Ground-truth localisation system for humanoid soccer robots using RGB-D camera

Mohammad Shafiei Rezvani Nezhad* and
Omid Amir Ghiasvand

Faculty of Computer and Information Technology Engineering,
Qazvin Branch,
Islamic Azad University,
Qazvin, Iran
Email: mshafiei@qiau.ac.ir
Email: omid@src.systems
Email: Amirghiasvand@qiau.ac.ir
*Corresponding author

Abstract: Localisation and behaviour are two vital systems for every autonomous mobile robot. As the result, establishment of a localisation ground-truth detection system is a necessity for the analysis. Depend on the requirements of the application, there are some constraints for such system like: diversity in types of providing information, accuracy of the data and ease of use – since the calibration is needed in both colour classification and geometry position – and also the cost of the system, which all should be considered. In this paper, a low cost system is presented using Asus Xtion depth camera, capable of providing ground-truth position information of the humanoid robots and the ball which is required to estimate the errors of a localisation system in RoboCup Standard Platform League (SPL) and Humanoid kid Size League (HSL) soccer robots. Furthermore, the user input requirements are reduced to very few compared to other similar systems. Source code of the system is available online at https://github.com/mrlspl/.

Keywords: ground-truth; localisation; humanoid soccer robot; depth camera.

Reference to this paper should be made as follows: Nezhad, M.S.R. and Ghiasvand, O.A. (xxxx) ‘Ground-truth localisation system for humanoid soccer robots using RGB-D camera’, *Int. J. Computational Vision and Robotics*, Vol. X, No. Y, pp.000–000.

Biographical notes: Mohammad Shafiei Rezvani Nezhad received his BSc in Software Engineering from Qazvin Azad University, Qazvin, Iran in 2014. He was a member of NAO Humanoid Robot lab in Mechatronics Research Center. He is a Master student in Saarland University, Saarland, Germany. His main research interests are machine vision and robotics.

Omid Amir Ghiasvand received his BS and MS from Qazvin Azad University, Qazvin, Iran, in 2007 and 2010 respectively and starts his PhD in 2013. He joined Mechatronics Research Laboratories in 2004 as a Researcher. From 2008, he is part of the Board of Director in Qazvin Azad University R&D Center and Director of Software Research Center (SRC). His main areas of research interest are software engineering, agent architecture, behaviour control, semantic search and knowledge representation and reasoning.

1 Introduction

Estimation of self-position in an indoor environment is a need of almost every autonomous robot decision making and navigation. Developing, and comparing such algorithms require ground-truth information which can also be used for verifying the decisions of the robot based on current environment state.

In this paper a low cost ground-truth localisation system is presented which minimises user's role in system calibration and provides most of the information from the soccer match including: robots, ball and referees positions in addition to team colour identification

Game field size has been in a fixed size for years, as the result, most of the ground truth systems cannot be used in larger fields. The main contribution of this work is the simplification of calibration procedure and the diversity in provided information. According to the large field in which robots are having interaction, there is a need for ground truth systems to be extendable. As the result, the calibration procedure will be cumbersome since many cameras are used. Therefore, in this paper the calibration procedure is further simplified. In Section 5, it is shown that the number of cameras can easily get extended to cover the field. SPL and HSL game fields [9 × 6 metres according to Standard Platform League Rules Book, 2013 and 6 × 4 according to Humanoid League Rules Book (2013)] and their corresponding robots – namely NAO and DARwIn – have been chosen as the system's testbed.

2 Related work

Available ground-truth systems debated in RoboCup scenarios are categorised based on using or not using marker on the robot and the total cost of the system.

Drawbacks of systems such as Niemüller et al. (2011), are mostly the cost of the system, the other drawback which also exists in Bischoff et al. (2012), can be named as time consuming setup procedure, including placing markers on robot and camera geometric calibration. Besides, markers on robot can distract robot detection algorithms. Thus, behaviour decisions such as obstacle avoidance would be affected. Although methods which use markers for object localisation are so accurate in estimating the location, but the cost of these systems is basic cause for which almost none of SPL teams have employed such systems.

Systems such as Zickler et al. (2009), Lochmatter et al. (2008), Ceriani et al. (2009), and Schoenmakers and Janssen (2013), are still time consuming as of the user have to calibrate the geometric position of the camera and also manually identify marker regions' colours to the software. Moreover, since the league's field size in RoboCup leagues especially SPL are likely to grow annually, this approach leads to increase the height of the camera position or reduce focal length of the camera lens which cause respectively need to a sensor with greater resolution or inducing more optical distortion. These methods are also used by number of teams in SPL such as Röfer et al. (2011), for testing the self-position estimation accuracy. Although these systems provide good accuracy since they also use markers, but cannot be used in a match because no additional part can be used in a robot soccer scenario.

Due to the drawbacks of mentioned approaches and thanks to low cost depth sensors – such as Microsoft Kinect and Asus Xtion^{TM} – Khandelwal and Stone (2011) and Pennisi et al. (2013) have suggested a method using point-clouds for ground-truth detection system. The calibration process used by Khandelwal and Stone (2011) requires user to identify ground points and field's landmarks in order to calculate the camera transformation to the field reference which is time consuming. Moreover, colour calibration must be performed to make the system able to tell the robots' team colour.

In method proposed by Pennisi et al. (2013), robot position is provided along with its orientation but the calibration procedure would still be time consuming. Since it must be performed on four depth cameras placed in the outer edge of the field each by which a set of landmark positions must be manually selected in images. These positions will then be used for calculating the transformation matrix. Additionally, the approach does not support any team colour identification.

Some Ground-truth systems are developed employing mono cameras. For instance, Schoenmakers et al. try to use several mono cameras each of which detects the objects independently thereafter merge the data in overlapped areas. Using such cameras and constrains of acquired data, causes detection of false positives for which the author is using a particle filter-based method to tell the false positives from true positives. Also there are confusions in tracking objects when the players get close to each other.

