Real-time landmark modelling for visual-guided walking robots

Renato Samperio* and Huosheng Hu

School of Computer Science and Electronic Engineering,
University of Essex,
Colchester CO4 3SQ, UK
Email: rsampe@essex.ac.uk
Email: hhu@essex.ac.uk
*Corresponding author

Abstract: The environment sensing through visual perception can be obtained by an efficient and reliable landmark modelling. This paper proposes a novel landmark modelling approach to the classification of both the predefined and novelty features in the robot environment. The approach integrates image descriptors for defined landmarks (natural and artificial) and novelty features being dynamically detected. It has been implemented in a quadruped walking robot that has a single camera in its head. Some experimental results are presented to demonstrate the feasibility and performance of the proposed landmark models.

Keywords: image processing; visual-based localisation; walking robots; landmark modelling; real-time system; RoboCup.


Biographical notes: Renato Samperio received his BSc degree from Instituto Tecnológico de Estudios Superiores De Monterrey in Mexico, and PhD degree from the University of Essex in the UK. Currently he is a Research Engineer in the IT Innovation Centre at the University of Southampton. His research interests include data fusion, system design, machine learning, computer vision, legged robots, robot navigation, sensors integration, data association, and cooperative robotics.

Huosheng Hu is a Professor in School of Computer Science and Electronic Engineering at the University of Essex, UK, leading the Human Centred Robotics Group. His research interests include biologically inspired robotics, service robots, human–robot interaction, evolutionary robotics, sensor integration, data fusion, artificial intelligence, embedded systems, mechatronics and pervasive computing. He has published more than 300 papers in journals, books and conference proceedings in these areas. He is one of founding members of IEEE Robotics and Automation Society Technical Committee on Networked Robots, a Fellow of IET and InstMC, a senior member of IEEE and ACM. Currently, he is also serving as one of Editors-in-Chief for International Journal of Automation and Computing.

1 Introduction

The availability of autonomous robots with specific purposes makes an execution possible in highly specialised human-like environments such as a football match. In Kitano et al. (1997), the RoboCup domain legged robots acquire visual information from an unstable and unreliable camera perception. Therefore, a real-time landmark modelling has been adopted for assimilating such surrounding features.

The landmark modelling is responsible for the assimilation and description of robot surroundings. The assimilation problem involves two types of landmarks: (a) artificial and (b) natural. The natural landmarks are part of the robot environment which are detected from a bottom-up processing approach. The artificial landmarks are any external environmental feature in a top-down recognition. The proposed method assimilates artificial landmarks from a fixed environment and it is supported by references of natural landmarks.

During robot localisation or navigation, landmarks are also described by shape characteristics or according to their functionality. Specifically, in this paper we present a hybrid landmark method which implements colour-coded and real-time landmark detection.

The techniques for detecting predefined landmark models do bring significant contributions to measure and track natural and artificial landmarks. In Becker et al. (1995), a dynamic environment uses predefined coded landmarks for detecting corridors and ceilings in an indoor environment. Fukuda et al.
maintained in a dynamic and real-time environment. We propose a simple processing heuristic which can be employed by Yoon and Kweon (2001) for detecting symbols or trees for natural landmarks representation as in Maosen et al. (2005).

Then, we make use of a similarity evaluation process which is employed by Yoon and Kweon (2001) for detecting additional undefined landmarks with invariable colour changes. Andrade and Sanfeliu (2001a) implement a real-time landmark learning by neural network, and they also demonstrate techniques for indoor undefined landmarks (Andrade and Sanfeliu 2001b). Moreover, Zingaretti and Carbonaro (1999) make use of a learning templates optimisation, as Nguyen et al. (2004) for a navigation based in regions of interests using optical flow segmentation. Furthermore, these techniques are time dependent for static landmark perception during an exploration phase. Therefore, we propose a simple processing heuristic which can be maintained in a dynamic and real-time environment.

