Towards Robot Imagination Through Object Feature Inference

Juan G. Victores, Santiago Morante, Alberto Jardón, and Carlos Balaguer

Abstract— **This paper presents a robot imagination system that generates models of objects prior to their perception. This is achieved through a feature inference algorithm that enables computing the fusion of keywords which have never been presented to the robot together previously. In this sense, robot imagination is defined as the robot's capability of generating feature parameter values of unknown objects by generalizing characteristics from previously presented objects. The system is first trained with visual information paired with semantic object descriptions from which keywords are extracted. Each keyword creates an instance of the learnt object in an n-dimensional feature space. The core concept behind the robot imagination system presented in this paper is the use of statistically fit hyperplanes in the feature space to represent and simultaneously extend the meaning of grounded words. The inference algorithm allows to determine complete solutions in the feature space. Finally, evolutionary algorithms are used to return these numeric values to the real world, completing an inverse semantic process.**

I. Language Grounding and Mental Models

The notion of linking words with physical objects, actions and abstract concepts is commonly referred to as symbol grounding, language grounding, semantic grounding, or bridging the semantic gap. Having being first defined in 1990 by Harnad [1], decades of research have provided different views ranging from psychology [2] to information retrieval for the semantic web [3] and, more recently, cognitive systems in robotics [4]. A survey of artificial cognitive systems [5] describes a purely symbolic approach to cognition (namely *cognitivist*), as opposed to a self-organized embodied *emergent* paradigm. Purely semantic approaches such as Latent Semantic Analysis (LSA) [6] and Hyperspace Analogue to Language (HAL) [7] have been criticized due to the ungrounded nature of the symbols they manipulate [8]. In contrast, the system presented in this paper manipulates values of features extracted from objects that are present in the environment, providing it with an *embodied* and *grounded* nature. Barsalou predicted that cognitive science will increasingly witness the integration of its different paradigms, with competition between them decreasing [2]. In this sense, the authors expect to contribute to this integration by extending the set of tools provided by symbolic architectures that manipulate embodied grounded information.

One of the first works in linking grounded information to language was VIsual TRAnslator (VITRA) [9]. Dynamical situations are provided via video to VITRA, which in turn analyses and performs automatic generation of natural language descriptions for the movements it recognizes. Another approach [10] uses simple user-robot interaction and language games to conceptualize an object, though no further language grounding or inference possibilities are studied. These and similar contemporary systems can be found throughout literature [11]. A more recent work in the field of inference has provided tools to discover unknown properties of an object from limited views of it [12]. Although it is highly focused on multi-modal categorization, it provides direct cross-feature mappings that allow it to infer certain features, such as missing auditory information only from visual information. However, the completion of information is not the main intention of the developed robot imagination system. Our ultimate goal –and to some extent, achievement– is to semantically tie descriptive words with physical features to then extend the reach of these words, thus enabling the creation of completely new object models from new word combinations. It is accurate to expect that these words are actually descriptions that an end-user provides to the robot for some reason. The use case we study in this paper is having the robot draw an object it has never seen before.

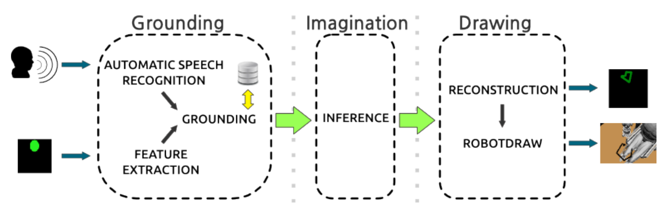

Fig. 1. The robot imagination system architecture in the drawing context.

The robot imagination system in this specific context can be decomposed into three different steps (depicted in Fig. 1), which may occur concurrently.

1) **Grounding:** Populating a knowledge representation database which can dynamically grow not only in the number of elements it contains, but also in number of features it stores per each element.
2) **Imagination:** Generalizing the meaning of words and generating a solution in the feature space, which is performed through an inference algorithm.
3) **Drawing:** Reconstructing a 'mental model' of the object (which may be presented on a screen) from the numerical values of the parameters in the feature space, and using the robot arm or available hardware to draw the object on a given surface.