In Liu et al. (2012), Zhang and Zhang (2011) and Herrera et al. (2012), the relative position between RGB and depth cameras is estimated using markers, which is not necessary in suggested system due to the appropriate amount of experimented error with the documented position values. In Tong and Barfoot (2011), multiple landmarks are required to be visible inside an image for an accurate calibration. In Marchant, a single laser scanner is used for gathering all field data. Such a system has drawback on invisible regions made by objects, placing multiple laser scanners around the field causes an increase in cost of the system. In addition, such a system cannot recognise objects with different heights.

As the result of literature review, three main factors could be named as properties of an appropriate ground-truth system:

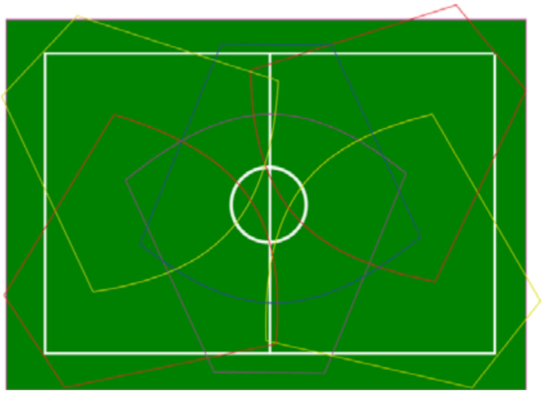
- total system cost
- ease of system calibration procedure
- diversity in types of provided information.

The system proposed by this paper, is low cost as it is shown in Section 3. As the contribution, the calibration procedure is further simplified. Moreover, the diversity of information provided by the system has been extended. i.e. the position of the referees in the field is also provided.

3 System architecture

Suggested system is developed using Asus Xtion Pro Live, but it can also use Microsoft Kinect cameras or any other RGB-D cameras supported by point cloud library (PCL) (Rusu and Cousins, 2011). It consist six RGB-D cameras which would be placed on the outer edge of the 2013 field of SPL, four of which are placed in the corners, and two placed across the middle line intersection to the side lines. Suggested cameras formation shown in **Error! Reference source not found.** is selected mainly because at least one field lines intersection must be visible by each camera to be recognised as a landmark with known position relative to field reference point. The reason would be precisely discussed in Section 0. All six cameras are connected to a single computer and all data are processed through single software. Although the field coverage is fulfilled by 4, but due to the error of point-cloud data relative to the object's distance to the camera, 6 would reduce the error.

Figure 1 Cameras formation and field coverage (see online version for colours)



3.1 Code base

Algorithms presented in this paper, are implemented in C++ language and are available online (MRL-SPL Team, 2013). The algorithms described in this paper are implemented by the authors. But many other algorithms are previously implemented by PCL (Rusu and Cousins, 2011) or OpenCv Library (Bradski et al.) and are used here. These algorithms are specified in this paper by *pcl* or *cv* suffix respectively for each library (e.g., *pclHough*, *pclRANSAC*, *cvCanny* etc.).

3.2 Knowledge representation of the environment

Images captured by RGB-D cameras, are stored in camera coordinate system r_i (i corresponds to i^{th} camera). Object detection process is performed in this coordinate system. As the result, all outputs of object detection system is relative to r_i. Since there are many cameras placed around the field, they should all be transformed to a single coordinate system called r_{field}. This coordinate is placed in the centre of the field, X axis pointing parallel to the height of the field and Y axis parallel to its width (Figure 2). The calculating the transformation (T_i) is described in Section 3.3.3.

3.3 Camera geometric calibration

Since robots are playing inside game field and their estimated self-position is expressed relative to the field reference [r_{field} Figure 2(a)], for that reason in the ground-truth system the position of every object must be expressed to the same reference. Assuming to have the objects' position relative to a camera reference point (r_i), these points can be transformed into their position in r_{field} by a transformation matrix which can be computed in three steps with certainty:

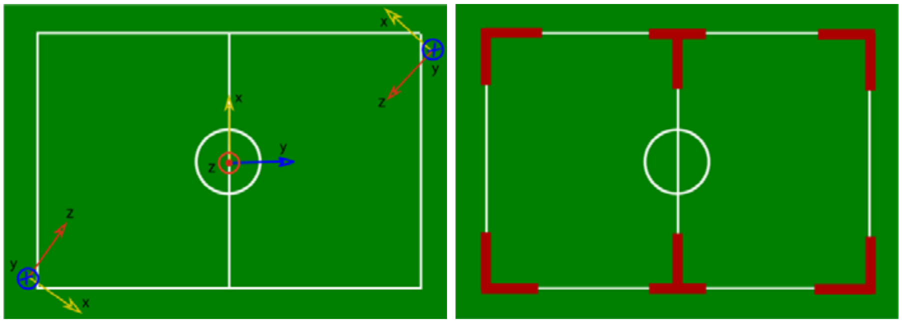
1 detecting an object relative to r_i with a previously known position in r_{field}

2 an accurate estimation of roll and tilt of field surface

3 detecting a previously known vector in r_{field}.

Fulfilling steps 1 and 2 let us calculate the normal vector of field surface. These steps are respectively discussed in following sections and concluded with how one can provide the transformation matrix whereby.