Moreover, Briggs et al. (2000) experiment with landmark matching that can be realised by obtrusive patterns on walls; Lamon et al. (2001) employ previously learnt features in indoor environment and Bianco and Zelinsky (2000) integrate a learnt model during an exploration stage. Despite recent works on single landmark tracking, the studies and algorithms of frequently observed landmark models for indoor unsupervised online learning as in Marsland et al. (2005) can implement sonar-based reinforced learning for a fuzzy set of rules such as the technique developed by Thongchai (2002). Also, some examples of alternative techniques are used by Nehmzow and Vieira-Neto (2004) for assigning colour histograms in interesting features and Vazquez-Martin et al. (2005) for detecting visual attention based in vertical regions.

Zoghlami et al. (1997) make use of landmark tracking schemes identify memorised features which have appeared previously using invariantly perceived geometric features. As it is presented in Quoc et al. (2004) and Luke et al. (2005), an additional memorisation phase can help to describe the environment in a slow robot motion and improve the quality of landmark description with SIFT descriptors created by Lowe (2004). Another approach from Watman et al. (2004) matches predefined templates of natural indoor features with a fast real-time detection.

The recent works contribute to landmark tracking performance and reduction of time during the recognition phase. Additionally, the work presented in this paper is also based on the main studies of landmark modelling for an indoor and dynamic tracking of natural and artificial landmarks. Such an approach is realised through a symbolic representation of image descriptors from a coloured blobs and edges image segmentation.

The rest of the paper is organised as follows. Section 2 describes the initial stage of image analysis for a landmark modelling process. Section 3 explains the process for interpreting predefined and real-time landmarks. Section 4 describes a landmark modelling procedure for detecting undefined environment and frequently appearing features. Then, Section 5 describes results obtained from identifying acquired landmarks in real time. Finally, a brief conclusion and future development are presented in Section 6.

2 Symbolic representation

This section describes an approach which represents robot surroundings using edges and blobs interpretation for landmark modelling. The proposed landmark modelling starts with a symbolic representation for acquiring, interpreting and matching perceived visual features, as shown in Figure 1. Particularly, this process makes use of surfaces with similar colour saturation (blobs) and edges (lines). The blob and edge segmentation is realised in a pixel classification process with a representative computational cost and further evaluated measuring process.

Figure 1 Robot visual recognition process (see online version for colours)

Therefore, the robot perception system is in charge of processing obtained images and generates an appropriate symbolic representation using colour and line segmentation. Specifically, image segmentation relies on pixel analysis of direction and colour characteristics. In the following sections, a symbolic representation process is described for environment sensing and landmark detection based on blobs and edges detection.

2.1 Blob detection

This section describes an image processing method for obtaining blobs using multi-layered segmentation created by Wasik and Saffiotti (2002), but by using a sub-divided square-centred image region delimited by a horizon line. After colour classification, each colour layer is contained in a table with blobs and shape-descriptive operators, as mentioned in Sonka et al. (1993), Gonzalez and Woods (2000), Morse (2002–2004) and Marshall (1994–1997). These image operators are described below:

- **Area**: The total number of pixels containing a blob or a set of candidate blobs for a landmark integration.
- **Perimeter**: This is calculated by counting initial and final pixel positions for each run in a landmark candidate blob, including the counting of top and bottom runs.
Real-time landmark modelling

- **Elongation**: This is the height and width ratio in a rotated minimal bounding box represented as a rotated rectangle, which is the smallest rectangle in which the shape fits.
- **Eccentricity**: This is the length ratio of the longest shape chord to the longest perpendicular chord represented as an ellipse containing a blob or possible blobs. The value is between 0 and 1.
- **Compactness**: This is the ratio of the perimeter squared to the landmark blob area or set of candidate blobs (perimeter
  \(2 = \text{area}\)).
- **Rectangularity**: This is how rectangular a shape is or how much it fills its minimal bounding box (objectarea = area).
- **Euler number**: This is used for a set of connected regions that can be obtained by using the number of regions minus the number of holes.
- **Connectivity**: This is the number of neighbouring features adjoining the region.

