

RoboCup SSL Humanoid のためのカラーボクセルサーバの提案と応用

A Proposal and its Application of Color Voxel Server for RoboCup SSL Humanoid

植田康生 升谷 保博

Yasuo UEDA Yasuhiro MASTANI

大阪電気通信大学

Osaka Electro-Communication University

Abstract

In this paper, the authors propose “color voxel server” as a shared vision system for RoboCup SSL Humanoid, which generates coloring voxel data of 3D space on a field from images taken with multiple external cameras and supplies it via network. Kimura proposed an original “voxel server” in 2010, which is insufficient to distinguish objects. Because it generates binary voxel data. Therefore the proposed server gives color information to voxel without spoiling realtime performance by not using full color and reducing number of colors in the camera images. Seitz’s method of voxel coloring is used to solve color of voxel. It takes about 80[ms] for a prototype program to generate 4-color 10[mm]-cubic voxels of 4050[mm]×3025[mm]×500[mm] space from 8 camera images and supply compressed data via network. Moreover the authors show that it is possible to distinguish objects and decide action of robot based on color voxel data provided from the server.

1 はじめに

RoboCup サッカーの小型ロボットリーグ (Small Size Robot League, 以下 SSL)[長坂, 2011]とそのサブリーグである SSL Humanoid[升谷, 2010]は, 他のリーグと異なり, ロボット自身にはカメラが搭載されておらず, 外部カメラを用いてフィールド上のロボットやボールの位置を認識しロボットを制御している.

現状では, Figure 1 のようにフィールド上空のカメラでロボットの頭頂部に取り付けたマーカ板のパターンを

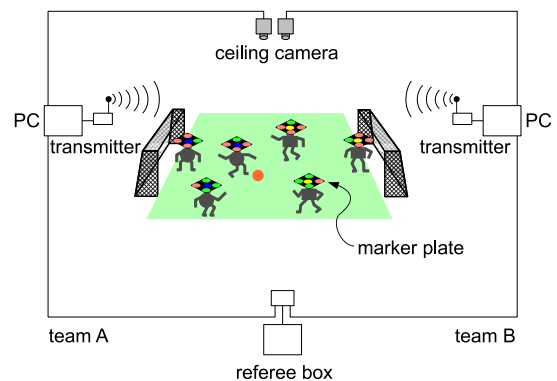


Figure 1: Initial phase of the SSL Humanoid (2009)

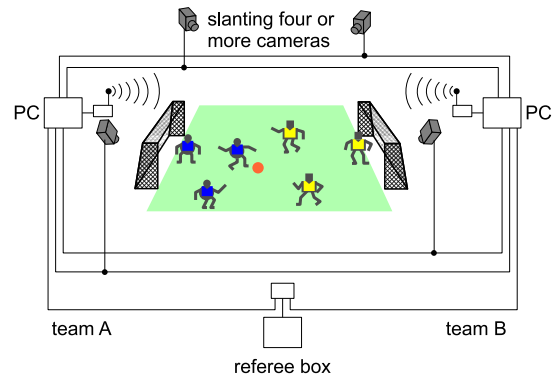


Figure 2: Final phase of the SSL Humanoid (2015)

撮影し, 2次元的な画像処理を経て, その番号や位置, 方向を算出している. しかし, SSL Humanoid のロードマップでは, Figure 2 のように 2015 年頃には複数の外部カメラで撮影したカメラに基づいて, 複数のロボットの 3次元的な形状を取得し, それに基づきロボットを制御することを目指している.

SSL では, 2010 年から視覚系の共有が義務付けられた. 参加チームは, カメラとその画像を処理する SSL-Vision と呼ばれるサーバプログラムを共有し, SSL-Vision から送られてくるフィールド上のオブジェクトの位置情報を利

Table 1: Identification by color of voxel

Voxel	Mode of expression
Nothing	0
Other	1
Orange	2
Blue	3
Yellow	4

用する．SSL Humanoid でも視覚系の共有は必須課題である．現状では，SSL-Vision を利用しているが，3次元の場合に何をどのように共有するかはまだ決まっていない．

