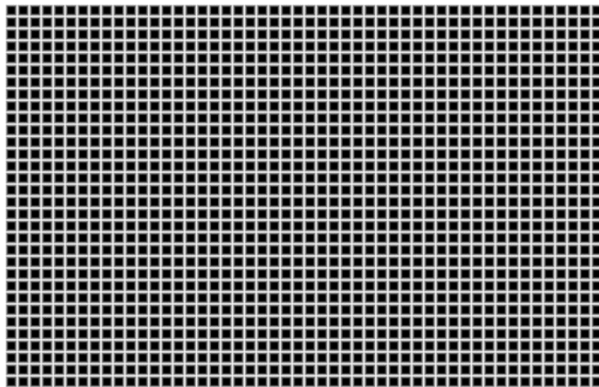
図 3.7 のフラットフレームにシェーディング補正を施し、コントラストを強調した画像を図 3.8 に示す。図 3.7 に存在するような周辺減光や輝度ムラが無くなっていることがわかる。



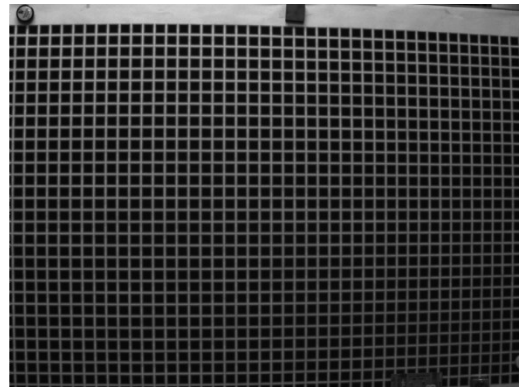
図 3.8 シェーディング補正後のフラットフレーム

3.4.3 歪み補正

カメラではレンズの歪曲収差が原因で撮影画像に歪が生じる．図 3.9(a)は一定サイズの格子が連続する画像であるが，これを撮影すると図 3.9(b)のように歪み格子の形が崩れているのが見て取れる．このような歪がある場合，左右の画像で物体が同じ形状に映らないため，ステレオマッチングができない．よってこれを補正する必要がある．



(a) 格子画像



(b) 撮影した画像

図 3.9 歪の例

レンズの歪の補正は図 3.9 に示したような格子画像を用いて行う．実際の格子のサイズは既知であるので，撮影して歪んだ格子のサイズが実際の格子のサイズになるように補正を行えばよい．

この補正処理を行うためには図 3.9(b)のような画像から一つ一つの歪んだ格子のサイズを取得する必要がある．格子のサイズは格子の交点座標がわかれば図ることができる．この，交点座標は，図 3.10(a)のようなパターンを用意し，図 3.9(b)に対してパターンマッチングをかけることで取得する．パターンマッチングは SAD を用いて行う．

パターンマッチングの結果の一部を図 3.10(b)に示す。



(a) パターン

(b) マッチング結果

図 3.10 歪み補正のためのマッチング結果

ただし、マッチング結果の座標は精度を高めるためにサブピクセル単位で求める必要がある。サブピクセル座標の算出は線形補間を持ちることで行う。線形補間では隣接がその傾きからサブピクセル座標を推測する。図 3.11 において、 x は求めるべき X 軸方向のサブピクセル座標、P2 は最も類似度が高い画素、P1、P3 はその隣接画素である。

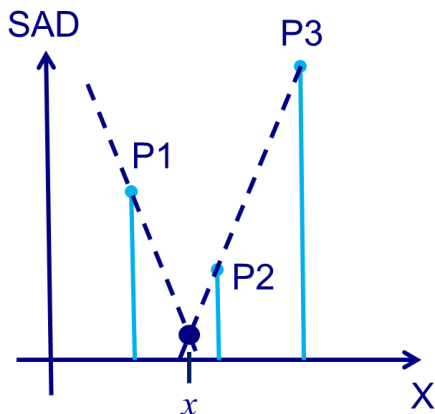


図 3.11 線形補間の例

これらの値を用いることで、式 3.7、式 3.8 からサブピクセル座標を求めることができる。ただし、3 点の中で SAD の結果の値が P3 が最大の場合は式 3.7 を、P1 が最大の場合は式 3.8 を用いる。同様の方法で Y 軸方向についても求めることができる。

$$x = \frac{P3 - P1}{2(P3 - P2)} + X_{p2} \quad \dots(3.7)$$

$$x = \frac{P1 - P3}{2(P2 - P1)} + X_{p2} \quad \dots(3.8)$$

歪み補正は補正後の画像の各画素が参照すべき元画像の座標を計算することで行う。
 図 3.12 の点 $P(x,y)$ の参照座標 $R(X,Y)$ は補正前の画像の交点座標 $p_{00}(x_{00},y_{00})$, $p_{01}(x_{01},y_{01})$, $p_{10}(x_{10},y_{10})$, $p_{11}(x_{11},y_{11})$ と格子の一辺の長さを 1 とした時の、点 $P(x,y)$ の位置の割合 W_x, W_y を用いることで式 3.9 と式 3.10 から求めることができる。

