From Pixels to Objects: Enabling a spatial model for humanoid social robots

Dario Figueira  Manuel Lopes  Rodrigo Ventura  Jonas Ruesch

Abstract—This work adds the concept of object to an existent low-level attention system of the humanoid robot iCub. The objects are defined as clusters of SIFT visual features. When the robot first encounters an unknown object, found to be within a certain (small) distance from its eyes, it stores a cluster of the features present within an interval about that distance, using depth perception. Whenever a previously stored object crosses the robot’s field of view again, it is recognized and mapped into an egocentric frame of reference. This mapping is persistent, in the sense that its identification and position are kept even if not visible by the robot. Features are stored and recognized in a bottom-up way. This work creates the foundation for a way of linking the bottom-up attention system with top-down, object-oriented information provided by humans.

I. INTRODUCTION

For humanoid social robots to autonomously interact with our human structured environments, they must be endowed with the capacity of perceiving objects as such. However, robot sensory apparatuses only provide raw sensory data. Taking vision as a sensor, how can it bridge the gap between raw pixels and the concept of objects? Moreover, how can it realize their relative positions within the surrounding environment, even when they are temporarily out of the cameras field of view?

The system presented here addresses these problems, by taking a bottom-up, developmental approach. The developed module builds upon an existing low-level attention system. The previous work provides a salience map with respect to a robot-centric coordinate system (ego-sphere) [1]. This saliency map, together with an inhibition of return mechanism (IOR), allows the robot to saccade from salient point to salient point. However, these salience points correspond to preattentive features, e.g., movement, color, and shape, that do not incorporate the concept of object.

The goal of the work presented here is to endow the robot with the capability of learning and recognizing objects. By integrating this capability into the existing architecture, the attention module will be able to acknowledge the salience of known objects, because they are recognized as such. Moreover, the capability of recognizing known objects by visual features paves the way for higher level modules, such as language, to implement complex cognitive functions.

Fig. 1. Ego-sphere: a spherical map of the surroundings

A spatial model for the robot is here understood as a model representing the environment surrounding it, namely the known objects, together with their physical positions relative to the robot.

In this work we add to the spatial saliency map implemented in [1] and endow the system with an interpretation of its surroundings. The system now maps known visual objects, so it can know where they are after looking away. Instead of the short-time memory of the previous works [1], [3], the system remembers where important objects are at longer time scales.

We implemented an algorithm to automatically store to a database new objects that are close to the eyes of the robot (the cameras). To do so, we compute a depth perception [4], [5] of the image to determine if something is in close proximity and under the robot’s scrutiny. If so, its representation is stored to a database to be later remembered and recognized.

We chose the Scale Invariant Feature Transform (SIFT) [6] algorithm to enact our recognition. We also used this algorithm to match corresponding points in pairs of stereo images, thus computing the disparity or depth of these points.

This work was partially supported by FCT (ISR/IST plurianual funding) through the POS,Conhecimento Program that includes FEDER funds.

D. Figueira, M. Lopes and R. Ventura are with Institute for System and Robotics (ISR), Instituto Superior Técnico, TULisbon, Portugal {dfigueira,macl,yoda}@isr.ist.utl.pt

J. Ruesch is with the Artificial Intelligence Laboratory, Department of Informatics, University of Zurich, Switzerland ruesch@ifi.uzh.ch

The robot considered here is the iCub humanoid robot [2]. However, just the head and torso modules were employed. While the head has the 6 degrees of freedom, the torso is fixed. Being the coordinate system anchored to the torso, it is a robot-centered coordinate system of the world.

To model space, and to fulfill the goal of helping an agent commute automatically its attention focus from recognized object to recognized object, we chose to model the environment as a salience map [3]. We project the surrounding space and objects into a sphere centered in the neck of the robot, an egocentric sphere or ego-sphere, as defined in [1] (Figure 1).
By using this algorithm to detect distance we defined a new object as a cluster of SIFT features in close proximity to the cameras, while already saved objects are detected continuously in the input images by the SIFT algorithm. Finally, the recognized objects in the robots field of vision are inserted in the egocentric map.

In the next section we shall describe our spatial model in a general sense leaving out the implementation details to be discussed in the third section. Then, in the forth section, we describe an experiment in which the functioning of our work is illustrated and present images to prove its worth. Finally, we finish with our conclusions and future work.

II. ARCHITECTURE

The architecture, displayed in Figure 2, has several interconnected modules to form a sensing-deliberation-actuation chain. It is motivated on the Itti and Koch model [7] where stimuli from various sources is represented and combined in a single salience map. The resulting egocentric map is processed and its maximum is selected as the point that wins the robots attention next. Finally the robot’s visual focus is switched to the new selected attention point.