これを受けて，木村らはフィールド上の3次元空間のボクセルデータを共有することを提案し，それを具体化する「ボクセルサーバ」を開発した[木村, 2010]．このサーバは，8台のカメラを用い，3025[mm]×4050[mm]×500[mm]の領域を10[mm]立方のボクセルで表現し，それをネットワークを介してクライアントへ配信している．また，クライアント側では，ボクセルデータからロボットのポーズ推定を行い，その結果からロボットの行動決定ができることを示している[植田, 2011]．しかし，木村の手法では，ボクセルデータは，個々のボクセルが空間を占めるか否かの2値の集合である．そのため，複数のロボットを区別することが難しい．また，ロボットと接触しているボールを別のオブジェクトとして識別できないという問題もある．

そこで，木村の方法を拡張し，複数のカメラ画像から色付けしたボクセルデータを算出し，それをネットワーク経由で配信する「カラーボクセルサーバ」を提案する．本稿では，その仕様や実現方法について述べ，実装したプログラムを評価する．さらに，サーバから提供されたカラーボクセルデータに基づいて，オブジェクトの識別とロボットの行動決定を行い，提案手法の有効性を検証する．

2 カラーボクセルサーバ

この節では，本稿で提案する「カラーボクセルサーバ」について述べる．カメラ画像の色情報に基づいてボクセルデータを多値化する．各ボクセルの色は，写実的に表現するのではなく，色数を数bitに留める．ここでは，Table 1に示す5値を用いる．これは，最低限の要件として，ボクセルなし，ボール，2つのチームを表す色，それ以外の色が表現することができれば良いからである．また，サーバ側では，多値化したボクセルデータを生成しクライアントへ送信するのみであり，色によるオブジェクトの識別などは行わない．これは，サーバ側で各オブジェクトの識別を行うと，処理時間がかかってしまい実時間での配信が難しくなるためである．また，このようにすることでボクセルデータの利用側で工夫や研究を行う余地ができる．

2.1 カメラの撮影対象

本提案では，カメラで撮影する対象に明確な色が付いていることを前提とする．オレンジ色のボールや，区別のつきやすい色のユニフォームのような物を装着しているロボット等である．ただし，ロボットにおいては全体が同じ色で覆われている必要はなく，一部で構わない．

2.1.1 複数色のシルエット画像生成

提案手法では，処理を減らすために，カメラ画像の段階で色数を減らし，必要な色だけを抽出する．カメラ画像の画素値と背景画像の画素値の差分を取り，差があった画素のみに対して色抽出を行う．画素値をHSVで表現し各成分が設定の範囲内であれば所定の色とする．もし，どの色の範囲内でもなければ，シルエット画像の画素値は白とする．カメラ画像と背景画像の差分がなければ，シルエット画像の画素値は黒となる．

2.1.2 ボクセルの生成

シルエット画像に基づき，視体積交差法的一种であるSpace Carving Method(SCM)によって，ボクセルデータを生成する．SCMは，シルエット制約に基づいて，オブジェクトの内点を残して，それ以外の点を削り落とす(ボクセルカットと呼ぶ)手法である．高速化のために，領域を分割しマルチスレッドで並列処理する．

2.1.3 ボクセルの色付け

色付けにはVoxel Coloring手法[M.Seiz, 1997]を用いる．複数の入力画像の色情報を用いて，3次元復元を行う手法である．ここでは，z軸と垂直な層を考え，カメラに近い層から順に処理を行う．各ボクセルと対応するシルエット画像の画素値をカメラ台数分取得する．取得する際に，すでに投影されているかどうかを示す可視判定マスクが1であれば，そのカメラの画素値は取得しない．取得した画素値を比較し，全てが同じ色であれば，ボクセルの色付けを行う．一つでも異なる色があれば，そのボクセルを削除する．しかし，異なる色が白の場合にはボクセルを削除しない．一つの層の処理が終了したら，取得した画素に対応する可視判定マスクを1に変更する．

2.2 通信プロトコル

クライアントへのデータ転送にはUDPマルチキャストを用いる．送信する内容はボクセルの1辺の長さ[mm]，ボクセル空間の3辺の長さ[mm]，ランレングス圧縮後の総データ量[byte]，ランレングス圧縮後のデータである．ランレングス圧縮では1つのランを2byteで表現する．上位3[bit]をラン値(ボクセルの値)，以降の下位13[bit]をラン長とする．