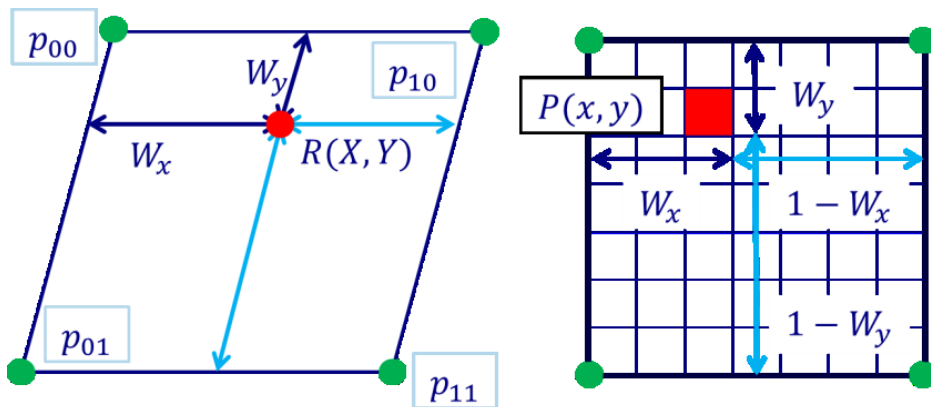


図 3.12 歪み補正の例

$$X = (1 - W_x)(1 - W_y)x_{00} + (1 - W_x)W_yx_{01} + W_x(1 - W_y)x_{10} + W_xW_yx_{11} \quad \dots (3.9)$$

$$Y = (1 - W_x)(1 - W_y)y_{00} + (1 - W_x)W_yy_{01} + W_x(1 - W_y)y_{10} + W_xW_yy_{11} \quad \dots (3.10)$$

式 3.9, 3.10 から求められる参照先座標は小数点の精度、つまりサブピクセル単位で求まる。したがって補正後の画素の輝度値は周辺画素から補間して求める必要がある。今回はバイリニア補間によって輝度値を求める。図 3.13 の点 L の輝度値は周辺画素の輝度値 l_{00} , l_{01} , l_{10} , l_{11} と座標の小数部の値 W_x, W_y を用いることで式 3.11 を用いて求めることができる。

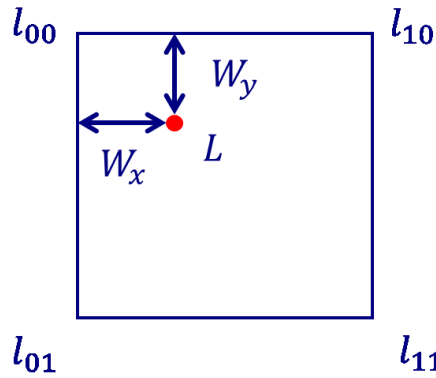


図 3.13 バイリニア補間による輝度値の算出

$$L = (1 - W_x)(1 - W_y)l_{00} + (1 - W_x)W_y l_{01} + W_x(1 - W_y)l_{10} + W_x W_y l_{11} \quad \dots (3.11)$$

図 3.9(b)の撮影した格子画像に歪み補正を行った画像を図 3.14 に示す．歪がなくなり格子が正しい形状になっているのがわかる．

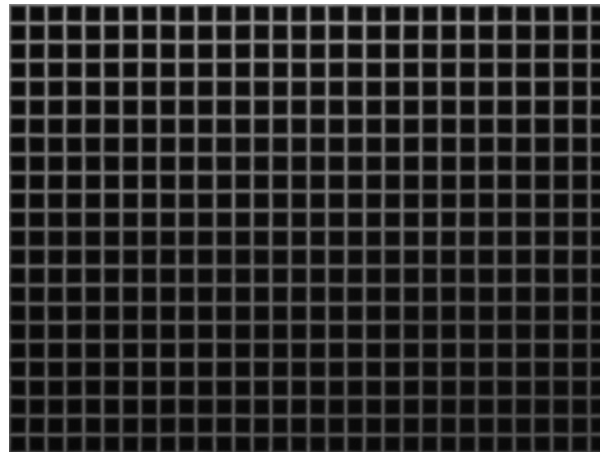


図 3.14 歪み補正後の画像

歪み補正の段階で左右のカメラのエピポーラ線を一致させる必要がある．図 3.15 はステレオカメラで講師画像を撮影した時の左右のカメラでの画像である．丸で示している交点がそれぞれのカメラでの格子の中心であるが，エピポーラ線が一致していないことがわかる．

