

Embodied Language Acquisition: a Proof of Concept

Aneesh Chauhan¹, Amanda Nascimento³, Bruno Werneck⁴, Luís Seabra Lopes^{1,2}

¹IEETA, Universidade de Aveiro, Aveiro, 3810-193, Portugal

²DETI, Universidade de Aveiro, Aveiro, 3810-193, Portugal

³Instituto de Computação, Universidade Estadual de Campinas, Avenida Albert Einstein, 1251 Caixa, Postal 6176, 13083-970, Campinas, SP, Brasil

⁴Grupo Olimpo, Rua T-55, n° 740, Setor Bueno, 74223-230, Goiânia, Goiás, Brasil
{aneesh.chauhan, lsl}@ua.pt, {amanda.nascimento, brunowerneck}@gmail.com

Abstract. For robots to interact with humans at the language level, it becomes fundamental that robots and humans share a common language. In this paper, a social language grounding paradigm is adopted to teach a robotic arm basic vocabulary about objects in its environment. A human user, acting as an instructor, teaches the names of the objects present in their shared field of view. The robotic agent grounds these words by associating them to visual category descriptions. A component-based object representation is presented. An instance based approach is used for category representation. An instance is described by its components and geometric relations between them. Each component is a color blob or an aggregation of neighboring color blobs. The categorization strategy is based on graph matching. The learning/grounding capacity of the robot is assessed over a series of semi-automated experiments and the results are reported.

1 Introduction

Robotics is often described as “the intelligent connection of perception to action” [1]. Although we are far off from an all-satisfying definition of intelligence, there is a consensus amongst researchers over a set of properties any intelligent system should exhibit [11]: ability to learn, ability to reason, responsiveness to external stimuli, and ability to communicate. Sentience can be described as a mix of these components.

Motivated by the need to create user friendly robots, there is an increasing interest in the field of robotics to build sentient robots. The state of the art robots (e.g. Rhino [2], Keepon [8], Leo [18]) have shown a certain degree of sentience. This paper introduces one such robot, which was built primarily to study the process of grounding a language.

For extended adaptability, flexibility and user-friendliness, a robot will need to understand, use and share the language with its human users, requiring human users to act as teachers or mediators ([12], [16]). Here language is being considered as a cultural product [10], which is created/acquired through social transmission. Both horizontal ([15], [17]) and vertical modes ([7], [16]) of language transmission have been studied in populations of robots.

In the present paper, language acquisition follows vertical transmission, where a human instructor gradually teaches the robot the names of the objects in their shared field of view. The presented robot grounds the names of these objects by associating them with their respective concept descriptions. The instructor can also provide corrective feedback in case of misclassification.

Language processing involves manipulation of symbols. By symbol it is meant a pattern that represents some entity in the world by association, resemblance or convention [14]. These symbols have no meaning, unless they are grounded in reality [5]. The robot presented in this paper grounds the names of the categories by associating them with sensor-based concept descriptions.

Language acquisition is an open-ended process. Categories as well as their corresponding names will have to be acquired incrementally. An open-ended category learning and language acquisition system must be supported by long-term learning capabilities. Long-term learning is usually included at the core of cognitive theories [6]. Work reported in this paper was built atop a lifelong category learning platform previously described in [12]. This paper introduces a novel component-based approach to object representation, category learning and recognition. The performance of this method was analyzed over a series of experiments.

The presented robot comprises a robotic arm and a camera to help it perceive as well as operate in its surroundings. Action capabilities include linguistic response to the user and the manipulation of objects by the robotic arm.

This paper is structured as follows: Section 2 describes the robot's hardware and software architecture. Section 3 provides an overview of robotic arm control. Section 4 details the novel component-based learning and categorization methodologies. Section 5 reports the conducted experiments and Section 6 presents the conclusions.

2 Agent Architecture

Physical Architecture. The robot is physically embodied with a robotic arm and a camera, while all its cognitive functions are carried out on an attached computer. The robot's world includes a user, a scenario visually shared with the user and objects whose names the user may wish to teach.