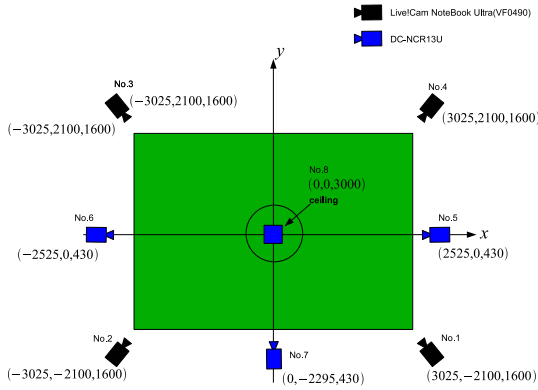


Figure 3: Camera layout

Table 2: Specifications of the experimental PCs

Server PC	
OS	Windows 7 Professional 64bit
Main Mamory	5GByte
CPU	Intel Core i7 950(3.19GHz)
Graphic Board	NVIDIA GeForce GTX260
Client PC	
OS	Windows 7 Professional 64bit
Main Mamory	4GByte
CPU	Intel Core i7 X940(2.13GHz)
Graphic Board	ATI Mobility Radeon HD 5800 Series



Figure 4: Camera images

2.3 実行例

提案するカラーボクセルサーバを実装したプログラムの実行例を示す．プログラムを実行する PC の仕様を Table 2 に示す．カメラの配置を Figure 3 に示す．Voxel Coloring に使用するカメラは，Figure 3 において黒色で表されている 4 台である．これらは，撮影対象の空間よりも上部に設置されており， z 軸と垂直な同じ平面上に設置されている．

フィールド上に青チームのロボットと黄色チームのロボットを 3 体ずつ配置する．また，ボールをフィールドの中央に置く．取得したカメラ画像が Figure 4，生成されたシルエット画像が Figure 5，それらから生成されたボクセルデータが Figure 6, 7 である．Figure 5 では差分のある画素のうちオレンジ，黄色，青色以外は黒で表し，差分のない画素は緑で表している．カラー化する前のボクセルサーバとの実行周期の比較を Figure 8 に示す．

この実行例から，オレンジ，黄，青，それ以外の 4 色でボクセルを表現できていることがわかる．しかし，多値

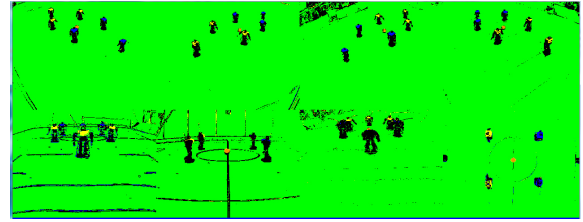


Figure 5: Silhouette images

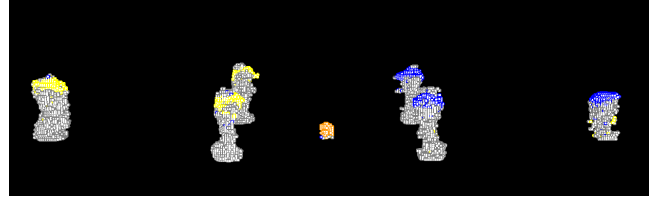
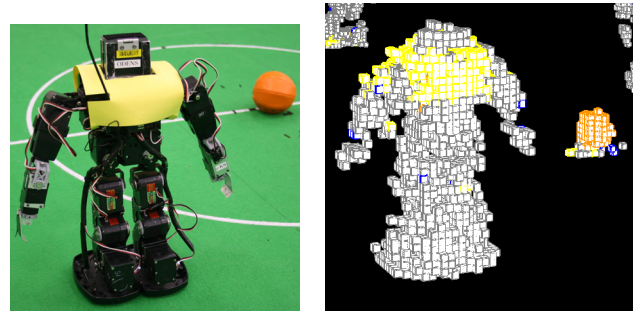