Figure 2 Camera coordinate systems (*ri*) in the corners. Transformation matrix Ti allows the transformation from *ri* to rfiled coordinate system, shown in the centre in (a). Line intersections used as landmarks (b). (see online version for colours)



(a) (b)

3.3.1 *Landmark detection*

Field lines' intersection [Figure 2(b)] can be used as landmarks by which the translation of the camera can be expressed. Since every camera can see at least one landmark, it would be an easy task to recognise. Using cvCanny edge detector on the *Cr* channel of colour image with a static experimentally selected threshold value, and running Hough transform method (cvHoughLines2), inclusive landmark lines would be recognised. Since many lines and intersections are usually visible to the camera, following criterions will be checked for the detected intersections:

1 closeness

2 similarity of the intersection angle to a right angle

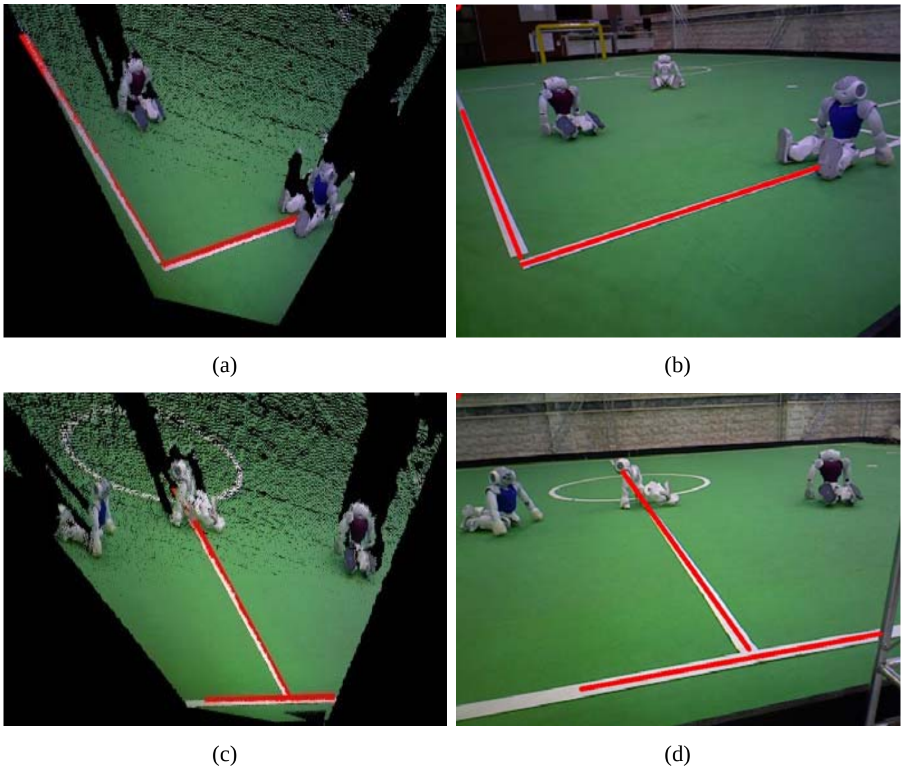
3 intersecting lines to include two parallel edges with 5cm mutual distance.

To be able to evaluate mentioned values, the position of the extracted line vertexes relative to the camera are needed. Since there might be some objects placed on the line, a robust method is needed to find 3D vertexes. Therefore, RANSAC algorithm (pclSACSegmentation) is employed to find the line equation and the vertexes. Thereafter the vertexes are calculated by finding the minimum and maximum point values. Output is shown in **Error! Reference source not found.**.

3.3.2 Surface extraction

Considering cameras facing the field, surface extraction algorithms such as RASNAC are more probable to extract the appropriate surface in their very first iterations. To check whether the extracted surface belongs to the field, the distance of all line vertexes which belong to the detected intersections to the extracted surface would be measured. The one with the lowest distance summation is selected as the field surface.

Figure 3 (a, c) 3D line segments and (b, d) 2D in the presence of outliers (see online version for colours)



3.3.3 Rotation matrix

Constructing the three unit vectors of r_{field} coordinate axes in r_i are inclusive data to build the rotation matrix needed to rotate a given point from r_{field} to r_i. Detecting the line of a landmark parallel to the x axis of r_{field} and calculating its unit vector, gives another vector out of three mentioned. By using these two unit vectors and (1) the considered matrix is built.

Additional to the rotation matrix, translation values are also essential to build final transformation matrix. These values are calculated by rotating detected landmark position to r_{field}. Camera translation can be calculated using summation of landmark position and its rotated position. Transformation matrix is completed by placing translation values in their corresponding indexes inside. This is illustrated in (2).

$$n_s \times n_l = n_c \tag{a}$$

$$R = \begin{bmatrix} n_{lx} & n_{cx} & n_{sx} \\ n_{ly} & n_{cy} & n_{sy} \\ n_{lz} & n_{cz} & n_{sz} \end{bmatrix} \tag{b}$$

$$\mathrm{pt} = \mathrm{R} \times \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{bmatrix} \tag{c}$$

$$T_i = \begin{bmatrix} R_{11} & R_{12} & R_{13} & -pt_x + L_x \\ R_{21} & R_{22} & R_{23} & -pt_y + L_y \\ R_{31} & R_{32} & R_{33} & -pt_z \end{bmatrix} \tag{d}$$

Equation (1):

a three normal vectors (n_s, n_l, n_c respectively correspond to normal vector of the surface, normal vector of one one of the lines of landmark parallel to X axis of r_{field}, normal vector parallel to axix Y of r_{field})

b rotation matrix which rotates the points in camera coordinate to field coordinate

c rotated intersection point (pt)

