Semantic Image Search and Subset Selection for Classifier Training in Object Recognition

Rui Pereira[1], Luís Seabra Lopes[1,2] and Augusto Silva[1,2]

[1] IEETA, Universidade de Aveiro,
[2] Departamento de Electrónica, Telecomunicações e Informática,
Universidade de Aveiro,
{ruipereira,lsl,augusto.silva}@ua.pt

Abstract. Robots need to ground their external vocabulary and internal symbols in observations of the world. In recent works, this problem has been approached through combinations of open-ended category learning and interaction with other agents acting as teachers. In this paper, a complementary path is explored, in which robots also resort to semantic searches in digital collections of text and images, or more generally in the Internet, to ground vocabulary about objects. Drawing on a distinction between broad and narrow (or general and specific) categories, different methods are applied, namely global shape contexts to represent broad categories, and SIFT local features to represent narrow categories. An unsupervised image clustering and ranking method is proposed that, starting from a set of images automatically fetched on the web for a given category name, selects a subset of images suitable for building a model of the category. In the case of broad categories, image segmentation and object extraction enhance the chances of finding suitable training objects. We demonstrate that the proposed approach indeed improves the quality of the training object collections.

1 Introduction

In robotics, as in human cognition, reasoning and communication are activities that involve the manipulation of symbols. Symbols must ultimately be grounded in categories learned through observation, sensorimotor experience and interaction with other agents [1][6][13].

This paper focuses on category learning and symbol grounding through visual perception. This is relevant for grounding the names of (or internal symbols used to refer to) physical objects. In the computer vision literature, this problem has been normally addressed through approaches based on gathering training examples for a pre-defined set of categories, running an algorithm to induce some knowledge about the categories and, finally, testing the induced knowledge on a set of unseen cases [5]. A recent exception is the approach of [18], in which an incremental version of Support Vector Machines is used to acquire visual categories. In the context of human-robot interaction, some recent approaches also explore the combination of incremental learning and interaction with teachers to ground vocabulary about physical objects [11][12][13][14].

In a technological world with endless information resources easily accessible, the dynamic combination of direct perception of the environment and human-robot interaction can be complemented by unsupervised semantic searches e.g. on the Internet. The present paper explores this complementary path. Specifically, the paper focuses on semantic retrieval of images (from the Internet) and unsupervised image subset selection for visual category learning.

This work was carried out in the context of the development of UA@SRVC, a software agent that participated in the Semantic Robot Vision Challenge (2nd edition, Anchorage, Alaska, July 2008, sponsored by NSF and Google). In this challenge, the competing agents are initially given a list of names of categories (e.g. 'coffee cup' or 'Coca Cola can'). Spelling of these names follows some conventions: proper names have their first letter capitalized; common names are given in lower case; titles of books, films, etc., are given between quotes. In a learning phase, agents can search for information about the categories on the Internet. In the performance phase, they will have to search for specific instances of the categories in an environment prepared by the SRVC organizers (or, in the case of software agents, in images of that environment). Other aspects of UA@SRVC are presented in a separate paper[10].

The work presented in this paper is concerned with the Internet search phase. The basic problem addressed here is the following: given the name of a category of objects, gather a set of representative instances of that category and build a model that can be used later to recognize other, previously unseen, instances.

This is a case of unsupervised learning from uncategorized images [5][4] (as opposed to learning from labeled training data[19]): we cluster the images together according to their visual similarities. Several approaches for automated subset selection (or ranking) of uncategorized images exist. In [17], a method to filter Internet image search results based exclusively on visual content is introduced. Inconsistent or strange images are iteratively removed until the k-Nearest-Neighbor strangeness measure drops below a threshold for any of the remaining images. In each iteration, the same measure is used to choose the image that should be removed. A visual model learning method, applied to images obtained via Google, was proposed by [2]. Firstly, the query is translated to multiple languages. Then, using multiple-topic *pLSA* methods, image category models are learned. The same authors [3] had previously developed a method for visual category filtering on Google images, by identifying visual consistencies in the images and using that information to rank them.

The sparse multiple-instance learning (sMIL) approach [16] divides training images into positive and negative bags. Positive bags are those that contain at least one positive instance, while negative bags contain only negative instances. The classifier can discriminate the positive from the negative instances, even if their sparsity in the positive training bags is high. To discriminate between positive and negative bags, an objective function is used to determine a decision boundary, taking into account that the positive bags can be randomly sparse. It's assumed that images retrieved from the Web contain at least one category instance, thus forming a positive bag. In [7], the authors use the first *15* images

returned by a search engine as the seed of a category, to learn a Hierarchical Dirichlet Process category model. This model is then used to classify additional images. When these are positively classified they are also incorporated in the model.