Figure 6: Generated voxel data



(a) Real image

(b) Voxel atea

Figure 7: Camera image and voxel data

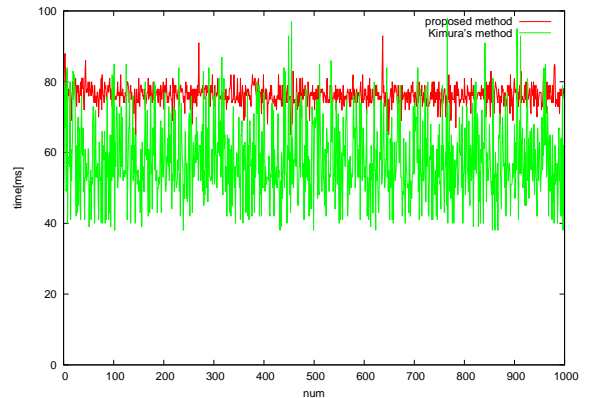


Figure 8: Comparison of the processing time of the proposed method and Kimura's method

化処理を行っているので処理時間が約 34%増加している．

3 オブジェクト識別

この節では，カラーボクセルデータに基づくボールやロボットの識別について述べる．

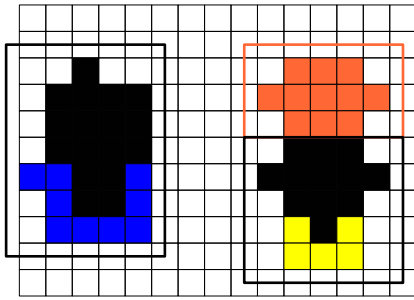


Figure 9: Two types of labeling

3.1 多値ラベリング

ボールのように、全体が1色で覆われているオブジェクトの色と、それ以外の色の2種類に分けてラベリングを行う。Figure 9のようにボールを覆っているオレンジ色の場合は、その色のボクセルだけでラベリングを行い、それ以外の色のボクセルはまとめてラベリングを行う。これにより、ボールがロボットと接触していても、同一のボクセル群にならず、ボールを認識することができる。

3.2 ボール・チーム識別

オレンジの値のボクセルのみでラベリングしたボクセル群の中からボールに近い体積のものをボールと認識する。Voxel Coloringでは表面のみ色付けされるため、ボールの内部にオレンジではないボクセルが存在するが、そのサイズが小さいため、影響はない。

ロボットに対しては、ラベリングされたボクセル群に含まれる青と黄のボクセルを数え、多い方の色のチームに決定する。

3.3 号機判定

停止状態の各ロボットにそれぞれ別のコマンドを送信する。一定時間経過後、ボクセル群の初期位置と現在位置の関係から、対応するロボットの行動を推定する。それを実際に送ったコマンドを比較し、号機番号を判定する。

3.4 ロボットの動作の監視

コマンドと動作の対応の監視を常に行う。ロボットに以前と異なるコマンドを送る際に、そのロボットの位置を動作の開始位置とする。次回以降の処理では、ロボットの開始位置と現在位置との位置関係から行動通り動作した際の想定した位置関係になっているか監視する。所定の回数連続で想定される位置関係でないと判定した場合、誤判定として検知する。

3.5 号機の再判定

号機の誤判定を起こした際に、再度号機判定を行う。また、異常行動の原因の一つとして、ロボットの方向を180[deg]誤認識している場合がある。この場合、例えば、後進コマンドを送ると、前進行動と認識される。このような場合