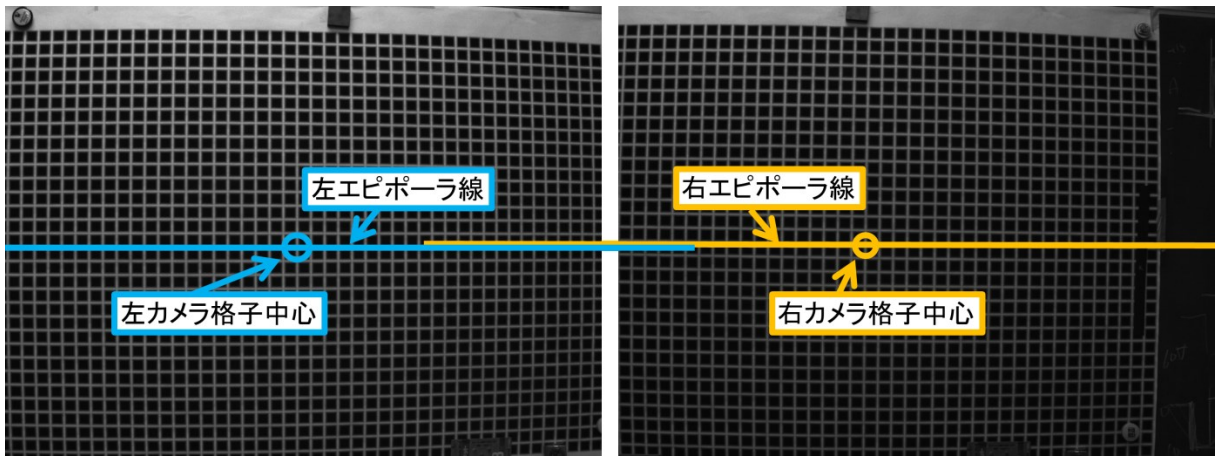


図 3.15 ステレオカメラで撮影した格子画像

このズレはそれぞれのカメラの格子の中心座標が一致するように歪み補正を行うことで解消できる。歪み補正を行ったステレオ画像を図 3.16 に示す。左右の画像のエピポーラ線が一致していることがわかる。

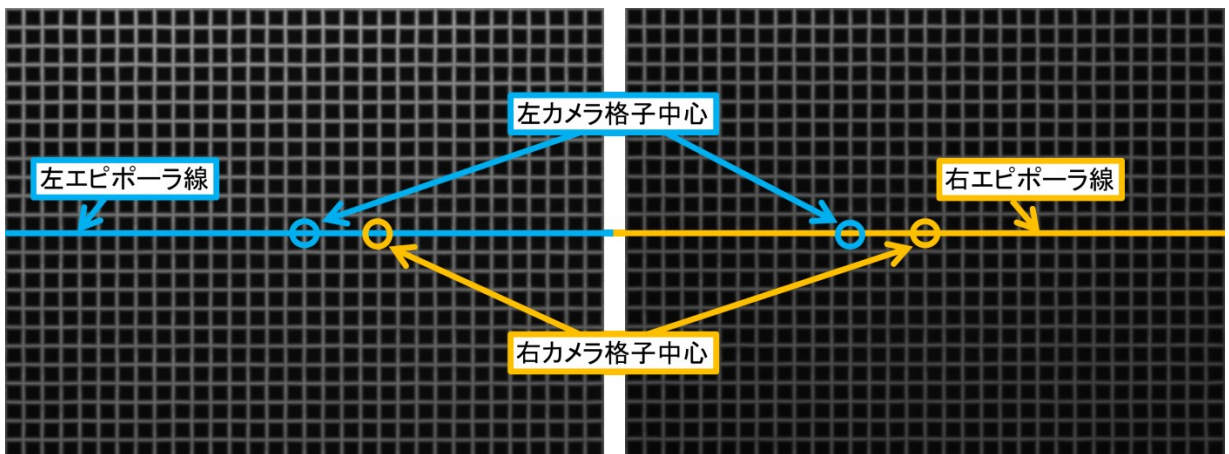


図 3.16 歪み補正後のステレオ画像

第 4 章 ボールとバンカーの検出手法

4.1 ボール検出

4.1.1 ボール検出手順

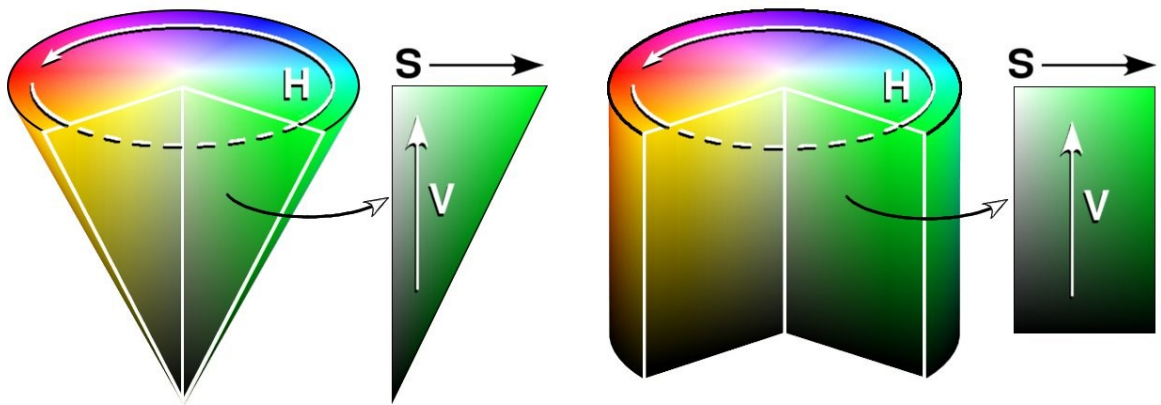
ボール検出は以下の様な手順で行う。また、今後、物体の実際の面積を実面積、画像上に写っている物体の面積を見かけの面積と呼称する。