As a first recognition stage, landmark modelling transforms segmented images into image descriptors. Then, the complete blob generation process uses blob descriptors from coloured surfaces as shown in Figure 2.

**Figure 2** (a) Original images from robot perspective; (b) images segmented in colour layers; and (c) images with blobs identification (see online version for colours)

As a result, images are reinterpreted as object descriptors using nominal values. As shown in Figure 2, this stage interprets descriptive information obtained from colour layers through an image processing analysis.

### 2.2 Edge detection

Edge detection is a feature acquisition process for identifying pixels, which change in brightness value, from a YCrCb image. The purpose of this stage is to obtain a table with green and white transition properties from a CMOS image. The edges are processed from a horizontal and vertical operator mask and located in the table for further implementation, as described in Algorithm 1.

**Algorithm 1:** Algorithm for edge detection

<table>
<thead>
<tr>
<th>Input: Vertical/Horizontal Mask Operators ({F_{h,v}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: Image sequences ({I})</td>
</tr>
<tr>
<td>Output: Edge properties table ({E_{i,j}})</td>
</tr>
</tbody>
</table>

(For every image generated onboard)

for \(\{I_{i,j}\} \subseteq \{I\}\) do

for \(\{F_{h,v}^{i,j}\}\) on \(\{I_{i,j}\}\) do

(Horizontal and vertical edge masks are applied)

\(\{F_{i,j}\} = \sqrt{(F_{h}^{i,j})^2 + (F_{v}^{i,j})^2}\)

(The edges are assigned in an Edge Table)

\(\{E_{i,j}\} = \sum \{F_{i,j}\} \cdot \{I_{i,j}\}\)

(The edges are weighted and evaluated)

\(\{E_{i,j}\} = \text{AnaliseEdgeProperties}(\{E_{i,j}\})\)

End

End

The process is described as follows:

1. Edge detection begins by selecting a mask operator which evaluates values for each image with variable brightness. In this case a horizontal and vertical mask operator is used, as is illustrated in Figure 3. The mask operators are used in an area of interest located in the centre of image and horizon line.

2. An edge brightness value is obtained from pixels in the area of interest and they are accumulated in an *Edge indicator* array. Then, a product is obtained from the pixels’ brightness in order to get an edge indicator from the following equation:

\[ E_{i,j} = \sum F_{i,j} \times I_{i,j} \]  

where \(E_{i,j}\) is the edge indicator, \(F_{i,j}\) are filtered values function operators and \(I_{i,j}\) are pixels inside the image.

3. It is important to highlight that this method handles two dimensional edge analysis alongside image content. In a more appropriate computationally costless form, edge detection is obtained from filter masks and described in the following equation:

\[ F_{h,v}^{i,j} = \sqrt{(F_{h}^{i,j})^2 + (F_{v}^{i,j})^2} \]  

where \(F_{i,j}\) is the edge filter mask with super indices \(h\) for horizontal and \(v\) for vertical. In other words, the distance between points is reduced to:

\[ E_{i,j} = \sqrt{|E_{h}^{i,j}|^2 + |E_{v}^{i,j}|^2} \]  

4. Finally, gathered values are located in an edge table adjusted to the size of the processed image. Similarly, an edge table is organised by properties referencing colour types depending on the transition between green and white colours.
The output at this stage is an edge properties table which contains information for identifying position, colour type and confidence of the perception. The edge detection process analyses a change in the brightness value regardless of image noise presence, as shown in Figure 4.

Figure 4 In column (a) are the original images taken a robot; in the middle column (b) are images as they are appeared using image segmentation and in column (c) are images with Sobel horizontal and vertical edge detection (see online version for colours)

3 Static feature identification

This section describes an identification process of acquired environmental features which are already defined. The environmental representation is divided into colour-coded beacons and pitch lines. The identified objects are represented (Murch and Chalup, 2004; Xu et al., 2007) in a blob or line interpretation in a real-time landmark model. Similarly, Rofer and Jungel (2004) present a line positioning method based on edge segmentation and colour analysis for identifying natural and artificial landmarks. More specifically, implemented natural landmarks are corners and ‘T’ intersection lines, and artificial landmarks are coloured beacons. This landmark identification relies in a unique ID number as it is shown in Figure 5.