Figure 1 illustrates the flow of information in the category learning module that was developed for the UA@SRVC agent. The agent takes as input a category name (or a list of category names) and starts by searching and retrieving images from the Web, using queries containing that category. In the case of broad categories, these images suffer a process of object extraction, in which images that contain more than one object are segmented into sub-images and noise is discarded as much as possible. Ideally, each sub-image should contain a single object. After this initial pre-processing step, we select a subset of images for training. The goal is to discard as many unrepresentative images as possible, while keeping a good set of category instances. The learning phase is concluded by building models for the categories that are invariant to scale, rotation and, to some extent, deformation.

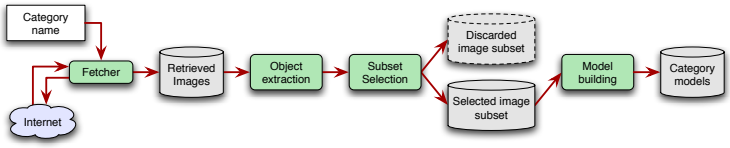


Fig. 1. System overview.

The paper is organized as follows: Section 2 describes image retrieval and pre-processing. Section 3 describes the used representations. Section 4 describes the unsupervised subset selection approach that is proposed. Section 5 describes the experiments that were carried out to evaluate subset selection performance and the results obtained. Finally, section 6 concludes the paper.

2 Image retrieval and pre-processing

Images are initially searched and retrieved using Google. Since many of these images may contain several (possibly irrelevant) objects, these objects are extracted and separately considered.

Image search and retrieval. A *Perl* script, developed based on the *WWW::Google::Images* module, was used to retrieve a set of images from the Internet using the *Google* search engine. Only JPG images with a maximum width of *1200* pixels are retrieved. The maximum number of downloaded images is *20* for specific categories and *40* for general categories. If more images than the maximum are found, only the best images, according to the Google ranking, are used.

General categories are searched with a query beginning in *allinurl:*, meaning all query words must be found in the image URL. Since, for the most part, general categories don't have many words in their name, we maximize the chances of retrieving good images if we force the name to be in the URL. If a search for a specific category doesn't retrieve the maximum number of images and the query contains quotes, then a new search is conducted, removing the quotes from the query. Since we don't use color information, when a search for a general category doesn't return the maximum number of images and the category name has a color in it (e.g. "red apple"), a new search is conducted, removing the color information from the query. When searching for a general category, if the maximum number of images is not reached, a new search is conducted, removing *allinurl:* from the beginning of the query. Note that this can happen after the removal of the color information from the query.

Object Segmentation and Extraction. When processing a retrieved image, the agent must check if the image contains objects other than the intended. An image with several objects, for example, will produce a shape representation that can't be used for anything useful, unless the goal is to detect those objects in the same relative positions in other scenes. These problems are solved by object segmentation and extraction. Due to space limitations, only a very coarse description of the used method will be provided here.

The image is first smoothed by using a Gaussian filter to reduce noise. A Canny edge detector is used to find the edges. The result is a set of white pixels over a black background. Next, the detected edge pixels are followed, trying to isolate individual objects. After detecting a pixel at the boundary of an object, region growing is used to extract its shape. The neighboring pixels of the detected pixel are scrutinized to determine if they should be added to the object being segmented, i.e., if it's also part of the contour. If a bifurcation is reached, its localization is saved and one of the branches is followed. When this path terminates, the process backtracks to the last found bifurcation and takes the unexplored branch.

After the contour is established, its extreme coordinates (bottom, top, left and right) are determined. In a first scan, if a pixel is determined to be inside the contour window and it has white pixels below, above and at both sides, then it's also marked as white. In a second scan, pixels that are inside the same window but are black are also added to the shape if the majority of their *8* neighbors is white. This is useful because a shape is not always defined by connected edges. In fact, there are several possibilities for a contour to be a candidate object or object part, as seen in Figure 2.

If we start in a point and end up in the vicinity of the same point (Figure 2(a)), or if the edge leads to the image boundary (Figure 2(b)) or if it is not continuous but defines an area (Figure 2(c)) then it's possible that this edge defines an object or a component of one.