- ① カラー画像を HSV 色空間に変換する
- ② ボールの色のピクセルのみを抽出して 2 値化する
- ③ オープニングとクロージングを行いノイズを除去する
- ④ 2 値化した画像にラベリングを行い各領域にラベル付けを行う
- ⑤ 領域ごとに視差の平均値を算出する
- ⑥ 視差から各領域の実空間上の座標を計算する
- ⑦ 各領域の距離と見かけの面積を元に実面積を算出する
- ⑧ 実面積がボールに最も近い領域をボールとして検出する

また、競技に使用するボールの色は青，黄色，赤から選ぶことができるが，我々は色抽出を行いやすくするためにフェアウェイの緑色と色相が最も遠い赤を選択した。

4.1.2 色抽出と HSV 色空間

まず，ボール検出のためにボールの色のピクセルのみを残して 2 値化を行うが，画像の色表現として一般的な RGB 色空間では同一の色でも明度や彩度の違いによって RGB3 つのパラメータが変動してしまい，特定の色を抽出するのが難しい．そこで色空間を HSV 色空間に変換して抽出を行う．HSV 色空間では，色を色相 H，彩度 S，明度 V によって表現する．彩度，明度が色相すなわち色から分離されており，色相のパラメータから色の種類を一意に同定できるため色抽出に適する．一般に HSV 色空間は図 4.1 のような円錐形や円柱形で表現される．角度が色相 H，半径が彩度 S，高さが明度 V を表す．



<http://ja.wikipedia.org/wiki/HSV%E8%89%B2%E7%A9%BA%E9%96%93>

図 4.1 HSV 色空間のモデル

RGB 色空間から HSV 色空間への変換は RGB の 3 つの値の中で最大のものの値を MAX, 最小のものを MIN と置くと HSV に関してそれぞれ式 4.1, 4.2, 4.3 で表される. ただし, MIN と MAX が同値である場合には色相 H は定義されない.

$$H = \begin{cases} \left(\frac{G - R}{MAX - MIN} \times 60 \right) + 60 & \dots \text{ if } MIN = B \\ \left(\frac{B - G}{MAX - MIN} \times 60 \right) + 180 & \dots \text{ if } MIN = R \\ \left(\frac{R - B}{MAX - MIN} \times 60 \right) + 300 & \dots \text{ if } MIN = G \end{cases} \dots(4.1)$$

$$S = MAX - MIN \quad \dots(4.2)$$

$$V = MAX \quad \dots(4.3)$$

HSV 色空間に変換した画像データの中からボールの色相を持つピクセルのみを抽出して 2 値化する. 図 4.2 の画像を 2 値化した画像を図 4.3 に示す.



図 4.2 ボール検出元画像

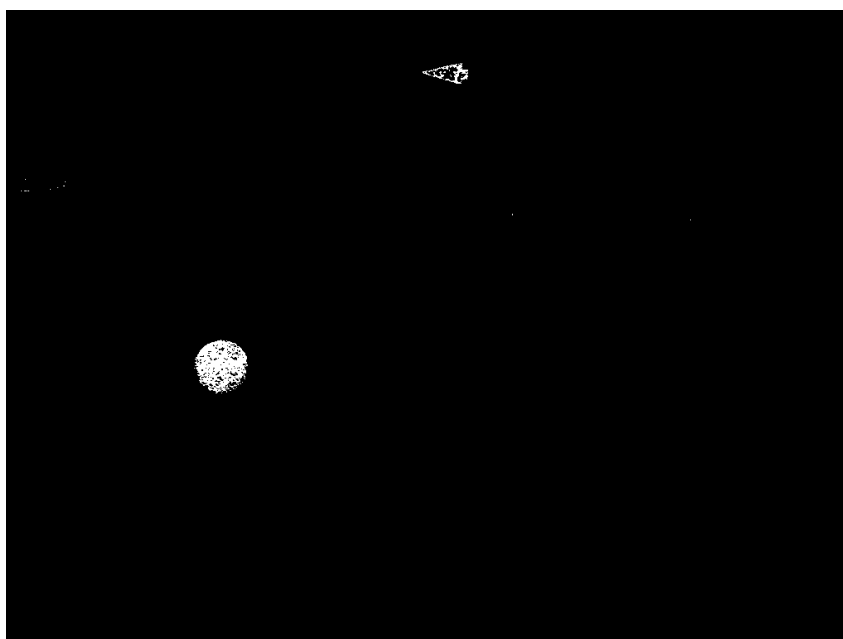


図 4.3 ボールの色相での 2 値化画像

4.1.3 ラベリング

2 値化した画像にラベリングを行う前に、オープニング、クロージングによるノイズの除去を行う。オープニングとは任意の回数縮小を行ってから、同じ回数膨張を行う処理で、クロージングはその逆に膨張を行ってから縮小を行う処理である。縮小とは、2 値画像において注目画素の周辺に一つでも黒の画素があった場合、注目画素を黒に置き換える処理である。膨張はその逆で注目画素の周辺に一つでも白の画素があった場合、注目画素を白に置き換える処理である。オープニングには細かな白のパターンを