The use of mental models is an important aspect of the robot imagination system. They play a fundamental role, as reconstructing a mental model of an object is a key

All of the authors are members of the Robotics Lab research group within the Department of Systems Engineering and Automation, Universidad Carlos III de Madrid (UC3M). jcgvicto@ing.uc3m.es

step towards any of the system's end application possibilities. Mental models are treated in several works involving conversational robots. The focus in [13] is the foundation of *object permanence* (awareness of an object when it is not visible) through a simulator, by instantiating visually-detected objects as their virtual equivalents. This allows the semantic description of objects from different perspectives ("touch the block on my left") and from different times ("touch the block you were just holding"). However, feature inference is not the aim of the paper, so it is not treated. In [14], the same system is equipped with force inference from vision. This is achieved by using the virtual instantiation of the real objects, and a dynamics engine to calculate the force necessary to move an object a determined velocity, measured with vision. Previous knowledge of objects for new mental model creation is not exploited in either of the works.

Continuing with this research line, and probably the most similar work to our contribution, [15] adds learning techniques to assign ranges of values of features to words. To achieve this task, they train the system by manually transcribing human-made descriptions of computer-generated coloured rectangles. Then, they consider every word as a potential label and filter them to use only relevant ones. The system assigns a subset of features to each word. Then, an algorithm compares feature distributions between descriptions formed with these words. Finally the system finds the subset of features for which the distributions are maximally divergent when the word is present and when it is not, and assigns these features to the word. The result achieved by this system allows to generate correct semantic descriptions of objects selected by a user on a screen, including spatial relations to other objects. Some apparent limitations of the system are the shape of polygons (only rectangles), and the fixed number of figures in the image (ten).

Our contribution intends to be unique and distinct from this work and others in the following aspects.

1) We perform the inverse semantic process. Instead of asking for a description of a selected object, we build feature models from semantic descriptions.
2) Our system is capable of managing as many features as the programmer is able to represent, and is not limited to fixed or pre-programmed shapes. Features can be extracted from every measurable object.
3) Through the use of hyperplanes, we try to catch all the linear dependencies and couplings that can occur between features without systematically discarding any feature for any given word.

This paper is organized as follows. Section II describes the mathematical formalization of the robot imagination system. Section III presents the programming implementation and experiments. Known and expected system limitations are presented in Section IV. Finally, Section V gives several conclusions.

II. Mathematical Formalization

The robot imagination system is capable of generalizing properties from presented objects, generating a model that is complete in the feature space when asked for it using representative words, and reconstructing a mental model that can be transfered to the real world. In the use case scenario (having the robot draw an object it has never seen before), this process involves three steps, that may occur concurrently: to split the words and save the data (grounding), to generalize the information and return a valid solution in the feature space (imagination), and finally, to interact with the external world to express this mental model formed (drawing). We now proceed to formally define each of these steps.

A. Grounding

In this step, features extracted from real sensor data and semantic words are loaded into a grounding database.

Let n be the number of extracted features. Let $F \epsilon \Re^n$ be the n-dimensional feature space. The grounding database is a set of f labeled points in F, which is initially empty. The grounding database allows the creation of labeled points given $< \boldsymbol{f_w}, w >$ incoming pairs, where $\boldsymbol{f_w}$ is a vector of scalars (the extracted feature values) whose elements become the coordinates of the newly created point in F, and w is a word that becomes the associated label. Fig. 2 depicts an example with two incoming pairs (where w are "green" and "square"), and n=2 ("rectangularity" and "hue"). Note that $\boldsymbol{f_w}$ is the same for both pairs, as the sensor input is identical.

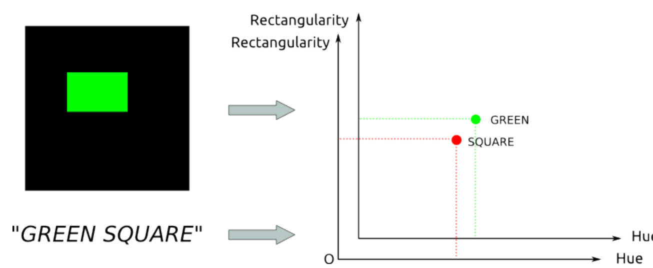

Fig. 2. Populating the grounding database with labeled points.