The robotic arm is an SG6-UT1 educational arm manufactured by Crust Crawler Robotics (Fig. 1). It is shipped with a PSC-USB board (Parallax Servo Controller with USB interface) and contains 6 servos for 5 arm joints and a gripper, comprising 6 degrees of freedom. The pay load of this arm is around 400 grams. A client interface was developed for the PSC-USB enabling to control the arm directly from the computer. The camera is an IEEE1394 compliant digital camera². It is placed in a fixed position over the work area of the robotic arm. This device acts as the primary perception instrument for the robotic agent. The computer runs the human-robot

¹ SG6-UT - a 6 degree of freedom robotic arm, supplied online at:
<http://www.crustcrawler.com/>

² IEEE 1394 compliant firwire camera called "fire-i", supplied online by Unibrain at:
http://www.unibrain.com/Products/VisionImg/Fire_i_DC.htm

communication interface, the visual perception module, learning and recognition systems and robotic arm control functions.

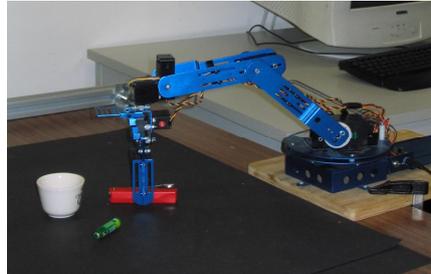


Fig. 1 – The CrustCrawler SG6-UT robotic arm

Software Architecture. The robot’s software includes a perception system, a learning and recognition system and an action system. The user, not visible to the robot, will act as the instructor. Fig. 2 illustrates the complete system architecture and the relations between its various parts.

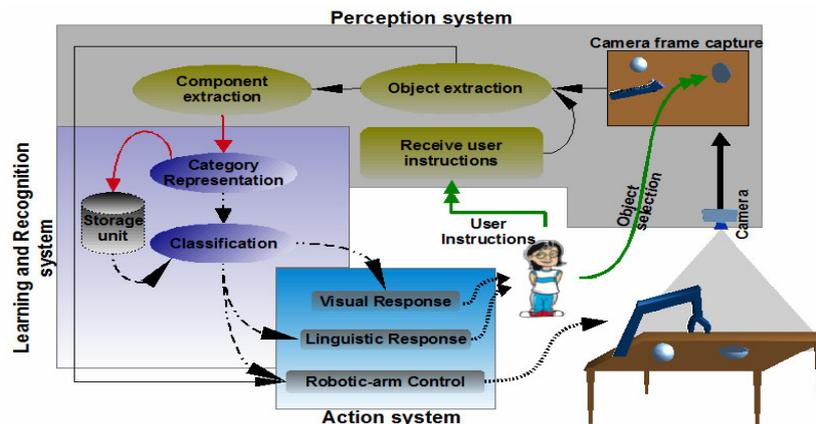


Fig. 2 – System architecture

Perception System. The keyboard, the mouse and the camera are the sources of input to the perception system. The perception system extracts the image of the object selected by the user (by mouse clicking). The publicly available library for vision routines `opencv`³ has been used for capturing the frames from the camera and extracting the object from the image. If the robot has been instructed to perform either learning or recognition, the extracted object is further processed to find its visual components. A novel strategy based on object’s color and shape information was implemented for extracting these components (see Section 4).

³ <http://opencv.willowgarage.com/wiki/>

Learning and Recognition System. An original approach to category learning using graph-based object and category representation is proposed. Section 4 is devoted to a thorough explanation of these modules.

Action System. The primary task of the action system is to provide appropriate feedback to the user. Depending on the user instructions and the information received from the previous cognition modules, the action system can perform actions of the following types:

1. Linguistic response: provide the classification results back to the user,
2. Visual response: Visually report the results of the “search and locate” tasks, and
3. Manipulation actions: the robotic arm manipulates the objects in the robot’s environment (details on arm dynamics and control are discussed in Section 3).

Human-Robot Interaction. Using a simple menu-based interface, the user can select (by mouse-clicking) any object from the visible scene, thereby enabling shared attention. The user can choose the instruction she/he wants the robot to follow as well as teach new words. The following instructions are supported:

1. Teach the category name of the selected object;
2. Ask the category name of the selected object, which the agent will predict based on previously learned knowledge;
3. If the category predicted in the previous case is wrong, the user can send a correction.
4. Ask the robot to pick the selected object using the robotic arm;
5. Provide a category name and ask the robot to locate an instance of that category;
6. Provide a category name and ask the robot to pick an instance of that category using the arm;
7. If the object identified by the robot in the two previous cases does not belong to the requested category, the user can provide its true category;
8. Once an object has been picked, select a location and ask the robot to place the object there.