The next step is the aggregation of edges potentially belonging to the same object. Since an edge doesn't necessarily define an object, edges are grouped

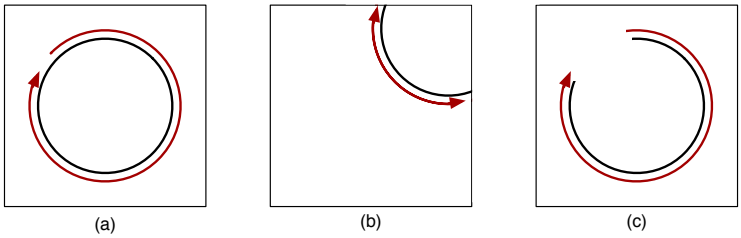


Fig. 2. Extraction cases. These are the basic cases we might encounter while extracting the objects. (a) The object has a defined an continuous contours. (b) The object contours touch the image edges and are continuous. (c) The contours are not continuous.

together according to the average distance of their pixels to their geometric centers, GC, and the distance between geometric centers. If the distance between the GCs of the edges is smaller than the sum of the average distances of their respective pixels to their GCs multiplied by a constant factor k[3], then the edges are aggregated and count as a single object. This is an iterative process, which stops when no more edges can be aggregated.

Finally, two filtering operations are carried out. The first will identify and remove small objects, which are likely to be noise. The other filtering step consists of removing the objects with a number of edge pixels clearly above average, which most likely are cluttered images.

3 Representations and similarity measures

Broad and narrow categories. The UA@SRVC agent divides categories into two main groups: broad (general) categories, whose instances can exist in a wide variety of forms and narrow (specific) categories, whose characteristics are regular and well defined, therefore resulting in high intra-category similarity. Broad categories are identified by common nouns in natural language (e.g.: chair, table, etc.). Narrow categories are identified by proper nouns (frequently brand names) or quoted expressions.

The need for this distinction between broad and narrow categories arises from the fact that some methods are more suited to broad categories while others are more suited to specific categories[10]. For instance, SIFT [9] local features are highly distinctive and are, therefore, very useful for modeling narrow categories, that are rich in descriptive features and with low intra-category variation. However, they fail on more general categories, low in descriptive features and with high intra-category variation. In contrast, shape representations are good for representing the common features of objects in broad categories, but may fail to

[3] In the current version, the multiplicative factor k has the value of 1.0.

capture the distinctive details in narrow categories. In this work, we use a global shape representation for broad categories and a SIFT-based representation for narrow categories.

Global shape context. The used shape representation is a polar histogram of edge pixels[10]. A frame of reference is located at the geometric center of the object. Then, the space around the centre up to the most excentric pixel of the object is divided into a slices (angle bins) and d layers (distance bins) [4]. The intersection of slices and layers results in a polar matrix (Figure 3) that will be mapped to a 2D histogram counting the number of pixels in each cell. This histogram is finally normalized by dividing the counts for each cell by the total count. The histogram is built in $O(n)$ time, where n is the number of edge pixels.

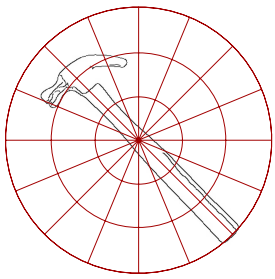


Fig. 3. Shape Context as a global descriptor.

The proposed representation is translation invariant since it's computed in a frame of reference centered in the object. It is also scale invariant because the histogram is normalized by the radius of the minimal circle enclosing the object and centered in its geometric center. The histogram itself is not invariant to rotation. To make a rotation invariant matching possible, we rotate one of the histograms a times while comparing the two shapes. The lowest distance is the one used to calculate the similarity between the two shapes.

The χ^2 distance is used to represent the distance between the histograms. For further details, including comparative performance evaluation, see [10].

SIFT local features. As mentioned above, the UA@SRVC agent represents objects in "narrow" categories through sets of local features extracted by SIFT (Scale Invariant Feature Transform) [9]. SIFT produces highly distinctive image features that can be used for matching objects with different scales, positions and orientations, as well as with some variations in illumination. These features are computed in reference frames aligned with image gradients.

[4] a=40 and d=10 were used.

When matching two objects, the features in each of them are paired according to a nearest-neighbor criterion. Then, instead of using a global threshold to discard matches, the distance to the nearest neighbor and to the second nearest is compared. If the ratio between the former and the latter is greater than *0.35*, the pair of features is rejected. Finally, similarity between two objects is given by the number of accepted pairs of features.

