

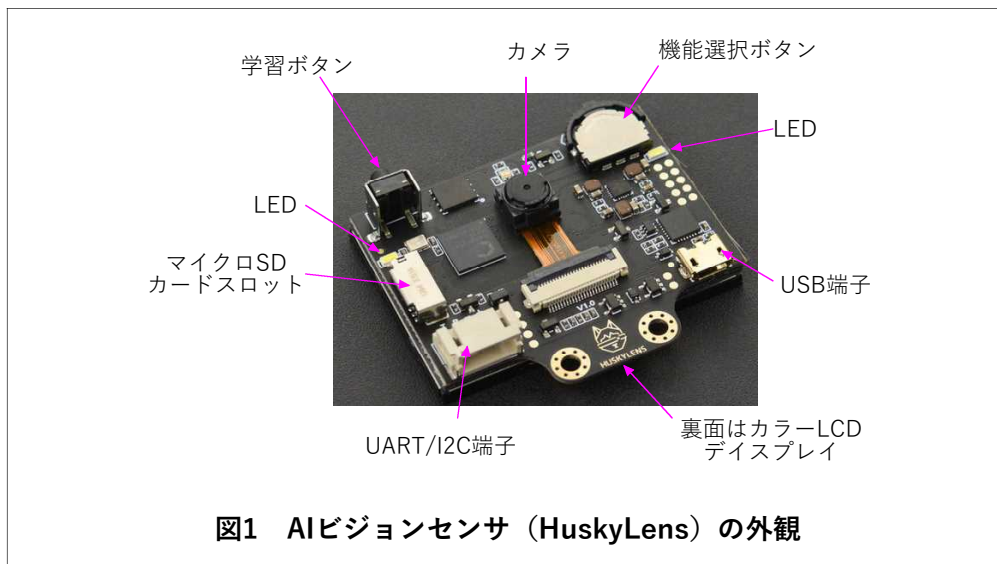
AIビジョンセンサによる自動追尾ロボットの製作

2年情報技術科 五十嵐 亜聖

指導教諭 高田 直人

1. はじめに

近年、AIビジョンセンサと呼ばれる商品が安価に入手できるようになった。AIビジョンセンサとは、カメラによって捉えた画像をアルゴリズムによって整理し、対象となる物体の認識や追跡、カメラ画像内での検出位置を演算・出力する画像処理センサである。従来なら、目的とする情報を取得するために、画像から対象物体の特徴を抽出するためのプログラムを記述しなければならなかった。しかし、AIビジョンセンサでは、カメラボードに搭載したプロセッサにAI学習機能のアルゴリズムが搭載されており、利用者は希望する対象物を登録・学習させるだけで、カメラボードが出力する結果を容易に利用できるようになる。



この研究は、自律移動ロボットに搭載する事前研究として、認証物体がつねにカメラ画像の中心に来るよう、カメラの位置を制御するシステムの研究を目標とした。研究で使用したAIビジョンセンサは、上海に本社をもつDFRobot社が開発したSTEM教育用のAIビジョンセンサHuskyLensである。このカメラボードは、図1に示すような外観をしており、AI学習機能を備えた7種類の画像認識機能を標準搭載している。2インチのカラーディスプレイを搭載しているため、パソコンと接続する必要がなく、カメラボード単体で、画像認識モードの設定や認識結果を確認することができる。

2. AIビジョンセンサ HuskyLens

2.1 HuskyLensのハードウェア仕様

HuskyLensは7種類のAI学習アルゴリズムを搭載した標準モデルと、さらにバーコー

ドと QR コード認証を追加した HuskyLens Pro の 2 種類がある。表 1 に主な仕様を示す。

表1 HuskyLensの主な仕様

使用プロセッサ	Kendryte K210
イメージセンサ	HuskyLens : OV2640 (2.0Mピクセルカメラ) HuskyLens Pro (OV5640 (5.0Mピクセルカメラ))
電源電圧	3.3V~5.0V
通信ポート	UART (TTLレベル) , I2C
ディスプレイ	2.0インチカラーディスプレイ, 320×240分解能
外形寸法	52mm×44.5mm
搭載アルゴリズム	顔認証, オブジェクト追跡, オブジェクト認識 ライン追跡, カラー認識, タグ認識 オブジェクト分類, QRコード認証(Pro), バーコード認証(Pro)