The object identification is a perception process that translates image processed information into landmarks from a perspective of the robot coordinate system. The process evaluates the perception of already defined landmark models from beacons and pitch lines. Moreover, both types of landmarks give a different amount of landmark information from results obtained in Section 5. In order to describe these dissimilarities, we presented tables for the perception of both and each type of landmarks based on frequency of visibility. The visibility can be interpreted as the reliability of the proposed landmark model for each specific landmark.

3.1 Blobs matching

After generating colour segmentation, the blobs’ properties are obtained from a colour table and are evaluated following shape and colour-coding conventions for feature identification. The process consists of analysing the connectivity between blobs in accordance to a specified invariant distance. However, each blob is previously filtered for individual and combined shape characteristics as illustrated in Algorithm 2.

Algorithm 2: Algorithm for identifying colour-code landmarks

<table>
<thead>
<tr>
<th>Input: Colour table {B}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: Recognised Landmarks {L}</td>
</tr>
<tr>
<td>foreach {B}_i \subset {B}_j do</td>
</tr>
<tr>
<td>isRangeHorizonLine ({B}_i);</td>
</tr>
<tr>
<td>(Check if it is inside evaluation area)</td>
</tr>
<tr>
<td>isValidColourRelation ({B}_i, {B}_j);</td>
</tr>
<tr>
<td>(Compare with closest blobs)</td>
</tr>
<tr>
<td>if {B}_i \cup {B}_j then</td>
</tr>
<tr>
<td>isValidBlobSize ({B}_i \cup {B}_j);</td>
</tr>
<tr>
<td>(Check size with rectangularity values)</td>
</tr>
<tr>
<td>isValidBlobDensity ({B}_i \cup {B}_j);</td>
</tr>
<tr>
<td>(Check compactness value for density measuring)</td>
</tr>
<tr>
<td>isValidPerimeterArea ({B}_i \cup {B}_j);</td>
</tr>
<tr>
<td>(Check perimeter and area are compatibles)</td>
</tr>
<tr>
<td>isValidOverlapping ({B}_i \cup {B}_j);</td>
</tr>
<tr>
<td>(Check overlapping using elongation axis equidistance)</td>
</tr>
<tr>
<td>isValidDistance ({B}_i \cup {B}_j);</td>
</tr>
<tr>
<td>(Check the distance between blobs)</td>
</tr>
<tr>
<td>isValidShapeProportion ({B}_i \cup {B}_j);</td>
</tr>
<tr>
<td>(Check for valid ellipsoid proportion using eccentricity)</td>
</tr>
<tr>
<td>isOld ({B}_i \cup {B}_j);</td>
</tr>
<tr>
<td>End</td>
</tr>
<tr>
<td>End</td>
</tr>
</tbody>
</table>
Real-time landmark modelling

In order to recognise an object from perceived image descriptors, we make use of filtering operators for identifying an expected object shape from each obtained image. The operators for locating artificial landmarks are described as follows:

1. check if the blob position is in the evaluation area
2. establish a relationship between blobs for comparison
3. check if a blob belongs to a colour scheme for predefined landmarks with its connectivity. If so,
4. validate blob size and density with threshold values
5. compare the blobs’ compactness compatibility for density measuring
6. determine if the blobs’ perimeter and area are compatible
7. check for overlapping using an elongation axis position from both blobs’ perspectives
8. check the distance between blobs using blob centre positions
9. compare eccentricity values using an ellipsoid and rectangularity proportion.
10. check for the last time the blob was detected.