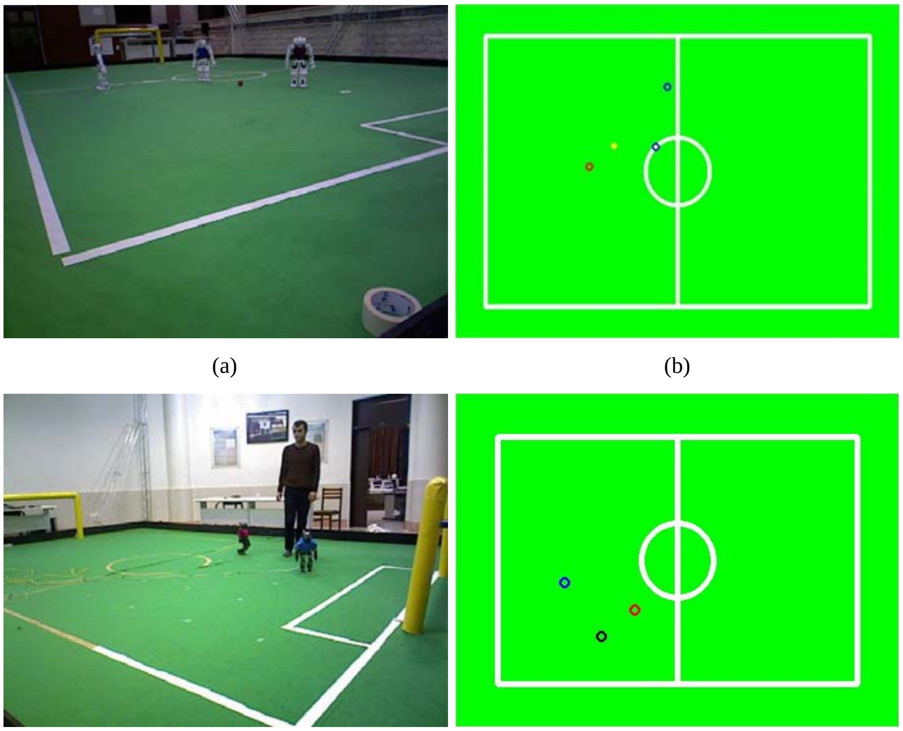
d transformation matrix by which it is possible to transform any point from r_i to r_{field}.

3.4 Object detection

Prior of detecting any object, any unnecessary point which does not belong to field objects or the referee must be removed. These points include those of the surface and points far from the camera and obviously out of the field. For removing the latter, simply the points that are far than 5 metre would be dismissed, for the former, the points with a minimal threshold distance to the surface would be removed. This threshold must be less than height of the smallest object which must be detected, radius of the ball (6cm). Figure 5 demonstrates the result.

After removing unnecessary points, objects would be detected by a Euclidean clustering method (pcl**Euclidean**ClusterExtraction). Threshold of 20 cm is set for clustering (due to maximum error probable in 6 metres) minimum points in each cluster is experimentally set to 8 for the ball and 50 for robot. These clusters still encompass both true and false objects in the field.

Figure 4 Object detection and colour classification output a and b belong to SPL field (9 × 6 metres) while c and d shows HSL (6 × 4 metres) (see online version for colours)



(a) (b)

(c) (d)

One of the drawbacks of using point cloud in such systems which are using perspective view point is the occlusion (e.g., when one robot is occluded by another robot. This case is not considered in this system. But such problems could be enhanced by an appropriate tracking method.

3.4.1 Non-object removal

Passing through object detection routine, leaves point-cloud clusters to be determined as specific objects or non-object clusters. The task of this section is to successfully remove non-object clusters and distinguish object clusters by their dimension properties in the first place, and their colour. Noise removal routines are performed considering desired objects size. Those which their height is below 10cm would be considered as ball candidates, between 10–30 centimetre false clusters that would be ignored, 30–70 centimetre are robot candidates and above 70, referees.

Considering the point-clouds existing beyond the field borders are removed, non-object clusters are divided into two basic classes:

1 point-clouds belong to referees legs which have the height close to robot’s due to the limited camera’s vertical field of view

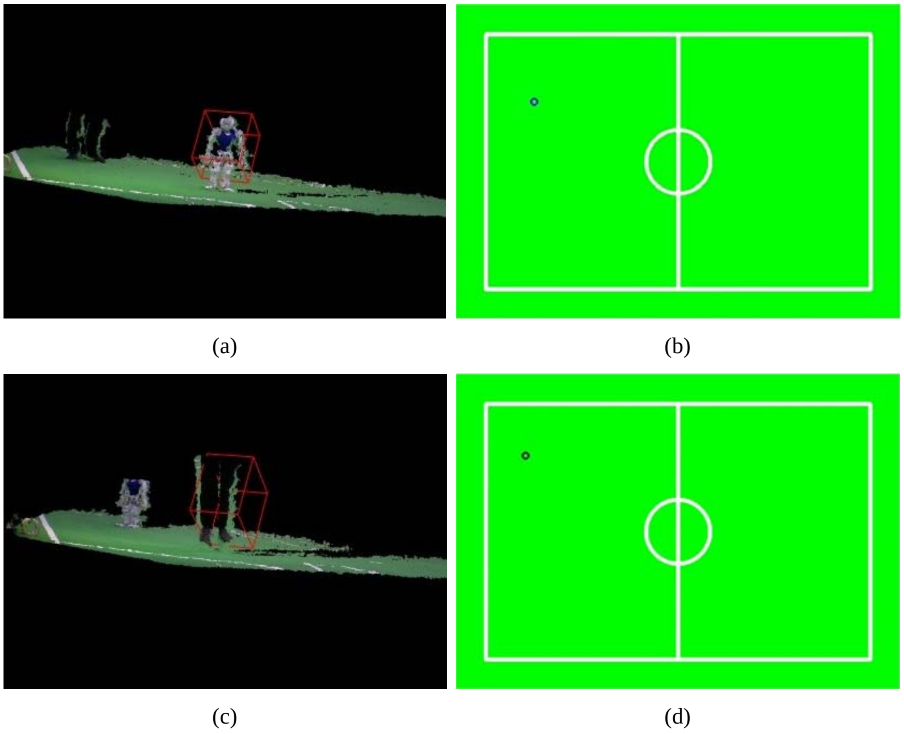
2 multiple detected objects when a robot is placed far from the camera caused by camera noise in far distances

3 field points estimated above the field caused by mentioned estimation error in far distances; first situation is illustrated by Figure 5.

In the following paragraphs solutions for omitting every non-object clusters are discussed.