As similarity computation does not depend on object segmentation and extraction, such pre-processing is not used in categories processed by SIFT.

Category representations. Broad (or general) categories are represented by the sets of objects that were selected, where each object is represented by its global shape context. Narrow categories are represented by the concatenation of the SIFT features of each individual object selected for that category.

4 Unsupervised subset selection

Image sets retrieved from the Web for a category will always have an amount of noise. This section presents an unsupervised method for selecting a subset of representative images, discarding those irrelevant or noisy. The approach is based on repeating a basic clustering algorithm a number of times and ranking images based on how often they were included in the largest cluster.

Object clustering algorithm. An unsupervised clustering process is conducted over the images obtained through the Internet search and pre-processing steps. Clustering the images according to their similarity usually results in: a large cluster with most of the good representatives of the target category (i.e. images containing true instances of the category and with little noise); several other (usually smaller) clusters with various outliers. The larger the percentage of good images in the initial set, the higher is the probability that the largest cluster actually contains good representatives of the target category.

Clustering is done using the *k-means*[15] algorithm with Lloyd's iterative refinement heuristic [8] and some additional modifications designed to solve our problem. Lloyd's heuristic starts with seeding a number k of clusters by randomly selecting k images from the initial set of N images, where $k < N$. Then, the remaining $N-k$ images will be added to the closest clusters. Since the object representations described above are not organized in vector spaces (the number of extracted SIFT features varies from object to object and the matching of shape contexts requires rotation), the used clustering algorithm does not compute centroids. Instead, the proximity of an object to a cluster is evaluated by average similarity to the members of the cluster (the similarity measures were identified in the previous section).

After setting up this initial group of clusters, the iterative refinement process starts. In each iteration, for each object, the average similarity to the remaining members of its current cluster and the average similarities to the members of the

remaining clusters are computed. If the minimum value is obtained for a cluster that is different from the current cluster of the object, the object is moved to that cluster. The process terminates when a complete iteration is run without transferring objects between clusters. A post-processing step makes sure there is only one cluster with the largest number of objects. If there is a tie between two or more clusters, the cluster having the closest non-member will receive this extra object, therefore becoming the single largest cluster.

Object ranking and selection. The basic clustering algorithm just described is run for different values of the number of clusters, k. It starts with $N/4$ and incrementally gets to $N/2-1$. Since randomness leads to variations in the clusters produced in each run, the whole process described until now is repeated a certain number of times, K, to provide a reliable sampling[5]. The total number of runs is then $K \times (N/2 - N/4)$.

The number of times, X_i, each object i was included in the main cluster in these runs is updated after every run. At the end, the images are ranked according to X_i. The final selection is determined by going through the images in descending order of X_i and adding them to the selection while the following condition holds:

$$\sum_{s=1}^{S} X_{i(s)} < \eta \times \sum_{i=1}^{N} X_i \tag{1}$$

where S is the number of images included in the selection, s is a rank position, $i(s)$ identifies the image in rank position s, and $\eta \in [0, 1]$ is the reject threshold [6]. When the selection process terminates, the remaining $N - S$ images are assumed as noisy/irrelevant and therefore discarded.

5 Performance evaluation

Images retrieved from the Web are matched among themselves and clustered, using the modified *K-means* algorithm described above, to select a subset that correctly represents the target category. Before this selection, we perform a manual, visual analysis of the retrieved images and count how many of them constitute suitable training images to build a correct model of the category. In the case of shape-based subset selection, images containing several instances of the target category, or with highly stylized objects, or with high background noise around the contours, or not containing any instance of the category are not good training images.

Building a good SIFT model for an object doesn't require such a strict selection as is necessary for a good shape model. First, since SIFT is used for

[5] $K = 100$ was used in the implementation

[6] $\eta = 0.85$ was used in the implementation.

specific categories, it will already start with some advantage. Furthermore, because SIFT is tolerant to occlusion, noise, etc., an image can have a partially hidden, deformed or not alone object and still be useful for building a model.

We compare the percentage of good images in the original set with the percentage of good images in the selected subset to determine if improvement exists.

Shape-based subset selection without object extraction. A total of *31* categories (some of which were used in SRVC'2007 and '2008) were selected to benchmark the subset selection based on shape analysis. For each of these categories, 37 images, on average, were retrieved from the Internet. Only 40% of the retrieved images could be considered good for training.