Afterwards, a landmark description is obtained with an angle, the distance positioning, the time of perception and a confidence value for the estimation from the robot’s perspective as in the previous works of Lenser and Veloso (2003) and Murch and Chalup (2004). The confidence value is obtained by assigning:

1. lower values to high head and body speeds
2. higher values to landmark positioning in relation to the image centre.

As a result, it is possible to state that goals are the most visible artificial landmarks with a 42.19% of all observations. In other words, this landmark gives more information during all perception cycles, as shown in Table 1.

3.2 Line selection

The line detection stage is used for identifying pitch lines as natural landmarks following the work of line classification made by Herrero-Pérez et al. (2004) for corners and ‘T’ line intersection. The process of landmark identification is described as follows:

1. First, initial candidates are set up as being the closest edges to the outer area of image centre.
2. Second, a matching process is realised for selected edges within the same line and angle range.
3. Third, an additional matching process verifies jointly located corners perceived as a “T” line feature.
4. Fourth, feature information is extracted to determine the distance and angle of the robot, the time of observation and the confidence value in the perception.

Accordingly, the line detection perception process is illustrated in Figure 4 and its visibility performance in Figure 7. Moreover, the most visible landmarks account for 44.08% of the visibility range, as shown in Table 2.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Visibility ranges for artificial landmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Freq.</td>
</tr>
<tr>
<td>Landmark 4</td>
<td>1391</td>
</tr>
<tr>
<td>Landmark 1</td>
<td>1551</td>
</tr>
<tr>
<td>Landmark 3</td>
<td>1900</td>
</tr>
<tr>
<td>Landmark 2</td>
<td>1998</td>
</tr>
<tr>
<td>Landmark 5</td>
<td>2294</td>
</tr>
<tr>
<td>Landmark 6</td>
<td>2697</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Visibility values for natural landmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Freq.</td>
</tr>
<tr>
<td>Landmark 13</td>
<td>139</td>
</tr>
<tr>
<td>Landmark 14</td>
<td>148</td>
</tr>
<tr>
<td>Landmark 9</td>
<td>185</td>
</tr>
<tr>
<td>Landmark 7</td>
<td>206</td>
</tr>
<tr>
<td>Landmark 11</td>
<td>206</td>
</tr>
<tr>
<td>Landmark 8</td>
<td>218</td>
</tr>
<tr>
<td>Landmark 10</td>
<td>218</td>
</tr>
<tr>
<td>Landmark 3</td>
<td>234</td>
</tr>
<tr>
<td>Landmark 4</td>
<td>240</td>
</tr>
<tr>
<td>Landmark 1</td>
<td>263</td>
</tr>
<tr>
<td>Landmark 5</td>
<td>466</td>
</tr>
<tr>
<td>Landmark 2</td>
<td>467</td>
</tr>
<tr>
<td>Landmark 12</td>
<td>685</td>
</tr>
<tr>
<td>Landmark 6</td>
<td>837</td>
</tr>
</tbody>
</table>
4 Dynamic feature identification

In this section, a landmark modelling procedure is proposed for detecting undefined environment and frequently appearing features. The procedure is accomplished in two stages: (a) Entity identification and (b) Model evaluation. During the first stage, blobs that constantly appeared are classified by descriptive properties and identified as unique Entities. In the second stage, resulting Entities are converted into landmark Models.

Initially, entities represent any observed environmental features and are grouped in a linked list. Afterwards, this entity list is evaluated with similarity values of their properties for generating a dynamic landmark model.

The dynamic landmark model is based on constantly acquired visual features created during an exploration phase. Afterwards, the dynamic models are evaluated showing the robot’s capability for locating undefined environmental features. The stages are explained in the following sub-sections.

4.1 Entity recognition

The first stage of dynamic landmark modelling relies on feature identification from constantly observed environmental occurrences. This information is obtained from colour surface descriptors and contained in an entity structure. An entity is integrated by pairs or triplets of blobs with unique characteristics constructed from merging and comparing linear operators. The procedure reinterprets surface characteristics using the following operations:

1. obtain and validate an entity’s position from the robot’s perspective
2. get blobs’ overlapping values with respect to their size
3. evaluate the compactness values from blobs related to the same bounding box
4. validate eccentricity with a bias factor of 0.3 from related blobs.