In the future, a speech based communication interface will be supported.

3 Robotic Arm Control

To pick an object using the robotic arm, it is necessary to know the exact placement and the orientation of that object. Since the camera is the only available sensor, the suitable picking position is found by further processing the extracted object image. A linear mapping between camera and arm coordinates is assumed.

Finding Grasp Position and Orientation. The object should be grasped such that it does not rotate or fall down. A suitable picking location (see Fig 3) is found by taking the following conditions into account:

1. The gripper should be centered at the geometric center of the object
2. The distances $A1B1$, $A2B2$ and $A3B3$ must be less than the maximum gripper width;

3. The areas of the triangles formed by the points $A_1A_2A_3$ and $B_1B_2B_3$ must be less than a threshold (ideally zero);
4. Angles α and β between the gripper ends and lines A_1A_3 and B_1B_3 respectively, should be close to zero.

Depending on the object's structure, there will be more than one possible location to pick it up. The first suitable location to pick the object is used. Once the joint angles have been obtained, these values are passed to the arm's servo controller, for it to pick up the object.

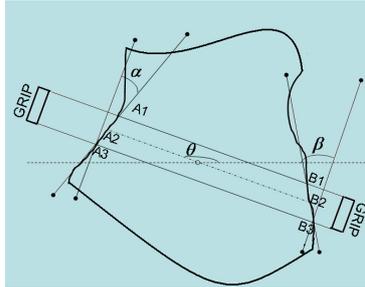


Fig. 4 – A hypothetical object and one possible gripper position.

Arm Kinematics. Fig. 4a shows a simplified model of the arm. R_1 , R_2 and R_3 are the arm link lengths and α_1 , α_2 , α_3 and α_4 are the joint angles to be determined. α_1 can be easily found by using basic trigonometry:

$$\alpha_1 = \tan^{-1}\left(\frac{y}{x}\right)$$

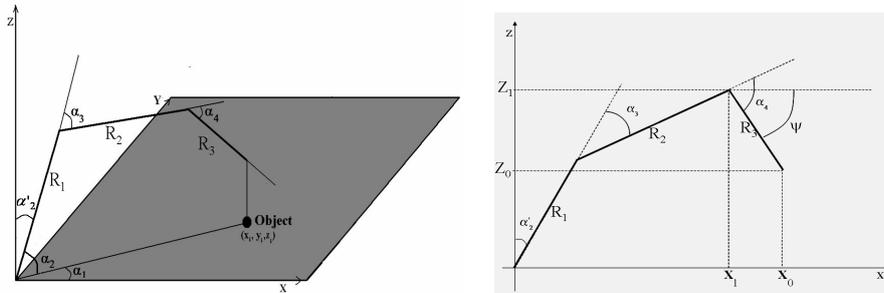


Fig. 3 – a) A simplified model showing arm joints and the object location; b) Robotic arm plane

To calculate the rest of the angles, consider the plane of the robotic arm, as in Fig. 4b. x_0 and z_0 are given by:

$$\begin{aligned} x_0 &= R_1 \cdot \sin(\alpha_2) + R_2 \cdot \sin(\alpha_2 + \alpha_3) + R_3 \cdot \sin(\alpha_2 + \alpha_3 + \alpha_4) \\ z_0 &= R_1 \cdot \cos(\alpha_2) + R_2 \cdot \cos(\alpha_2 + \alpha_3) + R_3 \cdot \cos(\alpha_2 + \alpha_3 + \alpha_4) \end{aligned} \quad (1)$$

We have two equations and three variables to determine. To solve this problem, the angle Ψ between the third link and the horizontal coordinate is forced to be $\pi/2$. This means, the wrist of the robotic arm will always be perpendicular to the base of the table. This causes no significant problem, since it is the best position for a successful grasp. Given this, Z_1 and W_1 can be determined by:

$$\begin{aligned} Z_1 &= Z_o + R_3 \\ X_1 &= X_o \end{aligned} \quad (2)$$

Using (1) and (2), α_3 and α'_2 can be determined as:

$$\begin{aligned} \alpha_3 &= \pm \cos^{-1} \left[\frac{(Z_1^2 + X_1^2 - R_1^2 - R_2^2)}{(2R_2R_1)} \right] \\ \alpha'_2 &= \pm \tan^{-1} \left[\frac{X_1(R_1 + R_2 \cos(\alpha_3)) - Z_1R_2 \sin(\alpha_3)}{Z_1(R_1 + R_2 \cos(\alpha_3)) + W_1R_2 \sin(\alpha_3)} \right] \end{aligned}$$

Now the final unknown angle can be given as:

$$\alpha_4 = \psi - \alpha'_2 - \alpha_3$$

4 Category Learning and Recognition

This paper uses a component-based approach to category learning and object categorization. An object instance is represented by its components and geometric relations between them. The relations between components are represented using non-directed labeled graphs. Taking into consideration the attributes of the nodes and edges, a novel strategy for categorization is also proposed.