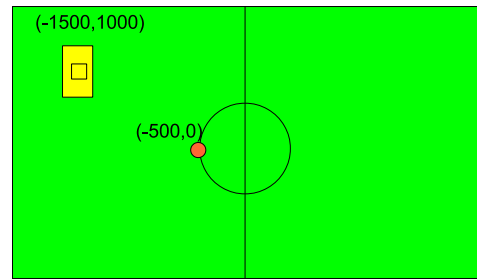


Figure 10: Initial positions of robot and ball

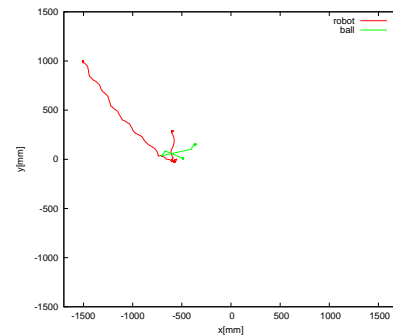


Figure 11: Trajectories of robot and ball by Kimura's method

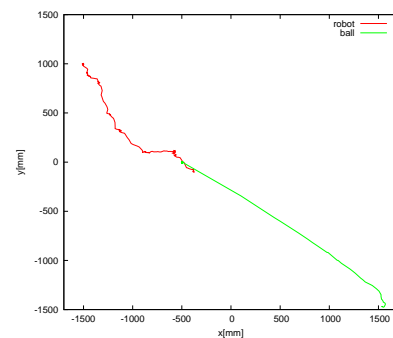


Figure 12: Trajectories of robot and ball by the proposed method

は、そのボクセル群を後進コマンドを送信したロボットの号機と判定し、方向の修正を行う。

4 評価実験

この節では、カラーボクセルデータに基づくオブジェクト識別の評価実験について述べる。クライアントPCの仕様をTable 2に示す。サーバ側では多値化したボクセルデータを生成し、クライアントに送信する。クライアントでは、受信したデータを基に行動決定を行う。

4.1 ボールキック行動の比較実験

提案手法を木村の手法と比べて検証を行う。そのための課題としてボールキック行動を用いる。ボールから離れた位置にいるロボットにボールをフィールドの中央に向けて蹴るように行動させる。木村の手法では2値のボクセル群の中から、体積がボールに近いものをそれと認識して

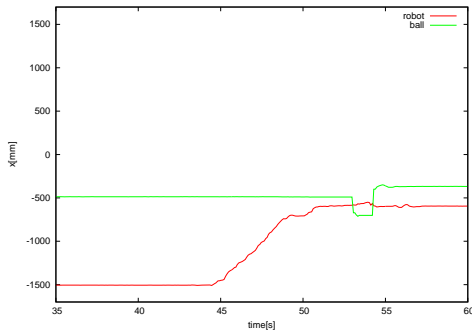


Figure 13: X coordinate of robot and ball by Kimura's method

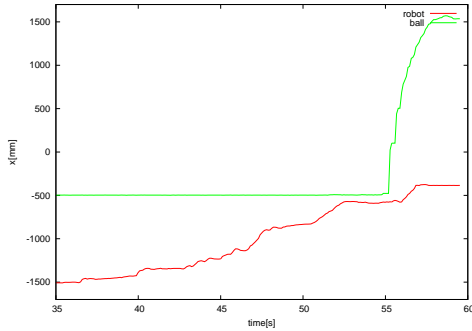


Figure 14: X coordinate of robot and ball by the proposed method

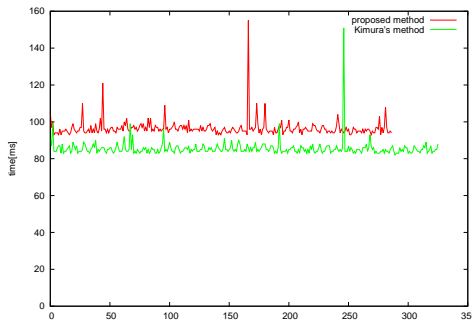


Figure 15: Comparison of the processing time of the proposed method and Kimura's method by client

いる．ロボットとボールの初期位置を Figure 10 に示す．

木村の手法によるロボットとボールの軌跡を Figure 11 に．x 座標の時間変化を Figure 13 に示す．提案手法によるロボットとボールの軌跡を Figure 12 に，x 座標の時間変化を Figure 14 に示す．両手法の処理時間を Figure 15 に示す．

木村の手法では，ボールに最も接近した 53[s] において，ロボットとボールが接触し，ボールとは別のボクセル群が誤ってボールと認識されているため，うまく蹴ることができていない．一方，提案手法では，ロボットがボールに接触しても，ボールを認識できており，ボールをうまく蹴ることができている．この結果から，木村の手法ではボールを認識できていない場面でも提案手法では問題な