The vertical field of view of the camera is bounded by two intersecting surfaces with 45 degree internal angle. The intersection of upper surface to the field's surface – if the intersection occurs below the camera depth range – prevents the tall objects – referees in this case – to be wholly visible. As the result, the upper surface boundary of camera field of view is calculated relative to r_{field} and employed to compute the minimum distance of clusters top point to the surface whereby it would be ensured that the cluster might belong to a referee's leg. In this case it would be removed.

Figure 5 (a) Noisy point-cloud after removing the surface and referee legs partly observed (b) Successful noise removal (see online version for colours)



(a) (b)

(c) (d)

As a result of point-cloud estimation error in 3.5–4 metre range, several clusters would be detected with the height close to robots. In order to homogenise this multiplicity, Euclidean clustering is employed and would be run in the field's 2D space with every robot detected for each camera separately. This avoids the error of one camera to influence other's possible accurate information in overlapped area. The reason will be widely described in Section 0.

For the same error which caused the previous non-object detection, in the range of 3.5–4 metre, field points are estimated above the surface which they are actually placed. By removing points closer than 2 centimetres to the extracted surface, mentioned points would cause detecting non-object clusters. Only prominent characteristic of the ball which is distinct to field's points, after regarding its dimensions, is its colour. After colour classifying clusters (described in Section 1.2) orange cluster in its specified dimensions is considered as ball.

3.5 Colour classification

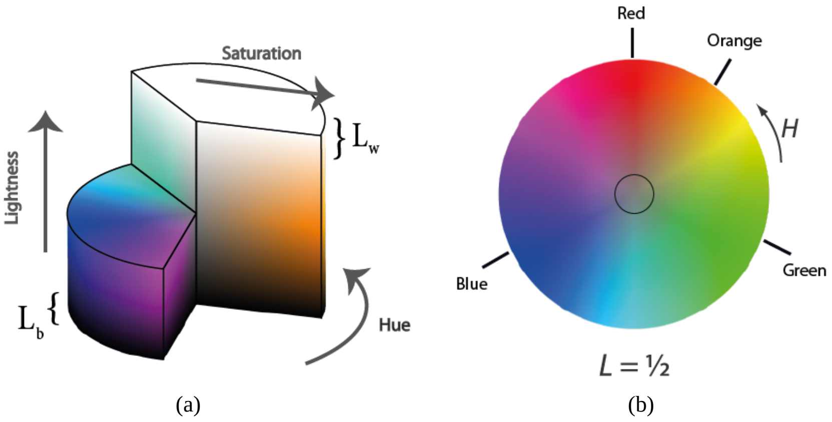
Employing colour tables by Bruce et al. (2000) is a de facto in RoboCup applications. The main reason is the fact that the colours which are close in colour spaces, such as yellow, orange and red are to be recognised in the field and no simple approach can tell the colour of objects apart in each match due to environment's luminance variation.

Some approaches have been introduced which use both colour and special values to perform unsupervised image segmentation. As instances, topographical distance of pixels used by Lefèvre (2010), using clustering technique leading to unsupervised image segmentation and Le et al. (2010), which utilises neighbouring pixels blocks along with their colours in order to perform supervised classification. Dellen et al. (2011) employed 3D special data to enhance it segmentation output. In contrast, Kashanipour et al. (2008), employ only pixels' colour values individually to decide their membership to colour labels which is performed by optimised membership functions.

But in this paper, a more simple approach is chosen to classify the colour of each object. As it is shown in Figure, in HSL colour space a single value of hue can represent a colour class if an input pixel is expected to be classified for either of only two distinct colour classes (red and blue in this case). As respects an object pixel's colour value always has the minimum distance in colour space to its actual colour class prototype, suggested approach for colour classification is using this distance as a criterion.

The task of colour classification is always telling one colour out of two as output class which is expected to be between either orange and green (ball and field noise) or red and blue (robots' team colours). Therefore the distance of every object's pixel hue channel is calculated to every available colour prototype. The prototype to which the pixel has lower hue distance is considered as the respective colour-class. After iterating this routine for every pixel of the object, total colour-class numbers are calculated and the one with maximum number is selected as object's colour-class. Figure 4 is demonstrating the result.

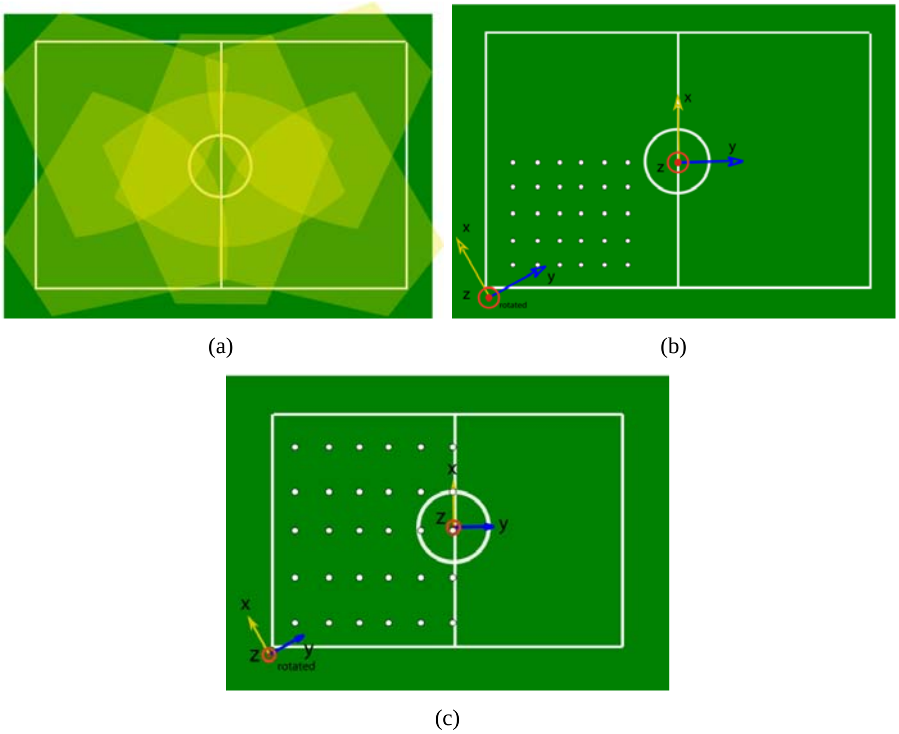
Figure 6 (a) Pixels with L value in L_w range are considered white, and those in L_b range as black (b) Pixels dismissed with low saturation value (centre circle) and prototypes hue values (see online version for colours)