搭載しているプロセッサ Kendryte K210 は、中国の半導体ベンチャー企業が開発し、TSMC によって製造されたデバイスである。画像処理・音声処理プロセッサとマイコン機能を統合した 64bit デュアルコアプロセッサで、AI や Iot 端末向けに開発された SoC (System On Chip) である。ユーザはプロセッサの存在を意識することなく、カメラボードを単なるセンサポートとして利用することができる。外形は 52mm × 44.5mm であり、カメラボード上に搭載された操作スイッチによって認識機能の選択や設定などを行うことができる。認証結果や画像情報はシリアル通信方式で取得することができ、TTL レベルの UART と I²C の 2 種類が用意されている。シリアル通信のプロトコルは、マイコン側から送信したコマンドに対するレスポンスがカメラボードから返される、コマンド-レスポンス方式になっており、手軽に利用できるようになっている。

2.2 画像認識機能と認識情報の出力

HuskyLens は、表 2 のような 7 種類(HuskyLens Pro は 9 種類)の学習機能を備えた画像認識モードが標準で用意されている。カメラ画像に事前に学習させた認識物体や認証タグがあれば図 2 のように、学習番号(ID 番号)と、画面内における中心座標位置 (X, Y) と大きさ (W, H) をシリアルデータとして出力してくれる。



図2 カメラ画面の座標と認識情報

表2 HuskyLensの認識機能の概要とデータ出力

認識機能	カメラ画像	機能の概要	データ出力
顔認識		特定の(1人)顔だけを学習させ、複数の顔の中から認識するone faceモードと複数の顔を認証するmultiple faceモードがある。写真でも使用可能。	顔認識した中心座標(X,Y)と認識画像の大きさ(W,H), ID番号
オブジェクト追従		指定されたオブジェクト(物体)を学習し、画面内で追跡する。ただし複数のオブジェクト追跡はできない。	認識した物体の中心座標(X,Y)と認識画像の大きさ(W,H), ID番号
オブジェクト認識		物体が何であるかを認識し、登録した物体を追跡する。HuskyLensには20種類の物体が登録済みである。(飛行機, 自転車, 鳥, ボート, ボトル, バス, 車, 猫, イス, 牛, テーブル, 犬, 馬, バイク, 人, 植木鉢, 羊, ソファ, 電車, テレビ)	認識した物体の中心座標(X,Y)と認識画像の大きさ(W,H), ID番号
ライン追従		指定された色の線(ライン)を認識し経路予測(ラインの方向ベクトル)を表示する。	方向ベクトルの始点(xo,yo)と終点(xt,yt)座標, 認識したラインの色のID番号
カラー認識		特定(1つ)の色を学習させると、その色の物体だけを認識・追跡する。	認識した物体の中心座標(X,Y)と認識画像の大きさ(W,H), カラーID番号
タグ認識		DFRobot社が提供するタグを認識し、登録したタグだけを識別する。使用できるタグは現在のところ587種類。複数タグの中から特定のタグだけを認識し、追跡する。	タグであることを認識した全ての中心座標(X,Y)と認識画像の大きさ(W,H), ID番号
オブジェクト分類		様々なオブジェクトの写真や絵を学習し、カメラに組み込まれた機械学習アルゴリズムをトレーニングできる。トレーニングが完了した後は、そのオブジェクトを認識し、ID番号を表示する。	認識した中心座標(X,Y)と認識画像の大きさ(W,H), ID番号

3. マイコンとのインタフェースプロトコル

HuskyLens は、マイコンとのインタフェースとして、UART と I²C の 2 種類の通信方式が用意されている。ここでは、UART によるインタフェースについて述べる。

3.1 通信速度

通信速度は自動検出、9600bps、115200bps があり、自動検出方式の使用を推奨している。本報告では、9600bps の通信速度を利用している。

3.2 コマンドとレスポンスのデータフォーマット

HuskyLens に対するコマンドはさまざまな種類があるが、主に使用するのは、接続確認 (COMMAND_REQUEST_KNOCK) とブロックデータの要求コマンド (COMMAND_REQUEST_BLOCKS) の 2 種類である。

表3 接続確認コマンドとその応答

	コマンド/レスポンス名	内容	Header	Header2	Address	データ長	コマンド	チェックサム
①	COMMAND_REQUEST_KNOCK	接続確認コマンド	0x55	0xAA	0x11	0x00	0x2C	0x3C
②	COMMAND_RETURN_OK	接続確認応答	0x55	0xAA	0x11	0x00	0x2E	0x3E

(1) **接続確認** マイコンとの接続を確認するためのコマンドは、表 3 の①のデータを HuskyLens に送信する。データフォーマットは、2 バイトのヘッダ(0x55, 0xAA)とコマンドの書き込みアドレス(0x11)、コマンドの後に続くデータ長、コマンド、チェックサムである。HuskyLens が①のコマンドを正常に受け取ると、レスポンスとして表 3 の②のデータ(6 バイト)を送り返してくるので、接続しているか否かを確認することができる。