Object and Category Representations. An object is represented using a graph model of spatial arrangement of its components, referred as *graph of components*. In such graph, each component is represented by a different node. The relation between a pair of components is represented by a non-directed edge connecting their respective nodes. Fig. 5c shows an example graph of components for the toy train object. Formally, a graph of components can be described as:

$$G = \langle C, R, A \rangle$$

where the three items are the following:

- $C = \{c_i\}$, $i = 1, \dots, n_C$, is the set of all components of the object, represented by nodes in the graph;
- $R = \{(c_i, c_j)\}$, $c_i \in C$, $c_j \in C$, is the set of existing contact relations between components, represented as edges in the graph; and
- $A = \{(c_i, c_j, c_k)\}$, is the set of all ternary relations between components such that $(c_i, c_j) \in R$, $(c_i, c_k) \in R$.

The properties of a graph of components are derived from the organization of the components inside the object. Each component c_i (*i.e.* each node of the graph) is characterized by the following features:

- a_i - relative area, given by the ratio of number of pixels in the component to the total number of pixels present in the object;

- r_i – relative radius, given by the ratio between the radius of a circle, with the same area as that of the component, and the radius of a circle with the same area as that of the whole object; and
- d_i - relative degree, given by the ratio between the degree of a node and the total number of nodes in the graph.

The contact relation $(c_i, c_j) \in R$ between connected components c_i and c_j is characterized by a single feature:

- d_{ij} - relative distance between two connected components c_i and c_j . This is measured as the ratio of the distance between the geometric centers of two connected components and the diameter of the object.

Finally, each ternary relation $(c_i, c_j, c_k) \in A$ is also characterized by a single feature:

- a_{ijk} - the relative angle between edges (c_i, c_j) and (c_i, c_k) . The ratio of the smaller angle between the two edges with respect to π is taken into use.

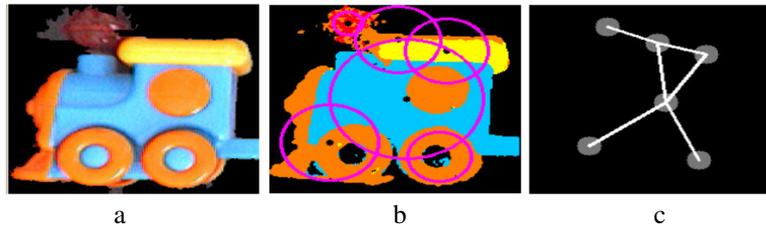


Fig. 5 – a) The “toy train” object; b) Extracted components and the circles with the same area as that of the components, centered at the geometric centres of the components (the components are extracted in HSV color space, but here, for visual perception the second image is converted back into RGB); c) Graph of components of the given object

The graph features listed above give a scale, translation and rotation invariant representation of an object. An instance-based approach is adopted for category representation. The learning module stores the object instance (i.e. the graph’s structural information as well as all its features) in an internal database. A category is represented by one or more instances that belong to that category.

Component Extraction. This work uses the HSV (Hue, Saturation and Value) color space for locating and extracting object components. A list of the most relevant color ranges in the given object is initially computed: $[R_1, \dots, R_n]$. Each range R_i is represented as a tuple (a, b, m, A) , where a and b are respectively the start and end values (hue) of the color range, m is the most frequent color value and A is the area of the object (number of pixels) in that color range. These ranges are found using the algorithm described in [13]. Once the color ranges have been found, the image is modified by replacing all the hue values in a range R_i by its corresponding most frequent color m .