![System Architecture Diagram](image)

The ego-sphere, in addition, keeps a short-memory of the previously looked upon positions, in the form of an inhibition-of-return mechanism (IOR). The IOR information reduces the salience levels of the already observed locations. The resulting behavior is the capability of the robot to fully explore its environment without being stuck on the absolute saliency maxima of the salience maps.

Our work adds a level of abstraction to the previous architecture: the concept of object, i.e., the agent can now learn about and recognize objects. But since this work is to be used by an autonomous agent, by definition it must be self-sufficient. Consequently, the problem of automatizing the decision of when and what to save to the database, arises. This problem is solved by using a special depth filter that calculates the depth, and segments the object in the image. When an object is detected in close proximity it triggers the creation of a new object in the memory, given that we don’t recognize it already. Since we operate on the assumption that the agent will have arms and hands, and it will use them to manipulate objects when exploring its environment, the threshold distance to save new objects will be similar to the arms length, i.e., the expected object distance to the agents eyes when a new object is being inspected by the robot.

Although at the present there is no input to describe the objects saved, the objects are saved just with a “label” that is simply a number (the order of appearance). When the robot has the possibility to ask humans around him for names to the objects he is discovering and storing, new possibilities for the creation of other modules that rely on the existence of these named objects arise.

III. IMPLEMENTATION

In this section we provide the details to perform object segmentation and recognition.

Many different approaches have been used in computer vision to enable recognition, for instance, eigenspace matching has been used successfully by Schiele [8], others have used Speeded Up Robust Features (SURF) [9], and many have benefited from David Lowe’s Scale Invariant Feature Transform (SIFT) [6]. Our approach, the latter approach, solves both problems of object segmentation and recognition. We chose SIFT over eigenspace matching for reasons such as invariance to scale and excelling in cluttered or occluded environments (as long as three SIFT features are detected, the object is recognized). And while the SURF algorithm is faster and performs generally well, SIFT’s recognition results are still superior [10]. The setback about using this algorithm is that it takes a lot of processing time, the most efficient implementations are not able to run it in real time (24 FPS) [11].

SIFT [6] is an algorithm that extracts, features from an image. These features are computed from histograms of the gradients around the key-points, and are not only scale invariant features, but also affine transformations invariant (e.g., rotations invariant). Furthermore, they are robust to changes in lighting, robust to non-extreme projective transformations, robust up to 90% occlusion and are minimally affected by noise. We use the SIFT algorithm to enable the recognition in our system because of all this powerful characteristics. Due to the nature of the SIFT features, its second drawback is the inability to extract features from a texture-less object, as is shown in Figure 3, few or no features, in yellow dots, are found in areas with homogeneous color, such as on the table, on the ground, or on the wall. Moreover, since this algorithm was designed to work with planar surfaces, to be able to correctly recognize 3D objects, we save several snapshots of the objects in different perspectives. An object is represented
in the database by several sets of SIFT features, together with the relative positions of the features to each other. Each set representing a different snapshot of an object.

![Fig. 3. Example of SIFT feature extraction; the yellow dots correspond to the extracted features positions.](image)

### A. Depth Perception

The common way to determine depth, with two stereo cameras, is by calculating a disparity map. Disparity is simply the subtraction of the 2D coordinates of corresponding points (from the left image to the right image). A dense disparity map consists in a matrix, of the size of the images, in which the value of each coordinate indicates how many pixels had to be shifted until we find the match of our current pixel in the other image. Some ways to match corresponding points can be: pixel by pixel probabilistic matching with a Bayesian formulation [5]; or histogram matching of the neighborhood of the pixel [12].

The SIFT features, with their invariance and robustness, enact a way to solve the problem of matching corresponding points in stereo images. We generate a sparse disparity map by extracting the SIFT features from stereo images, and look for matches between both sets. Assuming that the robot’s eyes are roughly aligned in the horizontal (i.e., mis-alignment of under 30 pixels) we compute the disparity between matching features from the pair of stereo images. Matches that have a high horizontal disparity are assumed to be part of an object in close proximity to the robot’s face and matches with low horizontal disparity belong to the “background.” Matches with high vertical disparity or negative horizontal disparity are outliers, i.e., bad matches.

Using a batch of real images we get the following results summarized in Table I. In the first column we have the number of features detected in the left image, in the second column we display the number of matches found between the left image features and the right image features, while on the third column we show how many of those matches were bad matches, outliers.