After selecting the good images in the initial set, we check to see which ones were selected. Figure 4(a) plots the percentage of good images in the selection versus the percentage of good images in the initial set. The categories are identified by numbers in Figure 4 and their names are listed on the right. The linear regression line is given by the function $f(x) = 1.33x - 0.04$. For $x > 0.15$, the regression line has the property of $f(x) > x$, meaning there's improvement. It can be concluded that subset selection improves the quality of the training set.

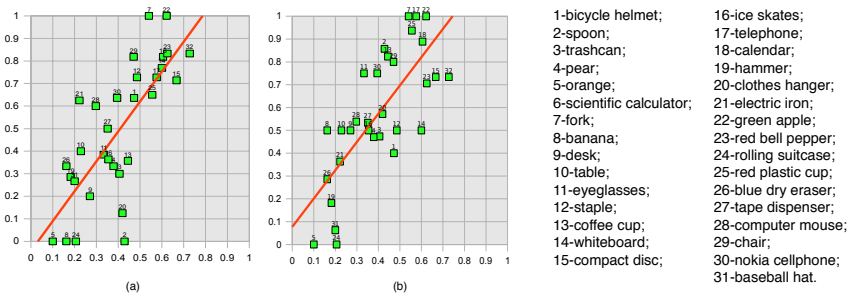


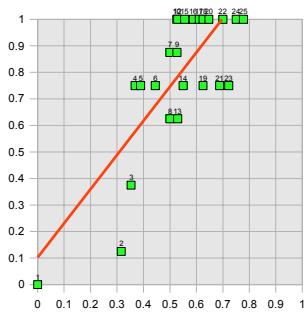
Fig. 4. Percentages of good images after shape-based subset selection (yy axis) versus the respective percentages of good images in the initial set (xx axis): (a) without object extraction. (b) with object extraction. Also shown are the linear regression lines for the plotted points.

Shape-based subset selection with object extraction. In this case, we run object extraction on the initial set, obtaining a larger set of smaller images. Figure 4(b) plots the percentage of good images after the subset selection as a function of the percentage of good images in the initial set. The linear regression line equation is $f(x) = 1.24x + 0.08$. It can be seen that for all values of x, the regression line has the property of $f(x) > x$, i.e. the values of the percentages of good images in the selected subset are superior to the initial percentages. This

line, compared to the one in Figure 4(a), for the same values of x, always has higher f(x) values, which means object extraction improves the final result.

SIFT-based subset selection. We selected *25* categories for this test. The categories are identified by numbers in Figure 5 and their names are listed to the right of this figure. For each category, 19 images were obtained through Internet search, on average. In the initial sets, the percentage of good images was 53% on average.

The performance of subset selection with SIFT for narrow categories is very good. After subset selection, this percentage increases to 79%. Figure 5 plots the percentage of good images in the subset (yy axis) as a function of the percentage of good images in the initial set (xx axis). The figure also shows a linear regression line $f(x) = 1.29x + 0.1$, where from the beginning ($x = 0$) the values of $f(x)$ are always higher than x, meaning there is a significant improvement of the subset with respect to the initial group. The slope is inferior to the ones of Figure 4 (shape-based subset selection), even though we have a significantly better improvement in narrow categories using SIFT-based subset selection. This fact can be explained by the number of categories located in the lower values of x, i.e., categories with poor percentages of good images. In the case of Figure 5, this quantity is smaller, a basis that also accounts for the better performance.



1-Coca-Cola can;
2-Apple iPhone;
3-Butterfinger candy bar;
4-Dvd "300";
5-CD "Final Straw" by Snow Patrol;
6-Cadbury Caramilk;
7-Twinings Earl Grey Tea;
8-Nintendo Wii box;
9-CD "Look-Alikes Jr." by Joan Steiner;
10-Ritter Sport Dark Chocolate;
11-Aerius Allergy;
12-CD "Introducing Joss Stone";
13-DVD "Madagascar";
14-Colgate Total;
15-Dasani water;
16-book "Artificial Intelligence: A Modern Approach";
17-The Economist magazine;
18-cd "Hey Eugene";
19-Kellogg's Corn Flakes;
20-Pringles Can;
21-Fanta Orange Can;
22-Sony AIBO;
23-Ryobi drill;
24-book "Probabilistic Robotics";
25-Crayola Crayons box

Fig. 5. Subset selection with SIFT: percentages of good images after the subset selection (yy axis) versus the respective percentages of good images in the initial set (xx axis). Also shown is the linear regression line for the plotted points.