The resulting structure contains sets of values of coloured-based surface properties from the robot perspective. In each entity, a colour surface also represents information which requires to be evaluated in the following model generation stage.

4.2 Model evaluation

The evaluation phase generates landmark models from recognizing information of similar entities. These landmarks models are constantly updated by the information obtained from environmental perception. In this sense, the evaluation process carried out in this phase has been designed to merge recent and previously defined information. The merging process implements a bubble sort comparison for evaluating similarities between obtained values. Lastly, it includes an elimination of the least useful entities. The procedure is described in Algorithm 3.

```
Algorithm 3: Process for creating a landmark model from a list of observed features
Output: Map of observed features \{E\}
Input: A collection of landmark models \{L\}
Landmark models are generated by the following operations:
Evaluate Colour Combination (\{E\} \{C\}):
Evaluate Blob Distances(\{E\}, \{L\}) di
Obtain Time Stamp(\{E\}, \{L\}) ti
Create Entity (\{C\}, \{d\}, \{t\}) j
If information is similar to an achieved model foreach
\{L\} MATCHON \{L\} do
if j ∈ \{L\} then
Update modelled values and Update \{L\}(j)
Increase modelled frequency Increase \{L\} frequency
If modelled information does not exist
End
else
Create \{L\}+1 Create model and Increase
\{L\}+1 frequency Increase modelled frequency
End
After one minute if time > 1 min then
MergeList(\{L\}) Select best models
End
```

The algorithm is iterated for each observed feature from a list of candidate entities \{E\} to obtain a collection of landmark models \{L\}. The process requires three operations for adapting an entity to a landmark model:
1. colour combination for checking entities with the same type of colours as a landmark mode
2. descriptive operator matching with a ±0:3 range ratio from defined models
3. time stamp and frequency for filtering long-lasting models using a removing and merging process of non-leading landmark models.

Then, the merging process is realised comparing evaluated similarity values. Specifically, similarity values are evaluated using equation (4) and the probability of perception using equation (5):

\[
P(i, j) = \frac{M(i, j)}{\sum_{k=1}^{n} M(k, j)}
\]

\[
M(i, j) = \sum_{i=1}^{n} E(i, j, l)
\]
where \( N \) represents the list of achieved models, \( i \) is the sampled entity, \( j \) is the compared landmark model, \( M(i, j) \) is the landmark similarity measure obtained from matching an entity’s descriptors and assigning a probability of perception as described in equation (6), \( P \) represents the total descriptors, \( l \) is the landmark descriptor and \( E(i, j, l) \) is the probabilistic Euclidian distance of each landmark model compared, estimated using equation (7):

\[
\sum_{k=1}^{N} M(k, j) = 1 \tag{6}
\]

\[
E(i, j, l) = \sqrt{\sum_{m=1}^{L_L} \frac{(i_m - l_m)^2}{\sigma_m}} \tag{7}
\]

where \( L_L \) refers to all possible operators from the current landmark model, \( \sigma_m \) is the standard deviation for each sampled entity \( i_m \) in a sample set and \( l \) is a landmark descriptor value.

At last every minute an elimination of 10% of the least useful landmark candidates is performed. These eliminated entities allow to minimise isolated landmark models and update the frequency on constantly appeared landmark models.

After all stages are achieved, the updated list of entities becomes candidate landmark models. The list of landmark models can describe constantly appearing and undefined environmental features by the robot perception. The process contemplates coloured blobs and information obtained from their invariant perception between them.

5 Experiments

The performance for angle and distance is evaluated in three experiments. For the first and second experiments, the robot is placed in a fixed position on the football pitch where it continuously pans its head. Thus, the robot maintains simultaneously a perception process and a dynamic landmark creation. Figures 8 and 9 show the positions of 1683 and 1173 dynamic models created for the first and second experiments over a duration of five minutes.