The object image is then processed such that the neighboring pixels with same hue are aggregated to form the initial set of components. Connections are established between any two components physically in contact with each other (i.e. with adjacent pixels). These components are further merged based on the following steps:

1. Compute the geometric center and the average radius of each component and abstract the component as a circle with that radius and center;
2. If the circles of any two components overlap considerably (covering more than 90% of at least one of the components) they are merged to form a single component.
3. If the circles of any two components overlap, covering an area between 35% and 90% of at least one of the components, check for hue similarity. If the hue values present in one component are similar to the ones present in the other, merge the components;
4. Repeat these steps until no components can be merged.

Figure 5 displays the components and contact relations for the “toy train” object. A contact relation between any two components c_i and c_j is found based on the following criterion:

1. If c_i and c_j are physically connected or extremely close to each other, there is a contact relation between them. If any two border pixels, one from each component, are closer than a threshold (3 pixels in the implementation), then the components are considered connected.
2. If a component is found to have no neighbor using the previous rule, then it is considered connected to its closest component. In this case, for simplicity, the distance between (the borders of) the circles, centered in the geometric centers of the components, is measured.
3. The graph structure is derived directly from the contact relations. However, this strategy has the drawback that the graph may not always be completely connected. That is, instead of one connected graph, the graph of components may end up as a set of two or more connected subgraphs. This problem is solved by locating components, one in each subgraph, with the closest contact relation and then connecting these nodes.

Object Classification. To compute a measure of similarity between a target object O and a category instance I , the graph structure similarity of the respective graphs, G_O and G_I , is evaluated. This is based on locating the Maximum Common Subgraph (MCS) between them (Fig. 6). [3] defines MCS as “the largest set of all linked nodes that the two have in common”. It is important to notice here that MCS is not always unique. Once $MCS(G_O, G_I)$ is obtained, the following similarity measure is computed:

$$S_{MCS}(G_O, G_I) = \frac{g(MCS(G_O, G_I))}{g(G_O) + g(G_I) - g(MCS(G_O, G_I))} \quad (3)$$

where $g(G)$ corresponds to the sum of the total number of nodes and total number of edges of a graph G .

If $MCS(G_O, G_I)$ is not null, it implies the existence of a common structure of nodes and edges between the graphs G_O and G_I . Having a common structure makes it possible to build a mapping between nodes and edges of these graphs. This mapping is found by performing breadth first search [4]. Considering Fig. 6 again, we can see that the mapping between the nodes and edges of graphs G_O and G_I can be:

- Mapping between nodes: $a \leftrightarrow 1$; $b \leftrightarrow 2$; $c \leftrightarrow 3$
- Mapping between edges: $ab \leftrightarrow 12$; $bc \leftrightarrow 23$; $ca \leftrightarrow 31$

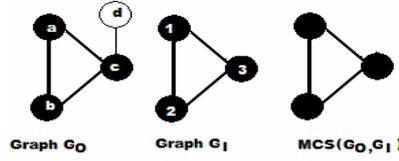


Fig. 6 – An example of $MCS(G_O, G_I)$

where “ \leftrightarrow ” indicates the mapping. In a similar manner the angles can also be mapped. Once the $MCS(G_O, G_I)$ and the mappings for the two graphs G_O and G_I are known, their dissimilarity is measured using the following auxiliary function:

$$d_X(G_O, G_I) = \sqrt{\frac{\sum_{i=1}^n \left(\frac{\sum_{j=1}^k (O_X(i, j) - I_X(\text{map}(i), j))^2}{k} \times M_X(i) + (1 - M_X(i)) \right)}{n}}$$

where

- X represents each of the three items of each graph of components (i.e. the set of components, C , the set of contact relations, R , and the set of ternary relations, A);
- n is the cardinality of X in G_O ;
- k is the number of features of each element of X (according to previous explanations, $k=3$ for the components in C and $k=1$ for the elements in R and A);
- $O_X(i, j)$ is the value of feature j for element i of item X of G_O ;
- $I_X(i, j)$ is the value of feature j for element i of item X of G_I ;
- $\text{map}(i)$ is the element in G_I to which element i of G_O is mapped
- $M_X(i) = 1$ if element i of item X of G_O is mapped to a corresponding element in G_I , otherwise $M_X(i) = 0$

Based on this, the dissimilarity between the two objects is computed as follows:

$$d(G_O, G_I) = w_C d_C(G_O, G_I) + w_R d_R(G_O, G_I) + w_A d_A(G_O, G_I) \quad (4)$$

where w_C , w_R , and w_A are weights, with $w_C + w_R + w_A = 1$ (in the implementation $w_C = 0.55$; $w_R = 0.35$; and $w_A = 0.1$).