(a) (b)

Source: Figures are a manipulation from originals (Rus, 2003)

Figure 7 Overlapped areas with maximum number of four cameras observing an area (left side and right side of the circle shown in a), (a) rotated r_i before translation (b) the criterions in HSL game field (c) SPL (see online version for colours)



(a) (b)

(c)

4 Overlapped areas

Due to the fact that ground-truth localisation system include six cameras around the field, many zones are regarded as overlapped areas which are visible by two or more cameras [Figure 7(a)].Correlation between object's distance to the camera and error, makes it rational to select the nearest camera to the object to use its data. Corresponding robots in cameras can be identified employing clustering technique. In other words, detected robots in all cameras which are expressed relative to r_{field} would be clustered at the first place, thereafter one of clustered robots with minimum distance to the camera which it is detected by, would be chosen as the one with most accurate information.

5 Experimental results

SPL and HSL game fields [9 × 6 metres (Standard Platform League Rules Book) and 6 × 4 (Humanoid League Rules)] and their corresponding robots - namely NAO and DARwIn-OP – have been chosen as the system's test bed. To prove the accuracy of the system, two methods are considered:

5.1 Position error disparity

Position error is measured between detected position and ground truth of robot position in the field. There are different disparity scheme of error values with different methods for computing the robot's position. Centre of mass [equation (3)] and centre of bounding box [equation (4)]. The results justify that the centre of bounding box is more accurate. Finally the overall error measure is shown in Table 1.

5.2 Standard deviation of landmark positions

By this method, the accuracy of geometric calibration is evaluated. In order to produce the results, calibration procedure is performed several times, and the standard deviation of landmark positions is computed.

In order to estimate the position error, two humanoid robots (NAO and DARwIn) have been placed on known positions which are used as criterions to calculate the error in their corresponding game fields [Figure 7(b) and Figure 7(c)] and calculated position by the system is compared relatively and the error is computed separately. The position error is directly influenced by method of determining robot final position in the field. As it is pointed out by Khandelwal and Stone (2011), determined position is highly affected by robots' orientation on the field. This is true for the position calculated by arithmetic mean of the robot point-cloud (centre of mass) and the reason is that the position would be placed where the point-cloud has more density which is the region closer to the camera in its direction. It will cause an unpredictable error in camera y direction [Figure 8(a) and Figure 8(b)]. On the other side, computing centre of bounding box of robot helps to have a more realistic estimation of centre of robot's torso rather than centre of mass approach (Table 1). Although the representative point shift along direction of camera will remain, but because there is a constant shift in all robot's orientations the error can be reduced by adding constant value to the results [Figure 8(c) and Figure 8(d)]. By fitting a linear equation on acquired data [(3) and (4)], it becomes obvious that the error of representative point have a small correlation with distance of point-cloud to the camera and there is a 0.0339 metres shift in $+x_{camera}$ direction and 0.01811 metres in $+y_{camera}$. For DARwIn robots, shift value is estimated as 0.11 metres in $+y_{camera}$ and approximately no shift is observed toward camera's x direction. Mean error for DARwIn robot is estimated as 0.058 (± 0.029) metres in x and 0.056 (± 0.028) in y direction. Error distribution after applying the shift values is shown in Figure 9 (for DARwIn). Error distribution before and after applying the shift values is shown in Figure 8 (for NAO).

Equation (2): equation fit for NAO robots which shows the representative position of robots have a constant shift from the actual data

$$\begin{aligned} X_{corrected} &= 1.006 * x_{input} + 0.0399 \\ Y_{corrected} &= 1.011 * y_{input} + 0.01811 \end{aligned} \tag{2}$$

Equation (3): centre of mass of robots. N is the number of points in the point cloud of the robot object

$$\text{CoMrobot} = \frac{1}{N} \sum_{i=0}^{N} P_i \tag{3}$$

Equation (4): centre of bounding box of the robot

$$\text{CoBrobot} = \left[\begin{pmatrix} \min(P_{ix}) \\ \min(P_{iy}) \\ \min(P_{iz}) \end{pmatrix} + \begin{pmatrix} \max(P_{ix}) \\ \max(P_{iy}) \\ \max(P_{iz}) \end{pmatrix} \right] \times \frac{1}{2} \tag{4}$$

$$i \in \{1, .., N\}$$

Table 1 Mean error of NAO robot position for two methods of position representation