取り除く効果があり，クロージングには細かな黒のパターンを埋める効果がある．図 4.3 にオープニングとクロージングを施した画像を図 4.4 に示す．図 4.3 に比べ細かなパターンが取り除かれ，白い画素の間が埋められており，より綺麗な 2 値化画像になっていることがわかる．

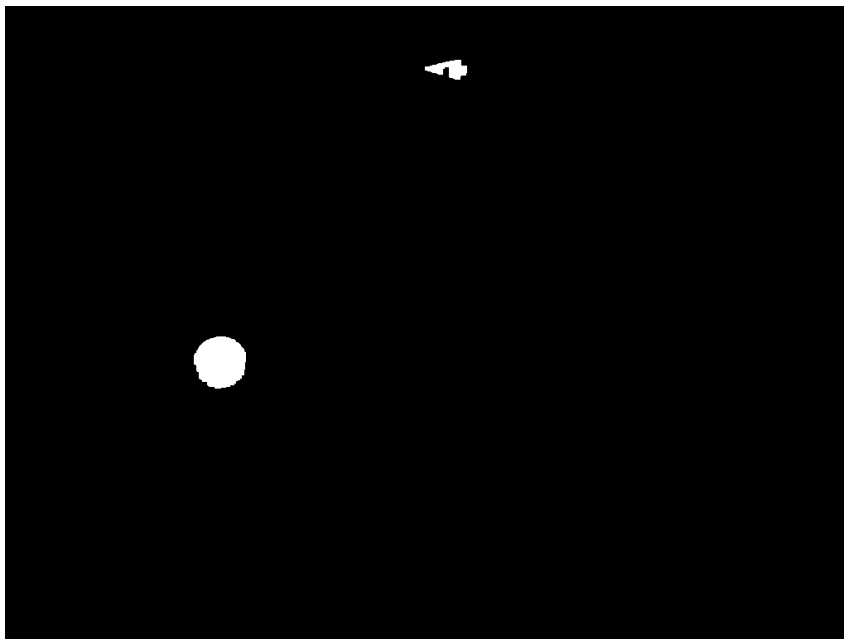


図 4.4 オープニング，クロージングを行った後の 2 値化画像

図 4.4 の画像に対してラベリングを行う．ラベリングとは 2 値画像の白い画素が連続している領域にラベル番号を割り振る処理である．ラベリングを行うことによって画像中の各領域を区別することができ，各領域の幅や面積を求めることができる．ラベリングにはいろいろな手法があるが，今回は単純な領域成長による方法を用いた．領域成長によるラベリングの手順は次のとおりである．

- ① 画像をラスタスキャンして白でかつラベル番号がふられていない画素を探索する
- ② 発見した画素にラベル番号を割り振る
- ③ 発見した画素の近傍に白の画素が無いか探索する
- ④ ②，③の処理を画素が発見されなくなるまで繰り返す
- ⑤ ラベル番号を更新し，①に戻る

図 4.4 の画像に対してラベル番号を割り振り，番号ごとに色分けした画像を図 4.5 に示す．

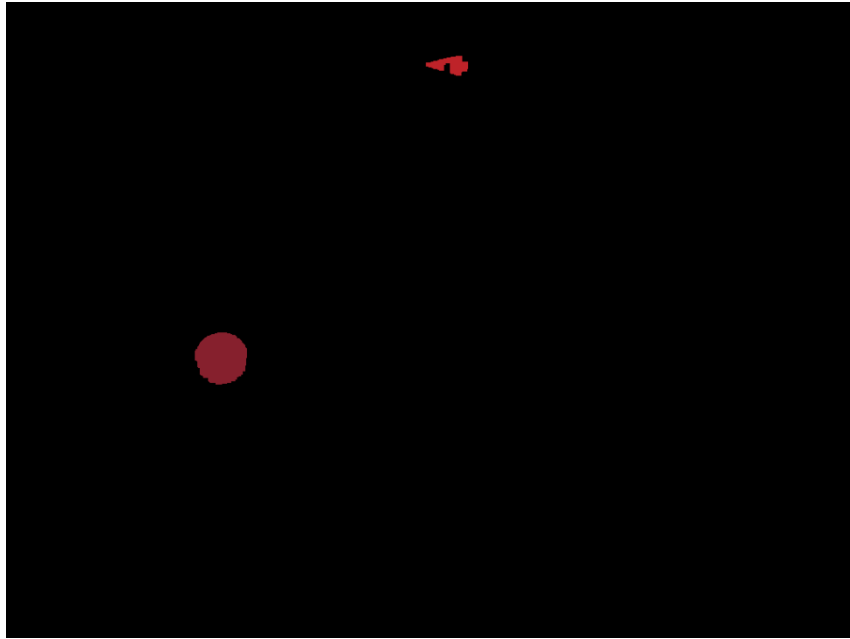


図 4.5 ラベリング結果

4.1.4 実面積を利用したボール判別

ボールのみを検出，抽出するためにはラベリング画像からどのラベル番号の物体がボールなのかを判別する必要がある．ボールの判別には円形度を用いる方法などが考えられるが，今回はステレオカメラであるので物体までの距離情報を用いることができる．物体までの距離を用いることで各ラベル番号の物体の見掛けの面積から物体の実面積を算出することができる．算出した実面積が実際のボールの実面積と最も近いラベル番号の物体をボールとして検出する．実面積を S と置くと，距離情報である視差量 d ，ステレオカメラの基線長 b ，見掛けの面積 A を用いることで式 4.4 から求めることができる．ただし，見掛けの面積 A は各ラベルのピクセル数に実際のカメラの画像素子の 1 ピクセルあたりの面積をかけたものである．