(2) **認識情報の取得** HuskyLens による認識情報の取得には、表 4 のような 6 バイトで構成されるブロックデータの送信要求コマンドをマイコンから送信する。

表4 ブロックデータの送信要求コマンド

	コマンド名	内容	Header	Header2	Address	データ長	コマンド	チェックサム
①	COMMAND_REQUEST_BLOCKS	ブロックデータの送信要求	0x55	0xAA	0x11	0x00	0x21	0x31

HuskyLens 側からは、16 バイトで構成される表 5 のようなインフォメーションデータが返信される。インフォメーションデータは各データが 2 バイトで構成され、認識した物体の数、認証 ID 番号などが通知される。さらに、認識物体がある場合は、インフォメーションデータに続いて表 6 のような 16 バイトで構成されるブロックデータが返信される。

表5 インフォメーションデータ

コマンド名	Header	Header2	Address	データ長	コマンド	データ										チェックサム
						L	H	L	H	L	H	リザーブデータ				
COMMAND_RETURN_INFO	0x55	0xAA	0x11	0x0A	0x29	0x01	0x00	0x01	0x00	0x05	0x00	0x00	0x00	0x00	0x00	0x4A
						認識したブロックまたは矢印の数		画像から認識した物体の ID 番号		フレーム番号		リザーブデータ				

ブロックデータは、カメラ画像において認識した物体の中心座標(X,Y)と大きさ(W,H),

認識した ID 番号で構成され、各データはそれぞれ 2 バイトの整数値として得ることができる。

表6 ブロックデータ

コマンド名	Header	Header2	Address	データ長	コマンド	データ										チェックサム
						L	H	L	H	L	H	L	H	L	H	
COMMAND_RETURN_BLOCK	0x55	0xAA	0x11	0x0A	0x2A	0x2C	0x01	0xC8	0x00	0x0A	0x00	0x14	0x00	0x01	0x00	0x58
						認識したブロックの中心座標 (X)		認識したブロックの中心座標 (Y)		認識したブロックの大きさ (W:幅)		認識したブロックの大きさ (H:高さ)		認識したID番号		

4. 目標物の自動追尾ロボットの構成

物体認識機能を活用し、目標物が常にカメラ画像の中心になるように、水平(Pan)方向サーボと垂直(Tilt)方向サーボを制御する自動追尾ロボットのハードウェア構成を図3に示す。

LFTK400SH はルネサスの SH7125 をコアにして、マイコンボード上に DC モータドライバ、RC サーボ、ロータリエンコーダ入力、シリアル通信端子、A/D 入力端子などを備えており株式会社エル・アンド・エフ (<https://www.l-and-f.co.jp/>)

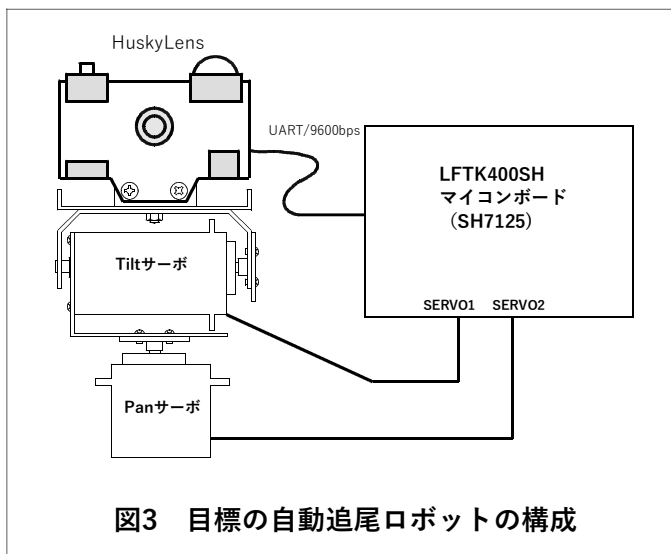


図3 目標の自動追尾ロボットの構成

より購入することができる。目標物を認識すると、カメラ画像における物体の中心座標の情報を入手できる。物体の中心座標が、カメラ画像の中心になるようフィードバック制御システムによって RC サーボを制御すれば、目標物の追尾システムを構築することができる。フィードバック制御システムは図4のように構築し、制御方式として PI 制御(比例・積分制御)を用いている。制御系に積分動作を用いている理由は、制御偏差 e が 0 になるまで積分動作による操作量が発生するため、目標値とのずれである定常偏差を 0 にできるからである。HuskyLens の認識情報は、リクエストコマンド関数 `cmd_request_block()`、