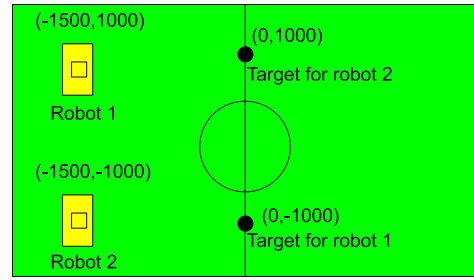


Figure 16: Initial and target positions of robots

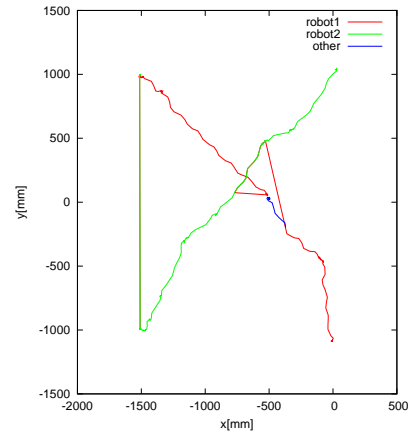


Figure 17: Trajectories of robots

いことがわかる．

また，処理時間を比較すると，提案手法の方が 1 回あたり約 10[ms] 長い．これは，ラベリング処理を 2 回行っているためとロボットの動作の監視を追加したためである．また，誤って存在しないロボットを認識している場合があり，そのロボットのポーズ推定も行うため，余分に処理時間がかかっている場合がある．

4.2 ロボットの号機判定実験

プログラム開始時や実行中にロボットの号機を正しく判定できているか検証する．フィールド上のロボットの初期位置と目標位置を Figure 16 に示す．同じユニフォームカラーの 2 台のロボットが交差するような移動経路をとるように目標位置を設定している．ここでは，2 台のロボットへのコマンド送信は 1 個のプログラムで行う．

各ロボットの軌跡を Figure 17 に示す．また，x 座標と y 座標の時間変化をそれぞれ Figure 18 と 19 に示す．また，ロボットの方向の時間変化を Figure 20 に示す．これらの図中の黒線は号機の再判定を行っているかを表し，紫線と水色線はそれぞれ号機の動作の監視結果 (ok または error) を表す．

Figure 19 によれば，24[s] に 1 号機と 2 号機の位置が入れ替わっているのがわかる．Figure 20 を見ると，1 台のみ角度が変化している．これは号機判断を行うために 1 号機に旋回行動を行わせているからである．その結果が