$$S = \frac{A \cdot b^2}{d^2} \quad \dots(4.4)$$

図 4.5 の中からボールと判別された物体のみを取り出した画像を図 4.6 に示す．ボールの領域のみが抽出されていることがわかる．

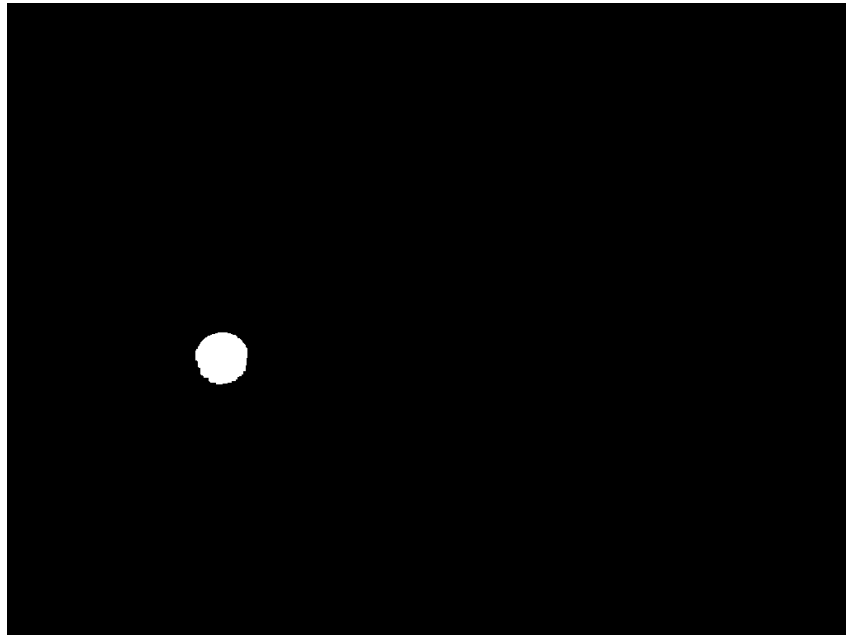


図 4.6 ボールの抽出画像

この手法を用いることで，図 4.7，図 4.8 のように見掛けの面積が近い物体が複数存在しても図 4.9 のように本物のボールのみを抽出することができる．



図 4.7 ダミーを置いた時の元画像

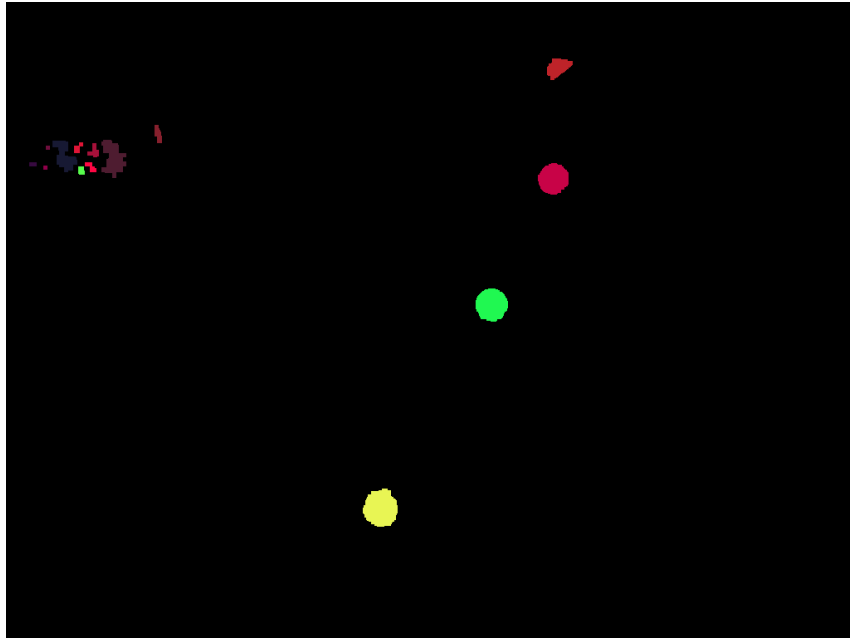


図 4.8 ダミーを置いた時のラベル画像

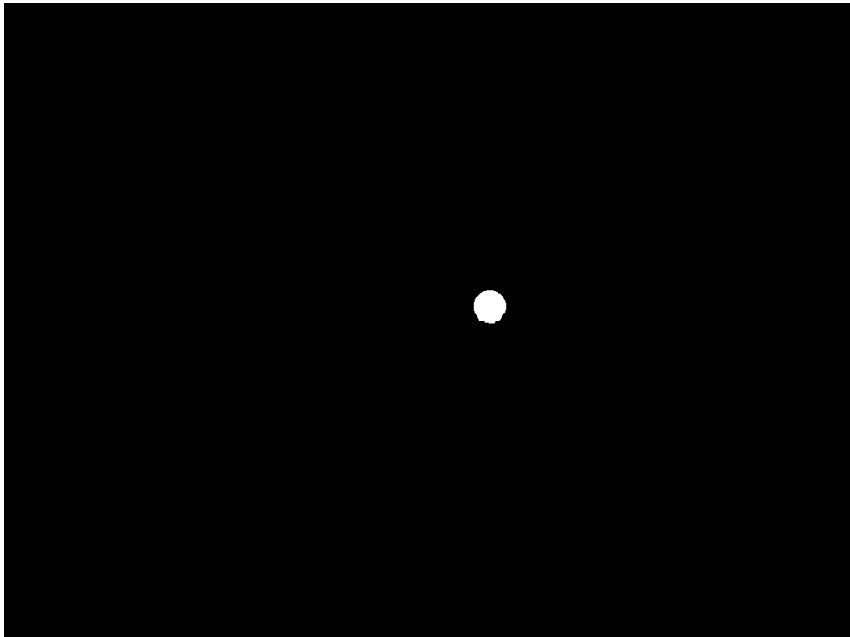


図 4.9 ダミーを置いた時の抽出結果

4.2 バンカー検出

作成したロボットはバンカーの走破性能を備えていない。また、バンカーに入ったボールをフェアウェイに打ち戻すことは困難である。このため、バンカーの検出を行って、ボールがバンカーに入らないようにバンカーを回避する必要がある。バンカー検出の手法はボール検出の手法を簡略化したものを用いる。手順は次のとおりである。

- ① カラー画像を HSV 色空間に変換する

- ② バンカーの色のピクセルのみを抽出し 2 値化する
- ③ オープニング, クロージングを行ってノイズを除去する
- ④ ラベリングを行って各領域にラベル番号を割り振る
- ⑤ しきい値を設け, 見掛けの面積が一定の範囲のラベルの領域のみをバンカーとして検出する

球体であるボールと違い, バンカーは撮影する距離や角度によって面積が変わってしまう. このため, 一意にバンカーを決定することができないため, 実面積による判別は用いなかった. 図 4.10 のような元画像を二値化し, ラベリングした画像を図 4.11 に, バンカーとして検出された領域のみを残した画像を図 4.12 に示す. バンカーのみを検出できていることがわかる.