Let S be the semantic space, composed by m words which are actually the maximum set of *non-repeated* words that have been registered in the grounding database (thus m will always be less or equal to the number of labeled points). G can denote a grounding function that corresponds to the direct semantic process, which can be defined as:

$$G : S \rightarrow F \tag{1}$$

And the translation from the semantic space S to feature space F can be defined as follows:

$$\forall w \epsilon S, G(w) = \{w_{f_1}, ..., w_{fi}, ..., w_{fn}\} \tag{2}$$

Where w_{fi} is the value of a single feature for any w.

B. Imagination

In the context of the developed robot imagination system, robot imagination is defined as the robot's ability to infer the features of unknown objects by generalizing characteristics from previously presented objects.

The first step of the inference algorithm is the generalization process. For each word w, a hyperplane h_w of order $n-1$ is generated. This is an attempt to capture and represent all the linear dependencies and couplings that can occur between features for the word w, and extend them across the feature space F. A total of m hyperplanes are generated.

A fair amount of machine learning regression algorithms can be used to obtain a hyperplane which is fit to a set of points. We make use of the Principal Component Analysis (PCA) algorithm due to its statistical and non-recursive nature (thus avoiding issues such as dependency on the initial guess and the risk of local minima). The most common use of PCA is to supply a lower-dimensional representation of an n-dimensional multivariate dataset with the minimum loss of information. For this purpose, the dataset is projected on a hyperplane that is defined by the Principal Components (PC, vectors that explain the maximum variances) of the original dataset. We exploit this intermediate step of the computation of the PC and use the $n-1$ components as vectors that define our h_w hyperplanes across the maximum variances (maintaining the least variance, hypotesized as the most relevant for the word). The process is the following:

1) **Mean Subtraction:** The mean of each feature is subtracted from each element.
2) **Covariance Matrix Calculation:** This allows us to know the relation between the features. As previously, F is the features space, and we now denote the data of all the words for a single feature as F_i (Eq. 3).

$$\mathbf{F} = \begin{bmatrix} F_1 \\ \vdots \\ F_n \end{bmatrix} \tag{3}$$

We define the covariance between two features as in Eq. 4.

$$\operatorname{cov}(F_i, F_j) = \mathrm{E}\left[(F_i - \mathrm{E}(F_i))(F_j - \mathrm{E}(F_j))\right] \tag{4}$$

Where $\mathrm{E}(F_i)$ is the expected value of the corresponding feature F_i. Currently, all our feature weights are the same, so the expected value can be represented as a simple average. With these elements, the covariance matrix Σ is expressed as in Eq. 5.

$$\Sigma = \begin{pmatrix} \operatorname{cov}(F_1, F_1) & \cdots & \operatorname{cov}(F_1, F_n) \\ \operatorname{cov}(F_2, F_1) & \cdots & \operatorname{cov}(F_2, F_n) \\ \vdots & \vdots & \vdots \\ \operatorname{cov}(F_n, F_1) & \cdots & \operatorname{cov}(F_n, F_n) \end{pmatrix} \tag{5}$$

3) **Eigenvectors and Eigenvalues:** The eigenvectors of the covariance matrix represent the components of the data, and their eigenvalues associated inform about the quantity of the variance explained by their respective component. Finally, we order the eigenvectors in descending order by means of their eigenvalues.

These sorted eigenvectors are the PCs. We extract the ordered $n-1$ components, from which we generate the corresponding h_w.

The second step in the inference algorithm is determining the geometrical figure that contains all of the valid solutions. Note that in the previous step in the inference we have already extended the reach and therefore, to a certain degree, the "meaning" of each word. Given a query composed by q words (q would be 2 in a query such as *"draw a green box"*), we define the geometrical figure that contains all of the valid solutions M as the one resulting from the intersection of the hyperplanes of the corresponding query words. Formally,

$$M(w_1, ..., w_q) = h_{w_1} \cap ... \cap h_{w_q} \tag{6}$$

The order of the resulting geometrical figure M (a point, a line, a plane, or a hyperplane) can actually be computed as $n-1-(q-1)$ in $F \epsilon \Re^n$, from which we derive two special cases, and then describe the general case.