<table>
<thead>
<tr>
<th>Images</th>
<th>Features</th>
<th>Matches</th>
<th>Outliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>27564</td>
<td>9545</td>
<td>165</td>
</tr>
<tr>
<td>percentage</td>
<td>100</td>
<td>34.6</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Comparing the extracted features of different images in different resolutions, a threshold for the horizontal disparity $T_h$ was found empirically to be the width of the image divided by 6.4. Moreover, the vertical threshold $T_v$ to determine outliers was also empirically found to be the height of the image divided by 16.

If the matches between detected features are “close enough” (each match having its horizontal disparity greater than the threshold), the group is stored to the database as a new object. Only the features that are correctly matched between the two stereo images with high horizontal disparities are stored, because only these features are believed to belong to the close object. For instance, the features from the background being seen by a hole in the object are discarded.

Figure 4(a) and Figure 4(b) exemplify in blue crosses the features that are correctly matched between the two stereo images as being the same, and therefore store to the database as a new object (if not recognized as part of an already known object).

### B. Recognition

To decide upon the presence of an object in the image, SIFT relies on a voting mechanism that is implemented by a Hough transform. Defining pose as the position, rotation and scale of an object, each match votes on an object-pose pair in the image. The Hough transform is computed to identify clusters of matches belonging to the same object. Finally, a verification through least-mean-squares is conducted for consistent pose parameters along all matches (verifying if the matches found have correct relative positions).

After experimenting with several objects, having the robot store them to the database and then holding them farther and farther away, the algorithm is able to recognize them until roughly two meters away, when the number of extracted features starts to be too few. Of the many features stored in the database and shown in blue crosses in Figure 5(a), only the few extracted ones, depicted in purple filled squares, are needed to recognize the object (encased in a red frame) in Figure 5(b).

### C. Database and Mapping

New objects are stored into a database, which links object identifiers to sets of SIFT features. When known objects are encountered in the environment, their positions are mapped into the ego-sphere [1]. Thus, an object representation is stored in the database, while their positions, whenever recognized by the robot, are represented in the ego-sphere.
The ego-centric saliency map used for attention selection is obtained from the composition of several specialized maps: a visual map, containing saliency information extracted from visual features (e.g., motion, color), and an auditory map, obtained from sound stimuli captured by the robot’s microphones [1]. The saliency information stored in these maps is continuously decayed, according to a forgetting factor. In order to integrate the system described in this paper with the attention selection mechanism, the recognized objects are projected onto a third map. This map, together with the other two, contributes for the ego-centric saliency map. As the others, this map is also subject to a continuous decay of its information, but with a much longer forgetting factor.

IV. RESULTS

One of the experiments set up to show the correct recognition and mapping consisted of: showing two objects, in turn, for it to learn and store to the database, a book and a magazine cover; Then setting them up in front of him separated wide enough so that when the robot’s attention would be on one of this objects his field of vision will not cover the second object as well; Observing the resulting
behavior.

The robot, upon recognizing the both previously known objects in Figure 6, adds interest peaks in the ego-sphere [1] (Figure 7(a)). The Attention Selection [1] module tells the robot where to look, and he fixes his attention in the first object (Figure 7(b)). After some time of intense scrutiny a inhibition region is added to the Inhibition-of-Return map [1] (Figure 8(a)) which nudges the attention selector to continue exploring the environment of interesting points, the recognized objects. The Attention Selection module then indicates the robot to look at the now most salient region in the memory, the second blob in the ego-sphere, the second object in Figure 8(b).

![Fig. 6. Recognizing objects in the environment; Red: recognized first object, green: recognized second object](image)

Fig. 6. Recognizing objects in the environment; Red: recognized first object, green: recognized second object

V. CONCLUSION

The objective of this work is to implement a spatial model of the space surrounding the iCub robot that included the salient objects which the robot encounters in its explorations. This model is used to commute the robot’s attention focus automatically between objects, while not being dependent on the robots field of vision or on the objects visibility conditions.

We implemented this spatial model of the environment, we mapped the recognized objects of the surroundings, introducing salience peaks on the ego-sphere [1] in their positions. The robot iCub now also explores its environment focusing its attention in the recognizable objects around him.

With the long-term memory implemented, the objective of making this spatial model non-dependent on the robot’s field of vision was achieved. As depicted in the results, the robot returns its focus to previously observed objects that were at the moment not in its line of sight.

A. Future Work

At the present time we only use the positions of the recognized objects in our work. The also provided names of the objects and average disparity of the recognized matched features offer the foundation for plenty ideas for future work. One avenue is tracking specific objects in the environment. Another, is the search for specific objects in the surroundings to ascertain its existence or not. Additionally, the estimation of how far an object actually is, in absolute terms, is another avenue of possible work to be done.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Alexandre Bernardino’s reviews, comments and insights in the matter of Depth Perception.

REFERENCES


Fig. 8. Inhibition-of-return forces the robot to explore the rest of the environment.