インフォメーションデータの取得関数 `get_info()`、ブロックデータの取得関数 `get_block()` を作成し、SCI(シリアルコミュニケーション インタフェース)の送信・受信割り込みを使って取得した。画像情報から

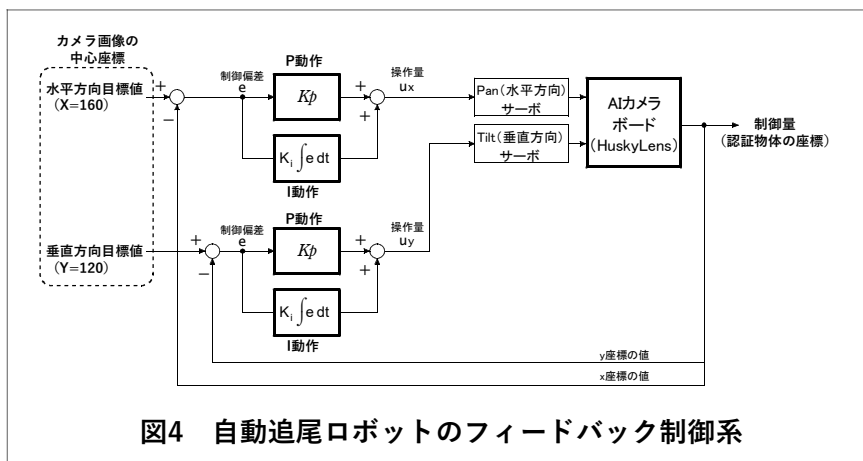


図4 自動追尾ロボットのフィードバック制御系

得た認識物体の座標情報から、RC サーボを制御する操作量の演算処理をリスト1に示す。

カメラ画像の情報から、RC サーボの操作量を計算する関数 `trace_ctrl()` は、約 50ms のサンプリング時間ごとに実行し、認証物体が常にカメラ画像の中心に来るように制御することができた。また、カメラの視野から認識物体がなくなった(認識できない)場合は、RC サーボの動作を停止するようにしている。

リスト1 PI制御によるフィードバック制御関数

```
void trace_ctrl(void) { //--- PI制御器によるターゲット追従制御
  int tmp;
  cmd_request_block(); // ブロックデータのリクエスト
  tmp = get_info(); // インフォメーション情報の取得
  if(tmp == 0 || info.id == 0) { } // 学習IDターゲットがない場合はその場で停止
  else {
    get_block(tmp); // ブロックデータの取得
    if(block.id == 1) { // ID=1のターゲットを見つけたら以下を実行
      //---Pan(水平)方向の制御
      pos_x = (short)(block.x - CENTER_X); // 水平方向の目標pixelからの偏差
      if(flag_x==0) sum_x += pos_x; // 積分動作
      u_x = (short)(pos_x * PGAIN_X + sum_x * IGAIN_X); // PI制御による操作量の計算
      if((u_x > 600) || (u_x < -600)) flag_x=1; // RCサーボの操作範囲外なら積分しない
      else flag_x = 0;
      set_servo2(u_x); // Pan用サーボの操作
      //---Tilt(垂直)方向の制御
      pos_y = (short)(block.y - CENTER_Y); // 垂直方向の目標pixelからの偏差
      if(flag_y==0) sum_y += pos_y; // 積分動作
      u_y = (short)(pos_y * PGAIN_Y + sum_y * IGAIN_Y); // PI制御による操作量の計算
      if((u_y>600) || (u_y<-600)) flag_y=1; // RCサーボの操作範囲外なら積分しない
      else flag_y = 0;
      set_servo1(u_y); // Tilt用サーボの操作
    } // end of if
  } // end of else
}
```

5. おわりに

AI ビジョンセンサ HuskyLens を活用し、事前に学習させた物体を認識し、認識物体がカメラ画像の中心になるよう制御する自動追尾ロボットの製作を行った。HuskyLens は、物体認証機能だけでなく、色認証や顔認証、ライン認証など、多くの認証モードが用意されているので、様々な分野での活用が期待できる。

最後に、研究助成にあたり尽力いただいた関係者の皆様に感謝を申し上げます。

参考文献

- [1] HuskyLens 説明書 : <https://www.dfrobot.com/product-1922.html>
- [2] Protocol Document : <https://github.com/HuskyLens/HUSKYLENSArduino/blob/master/HUSKYLENS%20Protocol.md>