1) Special Case: **q=n**
In this case, the order of M is null. The intersection of the hyperplanes generated by the query words is a single point in the feature space. In absence of degrees of freedom, the coordinates of the point are returned by the inference algorithm as the final solution in the feature space.
Fig. 3 depicts an example of this special case, where the feature space belongs to $\Re^2$, and the query is formed by exactly two words. Note how the initial generalization of the labeled points to the hyperplane allows the algorithm to extend to unexplored yet valid zones of the feature space.

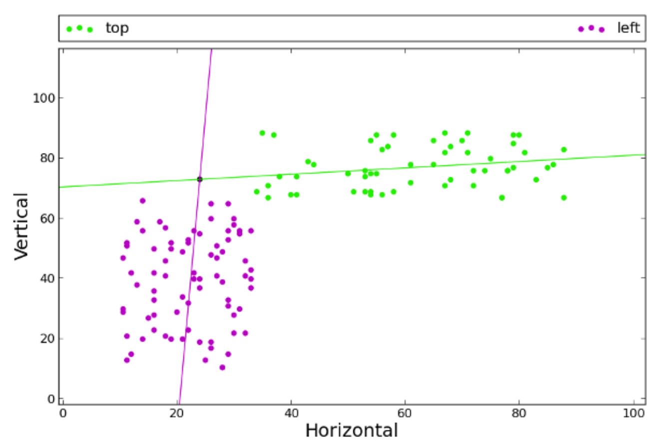

Fig. 3. Imagining features of the semantic query by correlating previously presented inputs. The black circle marks the intersection point.

2) Special Case: **q=1**
In this case, the number of intersections is null, and the order of M is $n-1$. Moreover, the only information is provided by a single word. In absence of possibilities of further correlating with information provided from other sources, the final solution in the feature space returned by the inference algorithm is the center of mass of the pointcloud that corresponds to the word of the query.

For the General Case, we search for a solution which belongs to the geometrical figure that contains all of the solutions we have considered valid, M, while maintaining values that are representative of the query words. With this in mind, we orthogonally project each center of mass of the query words on M, as this projection will provide solutions which are closest to their original points. These projections are accomplished using the modified Gram-Schmidt process for orthogonalization (classical Gram-Schmidt losses orthogonality due to rounding errors when calculated numerically). Gram-Schmidt processes are based on Eq. 7, which expresses the projection of a vector $\mathbf{v}$ onto another vector $\mathbf{u}$:

$$\mathrm{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\mathbf{u}\cdot\mathbf{u}}\mathbf{u} \tag{7}$$

To obtain a vector $\mathbf{v}$ orthogonal to $\mathbf{u}$, the equation would be as in Eq. 8.

$$\mathbf{u} = \mathbf{v} - \mathrm{proj}_{\mathbf{u}}(\mathbf{v}) \tag{8}$$

To obtain a vector orthogonal to more than one vector, the equation can be systematized in a recursive calculation for each vector (Eq. 9).

$$\begin{aligned} \mathbf{u}_k^{(1)} &= \mathbf{v}_k - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_k) \\ &\vdots \\ \mathbf{u}_k^{(k-1)} &= \mathbf{u}_k^{(k-2)} - \mathrm{proj}_{\mathbf{u}_{k-1}}(\mathbf{u}_k^{(k-2)}) \end{aligned} \tag{9}$$

Where k is the number of vectors to be orthogonal with. Fig. 4 depicts an example of this General Case, where the feature space belongs to $\Re^3$, and the query is formed by two words.

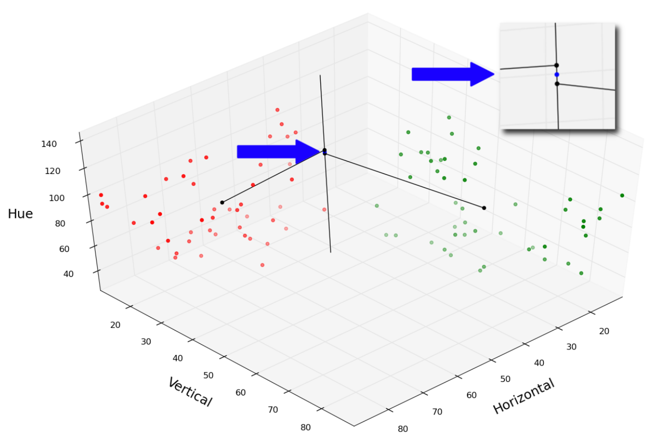

Fig. 4. Example of 3-features space but only 2 words as input. In the augmented region, we can see the final point resulting from averaging the projections.