Joining the similarity measure in (3) and the dissimilarity measure in (4), the final measure of graph similarity is given as:

$$S(G_O, G_I) = w_S S_{MCS}(G_O, G_I) + w_d (1 - d(G_O, G_I))$$

where w_S and w_d are weights (0.3 and 0.7, respectively, in the implementation).

Given an object to be classified, its graph of components is compared with the graphs of all the stored instances. The category of the instance most similar to the object to be classified (i.e. with the highest $S(G_O, G_I)$) is identified as the category of that object.

5 Experimental Evaluation

The objective of the experiments was to evaluate the performance of the proposed component-based learning approach in a supervised, incremental, open-ended and online learning system. The system was evaluated using the *teaching protocol* proposed in [12]. In each experiment, categories are incrementally taught until the learning capacity of the agent reaches a breakpoint. Classification precision is used as the primary performance evaluation measure. Its evolution from start to breakpoint is analyzed in each experiment.

It should be noted that, because of the exhaustive nature of this protocol, a single experiment can take weeks to finish. Authors have previously conducted a long duration experiment [14] which took more than a week's time. So as to help facilitate later experiments, the images from this experiment were stored along with their respective category names as provided by the human user. Overall, a database of almost 7000 images (pertaining to 68 categories) was created [14].

The most time consuming task, with respect to experiments, was having a human user to show a new object instance every few seconds. For that purpose, a simulated user was implemented. The actions of the simulated user are limited to the teaching actions of the human user: teaching, asking and correction. When the simulated user runs out of images in a particular category, the real user is called to show additional instances. This way, semi-automated experiments can be carried out. In total, a series of 10 semi-automated experiments were conducted (see Table I).

TABLE I
SUMMARY OF ALL EXPERIMENTS

Exp #	# categories at breakpoint	# Question /correction iterations	Classific. precision at breakpoint (%)	# instances per category (\pm std)
1	9	345	48	17 \pm 7.42
2	9	384	56	18 \pm 11.26
3	7	289	62	22 \pm 10.58
4	11	559	66	23 \pm 13.21
5	8	281	54	18 \pm 7.34
6	5	174	60	21 \pm 6.88
7	9	460	33	28 \pm 11.63
8	7	206	52	13 \pm 4.81
9	11	656	45	29 \pm 13.74
10	6	142	61	12 \pm 4.05

Figure 7 shows the evolution of classification precision in the first experiment. In general, the introduction of a new category to the agent leads to the deterioration in classification precision followed by gradual recovery. This process continues until the system starts to confuse the category descriptions to an extent that it can no longer recover. All the experiments followed similar evolution pattern. This pattern has been previously observed in [12] and [14].

In the first experiment, the agent could learn 8 categories, reaching the breakpoint after the introduction of the 9th category. For each of the introduced categories, the

performance evolved in the standard way: initial instability followed by recovery. After introducing the last category, classification precision mostly fluctuates around 55% without showing any sign of recovery. This trend is noticed for around 150 question/correction iterations leading the simulated user to conclude that the breakpoint was reached.

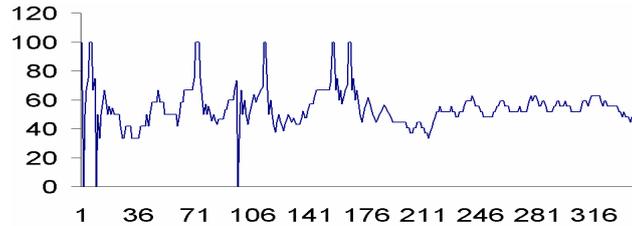


Fig. 7 – Evolution of classification precision versus number of question/correction interactions in a single experiment (breakpoint at the 9th category).

6 Conclusions and Future Work

This paper presented a physically embodied robot with the abilities to learn throughout its life, respond intelligently to external stimuli and communicate with its users. The robot has both cognitive and functional abilities. It interacts with human users in a shared environment, where the human user acts as a teacher and helps the robot in acquiring the vocabulary about the objects present in their shared surroundings.

The robot's learning system incrementally grounds the names of new objects introduced by a human user. This system was built on top of a novel component based category representation strategy. The components of the object are extracted using only color information. An original graph based methodology was proposed to represent the spatial arrangement of the components inside an object. A new classification strategy based on a graph similarity measure was also implemented.