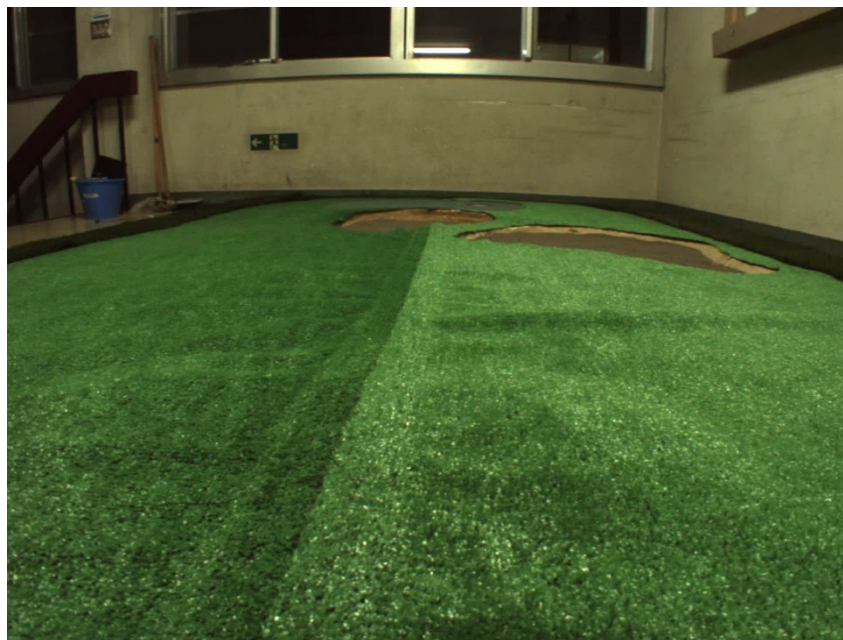


図 4.10 バンカー検出元画像

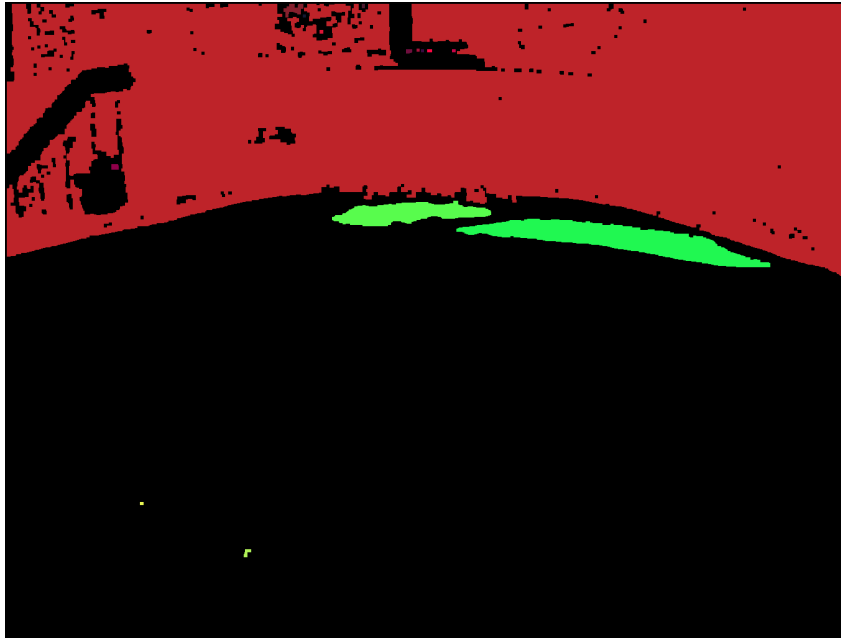


図 4.11 バンカー検出ラベル画像



図 4.12 バンカー検出結果

4.3 光源に対する対処

作成したロボットではボール検出，バンカー検出ともに色情報を用いている．しかし，色情報は光源による影響を非常に受けやすく，照明の変化によって容易くパラメータが変化してしまう．特に会場にはオレンジ色のナトリウム灯が使用されており，事前に測定した色のパラメータを用いると，会場で正常に検出が行えなくなる可能性がある．このため，我々は会場で素早くボールとバンカーの色のパラメータを調整できるように，簡単な GUI を作成した．GUI の画面を図 4.13 に示す．GUI の画面上のク

リックしたピクセルの色を検出に使用するパラメータとして利用することが可能である。

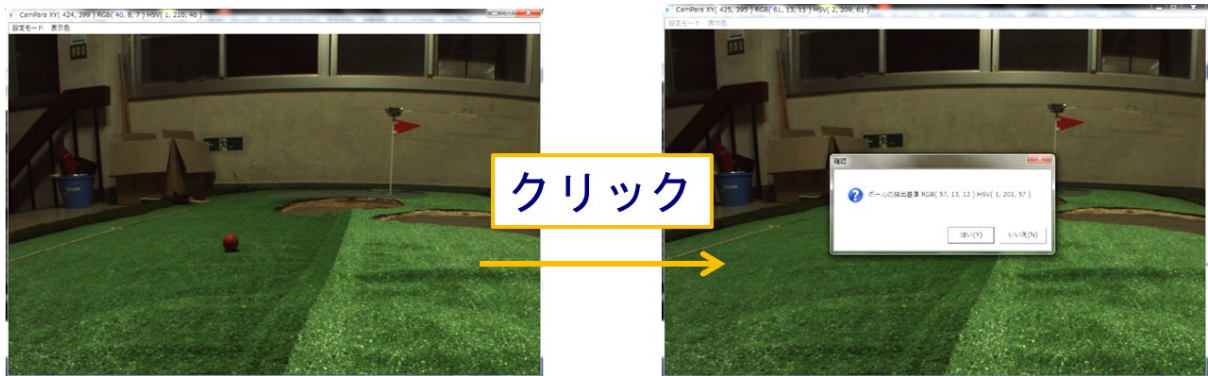


図 4.13 作成した GUI の画面

4.4 処理時間

処理時間はステレオカメラのキャリブレーション，ステレオマッチングを含めてもボール検出，バンカー検出ともに 1 秒未満である。ただし，プログラムの実装には以下の様な高速化手法を用いている。

- ・画像バッファはすべて 1 次元配列