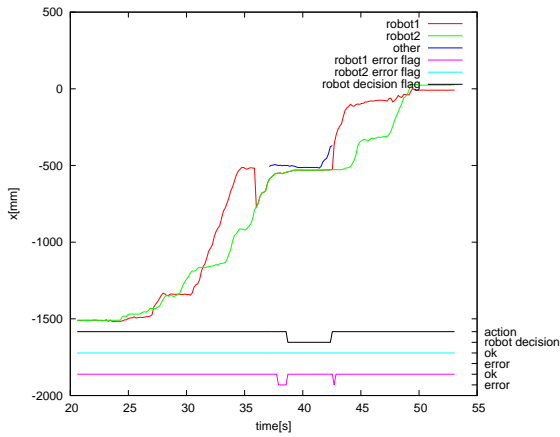


Figure 18: X coordinate vs. time of robot 1, 2, and other

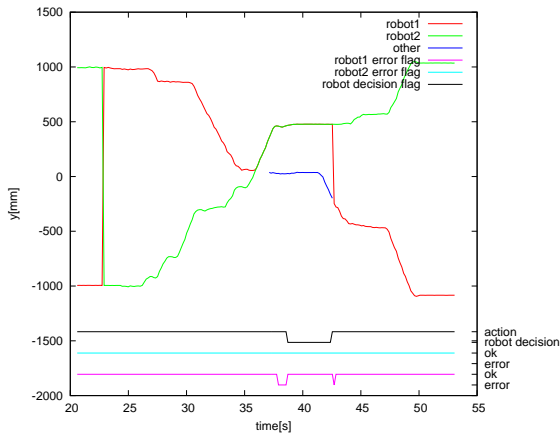


Figure 19: Y coordinate vs. time of robot 1, 2, and other

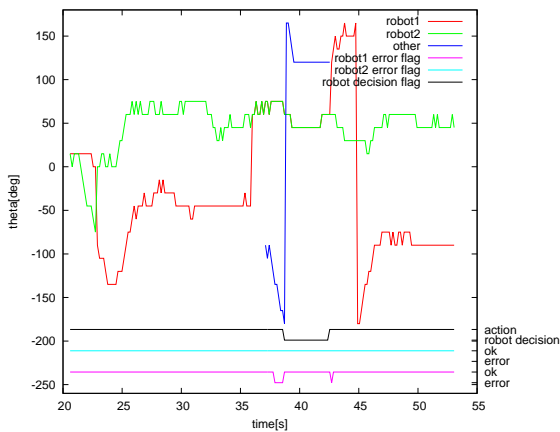


Figure 20: Direction vs. time of robot 1, 2, and other

ら、旋回行動を行っているロボットを1号機と認識することができていることがわかる。

次に、36[s]にロボットの位置が重なっている。これはロボットが接近したため同じボクセル群を自機のロボットだと判断している。1号機と2号機のどちらもが2号機のボクセル群を自機だと認識してしまい、1号機のボクセル群を別のロボットとして認識してしまっている。そのため、38[s]に1号機の異常行動を検知し、号機の再判定を行っている。このとき、1号機が後進している。43[s]に

号機判定が終わり、後進していたロボットを1号機、停止していたロボットを2号機と認識できている。その後、両機とも指定の目標位置に到達できている。

以上の結果により、ロボットが交差したり接触しても、ロボットの異常行動を検知することにより、号機の再判定ができ、その後の行動決定を正しくできていることがわかる。しかし、現段階では、ロボットの号機判定には、ボクセル群の位置と方向しか使っていない。ボクセルデータから推定されたヒト型ロボットのポーズを使うことによって、短時間で確実な判定ができると思われる。

5 おわりに

本稿では、RoboCup SSL Humanoidの共有視覚システムとして利用することを想定して、複数の外部カメラから得られた画像から色付けされたボクセルデータを生成し、それをネットワークで配信する「カラーボクセルサーバ」を提案し、実装したプログラムの実行例を示した。さらに、カラーボクセルサーバから提供された色付けされたボクセルデータに基づいて、オブジェクトの識別とロボットの行動決定を行い、提案手法の有効性を検証した。

今後の課題は、多値化処理を上手く並列処理することで実行周期を短縮することである。

参考文献

- [長坂, 2011] 長坂保典: ロボカップの道しるべ: 第3回 小型ロボットリーグ, 情報処理, vol.52 No.1, pp.95–110, 2011.
- [升谷, 2010] 升谷保博, 成瀬正: 外部カメラを用いたヒト型ロボットによるサッカー競技 RoboCup Soccer SSL Humanoid, 人工知能学会誌, Vol.25, No.2, pp.213–219, 2010.
- [木村, 2010] 木村堯海, 升谷保博: 複数の外部カメラ画像に基づくヒト型ロボットの3次元形状の実時間取得 RoboCupSoccer SSL Humanoid の3次元情報サーバをを目指して, 第31回人工知能学会 AI チャレンジ研究会, pp.33–38, 2010.
- [植田, 2011] 植田康生, 木村堯海, 勢川友樹, 升谷保博: 外部カメラ画像から得られた複数のヒト型ロボットのボクセルデータに基づく行動決定, 第33回人工知能学会 AI チャレンジ研究会, pp.7–12, 2011.

- [M.Seiz, 1997] Steven M.Seiz, Charles R.Dyer: Photo-realistic Scene Reconstruction by Voxel Coloring, Computer Vision and Pattern Recognition Conf, pp.1067–1073, 1997.