This limitation on the number of words learned implies that the discrimination capability of the presented strategy in its current state is very limited, and there is definitely room for improvement. We believe, however, that the approach is promising. It is worth mentioning that the results reported here are comparable to other works with respect to the number of learned words (12 words [12]; 5 words [9]; 12 words [19] etc.). However, certain works have reported learning considerably more words (e.g. 68 words [14]).

Future work will entail refining the quality of the component extraction approach, which depends highly on the image quality. An efficient handling of this noise will be crucial in taking this method further. Improvement of similarity measures, in a way that is more robust to variations in detected components, is also expected to improve the overall performance.

Acknowledgments. The Portuguese Research Foundation (FCT) supported this work under contract POSI/SRI/48794/2002 (project “LANGG: Language Grounding for

Human-Robot Communication”), which is partially funded by FEDER, and a PhD scholarship to the first author.

References

1. Brady, M.: Artificial Intelligence and Robotics. *Artificial Intelligence*, vol. 26-1, pp. 79–121, Apr. 1985.
2. Burgard, W., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S., Cremers, A.B.: Real Robots for the Real World -- The RHINO Museum Tour-Guide Project. In *Proc. of the AAAI 1998 Spring Symposium on Integrating Robotics Research, Taking the Next Leap*, Stanford, CA, 1998.
3. Cunningham, C., Weber, R., Proctor, J.M., Fowler, C., Murphy, M.: Investigating Graphs in Textual Case-Based Reasoning. *ECBR2004*: pp. 573-586, 2004.
4. Diestel, R.: Graph Theory. Springer-Verlag: 2000.
5. Harnad, S.: The symbol grounding problem. *Physica D*, vol. 42, pp. 335-346, 1990.
6. Kennedy, W.G., Trafton, J.G.: Long-Term Symbolic Learning. *Cognitive Systems Research*, vol. 8, pp. 237-247, 2007.
7. Kirby, S., Hurford, J.: The Emergence of Linguistic Structure: An overview of the Iterated Learning Model. In: A. Cangelosi and D. Parisi (eds.), *Simulating the Evolution of Language*, Springer Verlag: pp. 121-148, 2002.
8. Kozima, H., Nakagawa, C.: Social robots for children: practice in communication-care. 9th IEEE International Workshop on Advanced Motion Control, 2006.
9. Levinson, S.E., Squire, K., Lin, R.S., McClain, M.: Automatic language acquisition by an autonomous robot. In *Proceedings of the AAAI Spring Symposium on Developmental Robotics*, pp. 21-23, March, 2005.
10. Love, N.: Cognition and the language myth. *Language Sciences*, vol. 26, 525-544, 2004.
11. Seabra Lopes, L., Connell, J.H.: Semisentient robots: Routes to integrated intelligence. *IEEE Intelligent Systems*, vol. 16-5, pp. 10-14, 2001.
12. Seabra Lopes, L., Chauhan, A.: How many Words can my Robot learn? An Approach and Experiments with One-Class Learning. *Interaction Studies*, vol. 8-1, pp. 53-81, 2007.
13. Seabra Lopes, L., Chauhan, A., Silva, J.: Towards long-term visual learning of object categories in human-robot interaction. In: J.C. Maia Neves, M.F. Santos, J.M. Machado (eds.), *New Trends in Artificial Intelligence*, APPIA, pp. 623-634, 2007.
14. Seabra Lopes, L., Chauhan, A.: Open-ended category learning for language acquisition. *Connection Science*, vol. 8-4, 2008.
15. Steels, L.: Language games for autonomous robots. *IEEE Intelligent Systems*, vol. 16-5, pp. 16-22, 2001.
16. Steels, L., Kaplan, F.: AIBO’s first words: The social learning of language and meaning. *Evolution of Communication*, vol. 4-1, pp. 3-32, 2002.
17. Steels, L.: Evolving Grounded Communication for Robots. *Trends in Cognitive Science*, vol. 7-7), pp. 308-312, 2003.
18. Thomaz, A.L., Breazeal, C.: Robot Learning via Socially Guided Exploration. In *Proc. of ICDL 2006*, Imperial College, London, 2007.
19. Yu, C.: The emergence of links between lexical acquisition and object categorization: A computational study. *Connection Science*, vol. 17-3, pp. 381-397, 2005.