The number of points resulting from the projections is q, all belonging to M. A weighted mean can be performed upon these points, and the resulting point will still belong to M. By default, we apply the same weight to each projected center of mass, so the inference algorithm will return the coordinates in $F \epsilon R^n$ of the mean as the final solution. Alternatively, a probabilistic distribution function can be computed, or more emphasis can be given to a certain word. The default solution can be considered:

1) **Complete**: Because it is completely defined in all $\Re^n$.
2) **Valid**: Because it is belongs to M.
3) **Balanced**: Because every word weights the same on the final solution.

The mentioned Special Cases are pure simplifications for efficiency of the General Case algorithm. Specifically, for $q = n$ we avoid orthogonally projecting the center of masses of the words on a point, as it will return the same point. Similarly, we save several steps in the case of $q = 1$. In this case, only one hyperplane is generated. The center of mass of the word is contained on M, so its orthogonal projection is again itself. Note that $q > n$ is currently not treated.

C. Drawing

The use case at study is having the robot draw an object it has never seen before. A fundamental aspect of this process is enabling the robot to generate a mental model of the object it has never seen before.

To achieve this, the features values inferred in the previous step are set as inputs to an evolutionary computation algorithm (EC), which performs a steady state selection algorithm [16]. The particularity of this algorithm is that, in each generation, after the selection and crossover process, the resulting individuals replace only a few of them at a time, instead of other algorithms which replace most of them. The following is a summary of the operator details:

1) **Selection:** A tournament is performed between random individuals. Their fitness values are compared and winners are selected for crossover.
2) **Crossover:** Winners are crossed and their children substitute the worst values from the previous tournament.
3) **Mutation:** Finally, each child may be mutated with a certain degree of probability.

The termination condition can be set to a certain number of generations without improvement in the fitness value.

For the process of generating the mental model, the EC controls the coordinated 2D positions of a set of points inside an image ("canvas"). These points are linked to form a shape. Then, we extract features from the generated image and compare them with original data. The points are considered 'fit' according to the fitness function.

Several generated mental models can be seen in Fig. 5. Notice that when query words do not refer to features such as colour and the shape, the values of these features remain uncertain, but tend to acquire tolerated values.

The mental models are generated by sets of coordinated 2D points, which can be used directly for the drawing application as a trajectory. Fig. 6 depicts a screenshot of the ASIBOT robot simulator [17] performing the drawing task.

III. Implementation and Experiments

The whole system consists in a set of YARP [18] interconnected modules. These modules are independent and

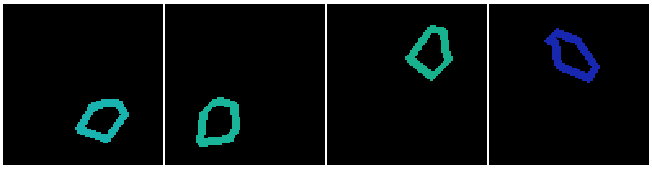

Fig. 5. Mental models generated from different queries: (a) "bottom right", (b) "bottom left", (c) "top right", (d) "top blue".

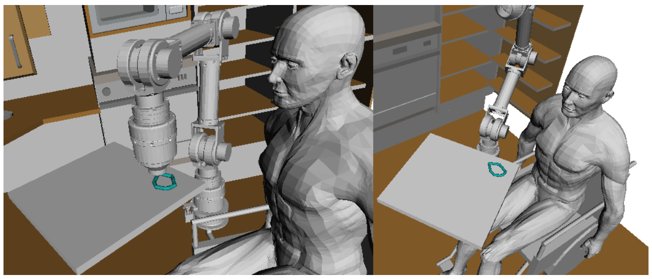

Fig. 6. Simulation of the ASIBOT robot arm drawing the mental model generated from a "bottom left" query.

interchange relevant information through ports. The modules are direct software implementations of the mathematical formalizations explained in the previous section.