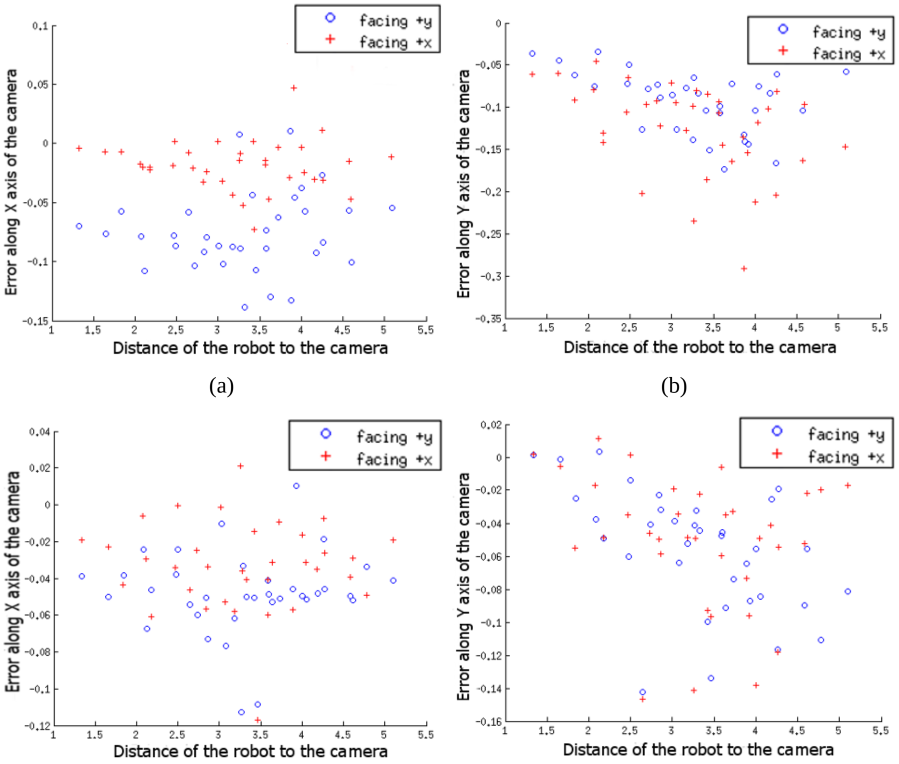
	Facing $+y_{field}$		*Facing* $+x_{field}$		*Overall*	
	Error in $+x_{rotated}$	*Error in* $+y_{rotated}$	*Error in* $+x_{rotated}$	*Error in* $+y_{rotated}$	*Error in* $+x_{rotated}$	*Error in* $+y_{rotated}$
Centre of mass	0.0926 (±0.0463)	0.0860 (±0.0430)	0.0444 (±0.0222)	0.1474 (±0.0737)	0.0678 (±0.0339)	0.1176 (±0.0588)
Centre of bounding box	0.0411 (±0.0205)	0.0503 (±0.0251)	0.0365 (±0.0182)	0.0629 (±0.0315)	0.0388 (±0.0194)	0.0566 (±0.0283)

Equation (5) equation fit which shows the representative position of DARwIn robot

$$\begin{aligned} X_{corrected} &= 1.00 * x_{input} - 0.00692 \\ Y_{corrected} &= 0.9986 * y_{input} + 0.1178 \end{aligned} \tag{5}$$

According to Table 1, it can be seen that, the error of the position is close to the error represented by Khandelwal and Stone (2011). And Pennisi et al. (2013), but with centre of bounding box and applying the shift toward x_{camera} and y_{camera}, these error values are reduced.

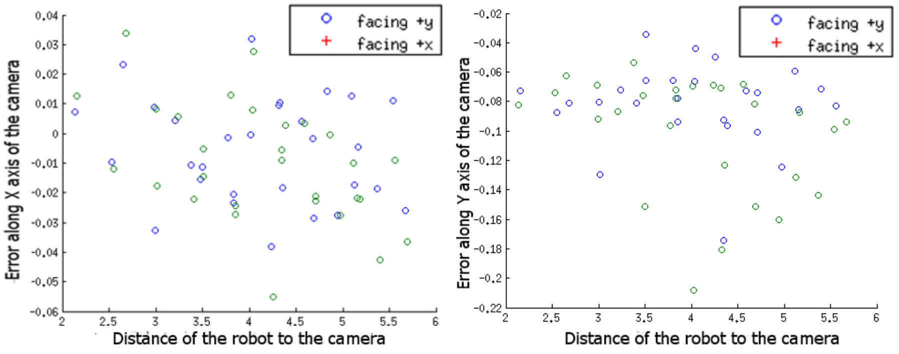
Figure 8 For NAO robots, (a) error in robot position x value relative to r_i, when the robot is facing +y and +x of the field (b) error in robot position y value relative to r_i, (c) (d) error values after position enhancement (applying the shift values) (see online version for colours)



(a) (b)

(c) (d)

Figure 9 Error values for DARwIn humanoid robots in HSL game field (see online version for colours)



(a) (b)

To evaluate the colour classification method, the numbers of false colour classifications have been counted in each position showed in Figure 7(b) for 60 frames. Evaluated number was 0.46 % of the times and the detected false colours were only observed in furthest positions relative to camera and mostly in overlapped areas this causes the false colour detection to happen even less times. Camera position estimation procedure is performed under four scenarios which differ by luminance condition and field size (9×6 SPL game field and 6×4 Humanoid kid size soccer robot). The calibration procedure is performed 20 times for each scenario and the standard deviation of an estimated landmark position (centre of the field) is 0.0243 metres in SPL game field and 0.0195 metres in humanoid kid size which is acceptable due to the overall size of system's error. The latter error value is comparable to Khandelwal and Stone (2011), which has the standard deviation of 0.0202 metres. The main drawback of calibration method used in this paper, is the error of camera orientation around z axis. This error exists according to the inaccuracy in line detection procedure. But it is acceptable according to the robot position accuracy illustrated in Figure 8. With this method of camera geometry calibration, no higher accuracy is gained but a semi-automatic method has been obtained which simplifies the calibration procedure.

6 Future works

The ground truth localisation system which is presented in this paper could be extended to use in all possible scenario in which robots move in bounded area not just in 2D but also in 3D space such as indoor autonomous quadcopter.

Only essential object would be line intersection landmarks observable to cameras. This could be reached by changing the configuration of landmark positions and objects' size and colour.

Calibration process can be performed online to avoid problems caused by probable movement of camera due to unexpected stroke. Strokes might happen since the cameras are close to the field and referees are mostly walking around the field.