Table 3

<table>
<thead>
<tr>
<th>Distance</th>
<th>Angle</th>
<th>Error distance</th>
<th>Error angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>489.02</td>
<td>146.89</td>
<td>256.46</td>
</tr>
<tr>
<td>SD</td>
<td>293.14</td>
<td>9.33</td>
<td>133.11</td>
</tr>
</tbody>
</table>

Table 4

<table>
<thead>
<tr>
<th>Distance</th>
<th>Angle</th>
<th>Error distance</th>
<th>Error angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>394.02</td>
<td>48.63</td>
<td>86.91</td>
</tr>
<tr>
<td>SD</td>
<td>117.32</td>
<td>2.91</td>
<td>73.58</td>
</tr>
</tbody>
</table>

In the third experiment, landmark models are tested during a continuous robot movement. This experiment consists of obtaining results at the time the robot is moving along a circular trajectory with 20 cm of bandwidth radio, and whilst the robot’s head is continuously panning. The robot is initially positioned 500 mm away from a coloured beacon situated at 0 degree from the robot’s mass centre. The robot is also located in between three defined and one undefined landmarks. Results obtained from dynamic landmark modelling are illustrated in Figure 10. This experiment required 903 successful landmark models detected over five-minute duration of continuous robot movement and the results are presented in Table 5.

Figure 9 Landmark model recognition for Experiment 2 (see online version for colours)

Initially, newly acquired landmarks are located at 500 mm and with an angle of \( 3\pi/4 \) radians from the robot’s centre. The results are presented in Tables 3 and 4. Both tables illustrate the mean (\( \bar{x} \)) and standard deviation (\( \sigma \)) of each entity’s distance, angle and errors from the robot’s perspective.
R. Samperio and H. Hu

Table 5

<table>
<thead>
<tr>
<th>Distance</th>
<th>Angle</th>
<th>Error distance</th>
<th>Error angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>305.67</td>
<td>12.67</td>
<td>90.30</td>
</tr>
<tr>
<td>SD</td>
<td>105.79</td>
<td>4.53</td>
<td>54.37</td>
</tr>
</tbody>
</table>

The results show magnitudes for mean (\( \bar{x} \)) and standard deviation (\( \sigma \)), distance, angle and errors from the robot perspective.

Each of the previous images illustrates landmark models generated during experimental execution, represented as the accumulated average of all observed models. In particular for the first two experiments, the robot is able to offer an acceptable angular error estimation in spite of a variable proximity range. The results for angular and distance errors is similar for each experiment. However, landmark modelling performance is susceptible to perception errors and obvious proximity difference from the perceived to the sensed object.

Previous images illustrate all generated landmark models during experimental execution. Also darker marks on all graphs represent an accumulated average from observed models. The average entity of all models presents an acceptable angular error in a real-time visual process. An evaluation of the experiments is presented in Box and Whisker graphs for error in position (Figure 11), distance (Figure 12) and angle (Figure 13).

Therefore, the angle error is the only acceptable value in comparison with distance or positioning performance. Also, the third experiment shows a more comprehensive real-time measuring with a lower amount of defined landmark models and a more controllable error performance.

6 Conclusions and future work

In this paper, we presented an implementation of real-time visual landmark perception for a walking robot. This approach interprets an object using symbolic representation of environmental features, whether natural, artificial or undefined. The experimental results show that undefined landmarks can be recognised accurately during static and moving robot recognition sessions. Further work will focus on adapting image descriptors for invariant scale description during real-time processing and reinforce filtering of the recognised models.

Acknowledgements

We would like to thank TeamChaos (http://www.teamchaos.es) for facilities and capabilities of their programming work and the technical team within School of Computer Science and Electronic Engineering at Essex University (http://essexrobotics.essex.ac.uk) for their support and help in this research. Also, part of this research was also supported by a CONACyT (Mexican government) scholarship with reference number 178622.

References