PC のメモリは一般にバースト転送であり，ランダムアクセスよりもシーケンシャルアクセスのほうが高速である。このため，画像の保存に 2 次元配列を用いると高速化の妨げになる。

- ・極力ランダムアクセスを減らす
同様の理由により高速化につながる。

- ・アドレスの再計算を減らす

四則演算によってアクセス先のアドレスを毎回計算するのは効率が悪い。そのため，アドレスは極力インクリメント，デクリメント及び加減算によって行う。C 言語の配列に添字を使ってアクセスすることはアドレスを再計算しているのと同義なので使用を避ける。

- ・PSADBW 命令を利用したステレオマッチング

IntelMMX の PSADBW 命令をインラインアセンブラで記述して利用し，SAD の演算を SIMD 実行することで高速化を行う。

- ・ステレオマッチングを行う画素密度を下げる

ステレオマッチングを行って視差を算出する画素を単純に減らす。作成したロボットでは縦横 4 画素間隔で視差の算出を行っている。

- マルチスレッド化

近年のマルチコア CPU を有効に利用するためには必須である。

第5章 まとめ

2013年に優勝したカトちゃん号の詳細，及びステレオカメラを用いた画像処理の手法について解説した．画像処理やボール検出の手法については様々な方法が考えられるほか，距離情報や色情報を得ることができるセンサはステレオカメラ以外にも多数存在するので，各自調査してロボットに最適なものを選択して欲しい．