To accelerate our experiments, we first train the system with a set of computer generated images and automated descriptions. The images and descriptions are fed to the grounding database. For the final experiment, the system was trained with 300 images with seven-word descriptions, analogous to those of Fig. 7. Each descriptive word could take upon 3 different values, thus the training population represents less than 13.7% of the possible combinations. An additional Gaussian noise (of 1% standard deviation) was added on the features related to the descriptive words.

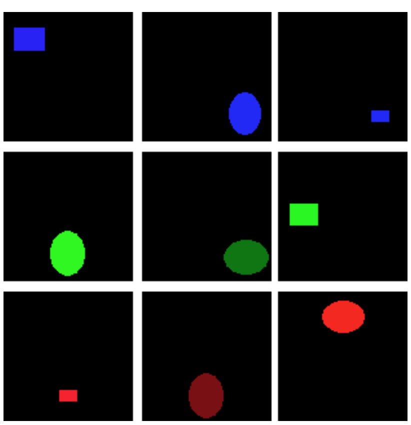

Fig. 7. Example of training images used to populate the space. For instance, the first image is labeled top-left-dark-blue-fat-straight-box.

In our current implementation, 12 features are extracted from the training images. The first two correspond to the vertical and horizontal position of the object in the image, the hue and value are used to define the color, and another 8 features (aspect ratio, total area, rectangularity, first and second axis, solidity, arc, and radius) are used to characterize the contour of the object.

In order to quantify the errors produced by the system (in absence of the errors introduced by the EC) for a given word, we have generated two complementary tables:

1) The errors induced by the scalability of dimensions (Table I): Though the inference algorithm has no innate knowledge of how representative a word is for a given feature or set of features, in the use case, we know that the word 'left' mostly describes the first feature. In this table, we express how the first feature of the algorithm deviates with respect to the uni-dimensional case as more features are incorporated. The same applies for 'top' with respect to the second feature.
2) The errors induced by a specific feature over the rest of the space (Table II): Deviations on features do not depend on dimensionality alone, but also on the noise or ambiguity a given feature can introduce. This table represents how the features related to the word 'left' and 'top' deviate when using only two features, depending on which features are used for the inference.

Regarding the EC, the tournament size for generating the mental model is performed between 3 individuals, the probability of mutation for the children is set to 70%, and the termination condition is set to 50 generations without improvement in the fitness value.

IV. Limitations

We are aware of certain limitations of the presented algorithms. The following is a brief description of these limitations, and proposed potential solutions.

1) **Context dependency:** The current implementation is not capable of correctly managing words that change depending on the context. An example of this could be the word "green", which could refer to a object's color, or to its state of ripeness. This may affect the robot creativity system as it can represent a total loss of linearity of the words with respect to any of the features. The issue could possibly be treated through the use of a previous un-supervised clustering step, for the robot imagination system to then treat words in different regions of the feature space as completely different words. However, no research has been performed in this direction yet.
2) **Periodicity:** The robot imagination system's raw implementation is susceptible to malfunction in the presence of repeating patterns. For example, in the HSV color space, the hue "red" can be represented as $0°$ or as $360°$. A hyperplane fit to these values can tend to settle at $180°$, which is obviously wrong. Again, this issue could possibly be treated through the use of a previous un-supervised clustering step, but research in this direction is yet to be performed.
3) **Specificity:** The presented robot imagination system intrinsically supposes that all words are extendable. In practice, we suppose that this is true, and base this assumption on the fact that even unique identifiers can

TABLE I

ERRORS INDUCED BY ADDING NEW DIMENSIONS FOR THE QUERY "TOP LEFT".

Number of Features	2	3	4	5	10	12
Deviation of 1	0.038	0.036	0.038	0.036	16.855	16.227
Deviation of 2	-0.308	-0.298	-0.298	-0.305	-14.238	-13.541

TABLE II

ERRORS INDUCED BY SPECIFIC DIMENSIONS FOR THE QUERY "TOP LEFT".

Feats.	{1,2}	{1,3}	{1,4}	{1,5}	{1,6}	{1,7}	{1,8}	{1,9}	{1,10}	{1,11}	{1,12}
Dev. 1	0.038	0.009	-0.002	0.012	0.004	-0.006	-0.034	-0.012	0.012	-0.398	-0.045
Dev. 2	-0.308	-77.116	338.902	-78.329	-46.805	-60.785	-78.055	-1.921	-63.425	-203.833	76.898

be used as "leverage" for the generation of new mental models. However, we leave how the use of unique identifiers as query words of the robot imagination system can lead to instabilities of the algorithms or unexpected results as an open question.