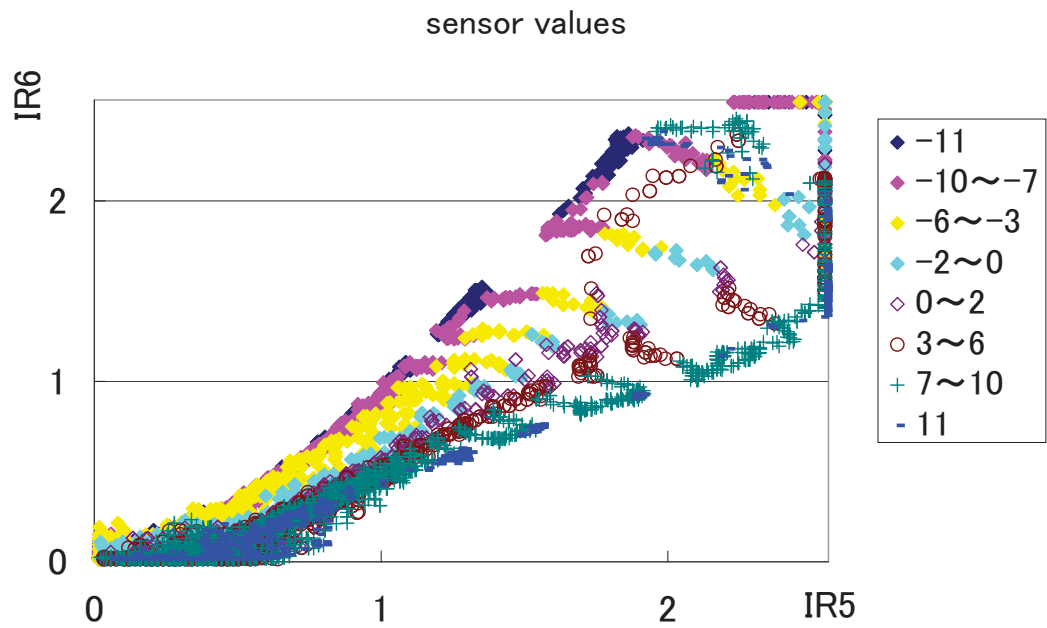


Figure 4: 内側 2 個の距離センサと角度の関係

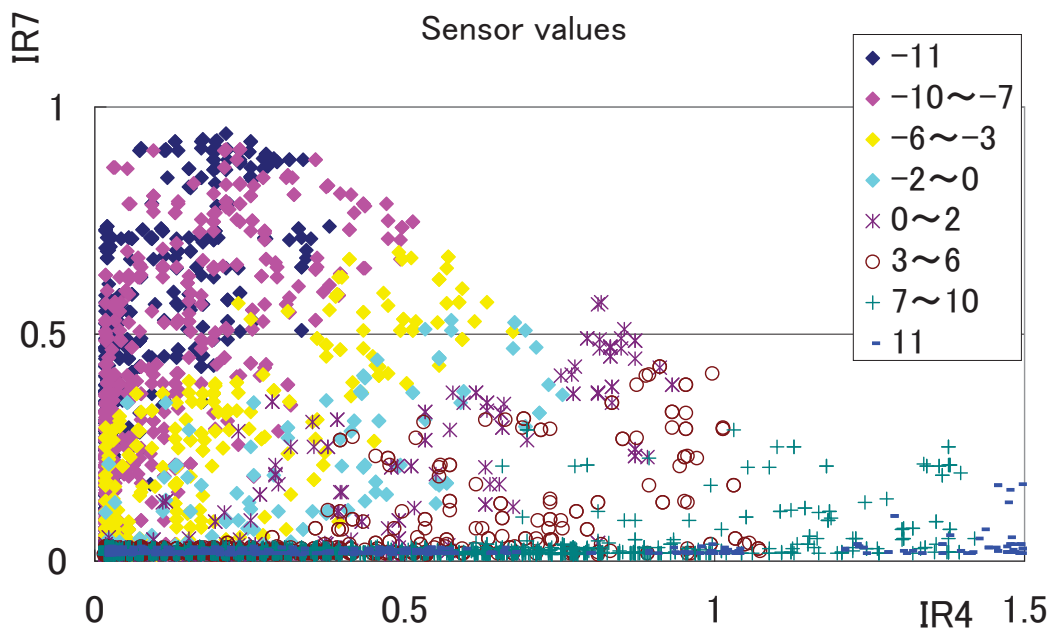


Figure 6: 外側 2 個の距離センサと角度の関係