Since robots standing on the field prevent any point-cloud to be acquired from the region behind it, there would be multiple invisible regions in camera field of view. Computing these regions can assist debugging robots' behaviour system by providing the probability of objects to be observed by particular robots regarding their position.

Tracking detected robots would be a simple task since no significant error is detected and there are no false positives. Tracked objects simplify debugging routines. It can also help detecting collision, which suggested system is not capable of. Although the points near those robots are tightly close to each other but the point-cloud is large enough to detect each of them. The robots' jersey shirt can also assist this action especially in case that the colours of shirts differ. Likely, robot tracking makes the system informed about number of robots to search for in collision area.

References

Bischoff, B., Nguyen-Tuong, D. and Streichert, F. et al. (2012) 'Fusing vision and odometry for accurate indoor robot localisation', *International Conference on Control Automation Robotics & Vision (ICARCV)*, December, pp.347–352, Guangzhou, China.

Bradski, G. (2000) 'The OpenCV Library', *Dr. Dobb's Journal of Software Tools*, Vol. 25, No. 11, pp.120, 122–125.

Bruce, J., Balch, T. and Veloso, M. (2000) 'Fast and inexpensive color image segmentation for interactive robots', *Intelligent Robots and Systems (IROS)*, Vol. 3, pp.2061–2066, Takamatsu, Japan.

Ceriani, S., Fontana, G. and Giusti, A. (2009) 'Rawseeds ground truth collection systems for indoor self-localisation and mapping', *Autonomous Robots*, November, Vol. 27, No. 4, pp.353–371.

Dellen, B., Alenya, G. and Foix, S. et al. (2011) 'Segmenting color images into surface patches by exploiting sparse depth data', *IEEE Workshop on Applications of Computer Vision (WACV)*, pp.591–598, Kona, Hawaii.

Herrera, C.D., Kannala, J. and Heikkilä, J. (2012) 'Joint depth and color camera calibration with distortion correction', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October, Vol. 34, No. 10, pp.2058–2064.

Humanoid League Rules (2013) [online] http://www.robocuphumanoid.org/wp-content/uploads/HumanoidLeagueRules2013-05-28.pdf (accessed August 2015).

Kashanipour, A., Kashanipour, A.R. and Shamshiri Milani, N. (2008) 'Robust color classification using fuzzy reasoning and genetic algorithms in RoboCup soccer leagues', *RoboCup 2007: Robot Soccer World Cup XI*, Vol. 5001, pp.548–555.

Khandelwal, P. and Stone, P. (2011) 'A low cost ground truth detection system using the kinect', *RoboCup 2011: Robot Soccer World Cup XV*, July, Vol. 7416, pp.515–527.

Le, T.T., Tran, S.T. and Mita, S. et al. (2010) 'Real time traffic sign detection using color and shape-based features', *Intelligent Information and Database Systems*, Vol. 5991, pp.268–278.

Lefèvre, S. (2010) 'A new approach for unsupervised classification in image segmentation', *Advances in Knowledge Discovery and Management*, Vol. 292, pp.113–131, Strasbourg, France.

Liu, W., Fan, Y. and Zhong, Z. et al. (2012) 'A new method for calibrating depth and color camera pair based on kinect', *International Conference on Audio, Language and Image Processing (ICALIP)*, July, pp.212–217, Shanghai, China.

Lochmatter, T., Roduit, P. and Cianci, C. et al. (2008) 'SwisTrack – a flexible open source tracking software for multi-agent systems', *Intelligent Robots and Systems (IROS)*, September, pp.4004–4010, Nice, France.

Marchant, R., Guerrero, P. and Ruiz-del-So, J. (2011) 'A portable ground-truth system based on a laser sensor', *Robot Soccer World Cup XV*, Vol. 7416, pp.234–245.

MRL-SPL Team (2013) [online] https://github.com/mrlspl/ (accessed July 2015).

Niemüller, T., Ferrein, A. and Eckel G. (2011) 'Providing ground-truth data for the Nao Robot platform', *RoboCup 2010: Robot Soccer World Cup XIV*, Vol. 6556, pp.133–144.

Pennisi, A., Bloisi, D. and Iocchi, L. (2013) 'Ground truth acquisition of humanoid soccer robot behaviour', *17th Annual RoboCup International Symposium*, July, Mexico City, Mexico.

Röfer, T., Laue, T., Müller, J. and Fabisch, A. et al. (2011) *B-Human Team Report and Code Release* [online] https://www.b-human.de/downloads/bhuman11_coderelease.pdf (accessed July 2015).

Rus, J. (2003) *HSL and HSV*, 25 June [online] http://en.wikipedia.org/wiki/HSL_and_HSV (accessed December 2013).

Rusu, R.B. and Cousins, S. (2011) '3D is here: point cloud library (PCL)', *IEEE International Conference on Robotics and Automation (ICRA)*, May, pp.1–4, Shanghai, China.

Schoenmakers, F. and Janssen, R. (2013) *Greenfield Augmented Reality* [online] http://www.techunited.nl/wiki/index.php?title=Greenfield_Augmented_Reality (accessed December).

Standard Platform League Rules Book (2013) [online] http://www.tzi.de/spl/pub/Website/Downloads/Rules2013.pdf (accessed December 2013).

Tong, C.H. and Barfoot, T. (2011) 'A self-calibrating 3D ground-truth localisation system using retroreflective landmarks', *IEEE International Conference on Robotics and Automation (ICRA)*, May, pp.3601–3606, Shanghai, China.

Zhang, C. and Zhang, Z. (2011) 'Calibration between depth and color sensors for commodity depth cameras', *IEEE International Conference on Multimedia and Expo (ICME)*, July, pp.1–6, Barcelona, Spain.

Zickler, S., Laue, T. and Birbach, O. (2009) 'SSL-vision: the shared vision system for the RoboCup small size league', *RoboCup 2009: Robot Soccer World Cup XIII*, Vol. 5949, pp.425–436.