V. CONCLUSIONS

In this paper, the authors have presented a system for providing robots with imagination skills through object features and semantic descriptions. The work provides exciting results and sheds some light possible applications, but is still at a very early stage. The developed system possesses certain characteristics that are yet to be proved beneficial or not in spaces of increasingly high dimensionality, such as the assumption that a word should be somehow mapped to every single feature of a given object.

Moreover, the authors consider the research on developing robot imagination systems a natural step in robotic research and cognitive science. The representation selected, where similar objects are close in a continuous space, appears to have a biological base. Recent research in the field of neuroscience has discovered that the objects human perceive are stored in a continuous, and smooth, semantic space in the cortical surface [19].

Future work with the robot imagination system must involve sensor data of various nature. Haptic, rugosity, and depth sensor information should be involved in the inference algorithm. Having the robot draw an object it has never percieved before should be considered an initial, proof-of-concept goal. We envision a future where robot imagination systems are used for advanced tasks, such as recognition of objects that have never been perceived before in domestic environments.

ACKNOWLEDGMENTS

Special thanks to MINECO ministry of Spanish Government for grant DPI2010-21047-C02-01 ARCADIA, and to Comunidad de Madrid and EU structural funds for grant S2009/DPI-1559 ROBOCITY2030 phase II.

REFERENCES

[1] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990.

[2] L.W. Barsalou. Grounded cognition: past, present, and future. *Topics in Cognitive Science*, 2(4):716–724, 2010.

[3] J. Joo. Adoption of semantic web from the perspective of technology innovation: A grounded theory approach. *International Journal of Human-Computer Studies*, 69(3):139–154, 2011.

[4] F. Stramandinoli, D. Marocco, and A. Cangelosi. The grounding of higher order concepts in action and language: A cognitive robotics model. *Neural Networks*, 2012.

[5] D. Vernon, G. Metta, and G. Sandini. A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation*, 11(2):151–180, 2007.

[6] T.K. Landauer and S.T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review; Psychological Review*, 104(2):211, 1997.

[7] C. Burgess, K. Livesay, and K. Lund. Explorations in context space: Words, sentences, discourse. *Discourse Processes*, 25(2-3):211–257, 1998.

[8] A.M. Glenberg and D.A. Robertson. Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning. *Journal of memory and language*, 43(3):379–401, 2000.

[9] G. Herzog and P. Wazinski. Visual translator: Linking perceptions and natural language descriptions. *Artificial Intelligence Review*, 8(2):175–187, 1994.

[10] L. Steels. Language games for autonomous robots. *Intelligent Systems, IEEE*, 16(5):16–22, 2001.

[11] D. Roy. Grounding words in perception and action: computational insights. *Trends in cognitive sciences*, 9(8):389–396, 2005.

[12] T. Nakamura, T. Nagai, and N. Iwahashi. Multimodal object categorization by a robot. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007.*, pages 2415–2420. IEEE, 2007.

[13] K. Hsiao, N. Mavridis, and D. Roy. Coupling perception and simulation: Steps towards conversational robotics. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. IROS 2003.*, volume 1, pages 928–933. IEEE, 2003.

[14] D. Roy, K.Y. Hsiao, and N. Mavridis. Mental imagery for a conversational robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.*, 34(3):1374–1383, 2004.

[15] D. Roy. A trainable visually-grounded spoken language generation system. In *Proceedings of ICSLP*. Citeseer, 2002.

[16] D. Whitley and J. Kauth. *GENITOR: A different genetic algorithm*. Colorado State University, Department of Computer Science, 1988.

[17] A. Jardón, J.G. Victores, S. Martínez, A. Giménez, and C. Balaguer. Personal autonomy rehabilitation in home environments by a portable assistive robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part C Applications and Reviews*, 2011.

[18] P. Fitzpatrick, G. Metta, and L. Natale. Towards long-lived robot genes. *Robotics and Autonomous Systems*, 56(1):29–45, 2008.

[19] A.G. Huth, S. Nishimoto, A.T. Vu, and J.L. Gallant. A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, 76(6):1210–1224, 2012.