These images, which are rich in features, provide a solid basis for the clustering algorithm to filter out the noise. Only one category - *Aerius Allergy* (category number 11) - gave worse results in the selected subset than in the initial set of images. It's also worth noting that, apart from category number *9* (*CD "Look Alikes Jr." by Joan Steiner*), which had *0* good images in the initial set, hence being impossible to improve, the category *Aerius Allergy* is the one with the lowest percentage of good images in the initial group. Searching for *Aerius Allergy* returns images with two different brand designs. In other words, this category

is heterogeneous and the two retrieved sub-categories don't share enough SIFT features to build the category model. Due to this heterogeneity and to the low number of good training images, the clustering and subset selection don't perform so well. On the positive end, we can also see that many categories originated a subset where *100%* of the images are good training images.

6 Conclusions and future work

This paper explored semantic web searches and unsupervised subset selection for gathering images that can be used for building models of visual categories. Although there may be other applications, the described functionalities appear relevant in the context of robotics as a complementary means for enabling robots to ground the symbols they use in reasoning and communication. English expressions refering to physical objects are mapped to visual categories. SIFT local features are used for representing narrow categories while shape contexts are used as global descriptors for broad categories. The use of shape contexts as global descriptors for broad categories provides a faster and simpler method than the method in which shape contexts were originally used.

Retrieving images from the Web using only keyword-based searches results in sets of images that cannot be directly used for category representations. In fact, it was observed that between 47% and 60% of the retrieved images are irrelevant or noisy. To solve this problem, a new unsupervised image clustering, ranking and selection method was proposed that, starting from a set of images automatically fetched on the web for a given category name, selects a subset of images suitable for building a model of the category. In the case of broad categories, image segmentation and object extraction enhance the chances of finding suitable training objects. We demonstrate that the proposed approach indeed improves the quality of the training object collections. For initial sets with 50% of good images, the final percentage of good images in the selected subset varies between 63% and 75%.

Improvements on the segmentation algorithm and inquiries on the possibility of a better method to discriminate two different shapes while extracting objects, as well as a better way to remove noise, should be studied in future work. Also, since our work ignores color information, categories such as "*green apple*" or "*red pepper*" do not benefit from this distinguishing feature, a limitation that should be tackled in future iterations. The integration of the system in a hardware platform would be an interesting step in its development.

7 Acknowledgements

The first author is currently with a research grant funded by Aveiro University. The participation of the UA@SRVC team in SRVC'2008 was partially funded by Google. The implementation of SIFT used in our work was developed by Rob Hess and is publicly available. The implementation also used OpenCV extensively.

References

1. Belpaeme, T., and Cowley, S. Extended symbol grounding. *Interaction Studies,8(1)* (2007), 1–6.
2. Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. Learning object categories from google's image search. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 1816–1823.
3. Fergus, R., Perona, P., Zisserman, A., and Science, D. E. A visual category filter for google images. In *In Proc. ECCV* (2004), pp. 242–256.
4. Fritz, M., and Schiele, B. Towards unsupervised discovery of visual categories. In *DAGM06* (2006).
5. Grauman, K., and Darrell, T. Unsupervised learning of categories from sets of partially matching image features. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), 19–25.
6. Harnad, S. The symbol grounding problem. *Physica D, 42* (1990), 335–346.
7. Li, L.-J., Wang, G., and Fei-Fei, L. Optimol: automatic online picture collection via incremental model learning. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (2007), pp. 1–8.
8. Lloyd, S. Least squares quantization in pcm. In *Information Theory, IEEE Transactions on* (1982), vol. 28, pp. 129–137.
9. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60* (2004), 91–110.
10. Pereira, R., and Seabra Lopes, L. Learning visual object categories with global descriptors and local features. *Progress in Artificial Intelligence: 14th Portuguese Conference on Artificial Intelligence - EPIA'2009, LNCS/LNAI, Springer* (2009), In Press.
11. Roy, D., and Pentland, A. Learning words from sights and sounds: a computational model. *Cognitive Science, 26* (2002), 113–146.
12. Seabra Lopes, L., and Chauhan, A. How many words can my robot learn? an approach and experiments with one-class learning. *Interaction Studies, 8(1)* (2007), 53–81.
13. Seabra Lopes, L., and Chauhan, A. Open-ended category learning for language acquisition. *Connection Science, 8(4)* (2008), 277–298.
14. Steels, L., and Kaplan, F. Aibo's first words: the social learning of language and meaning. *Evolution of Communication,4(1)* (2002), 3–32.
15. Steinhaus, H. Sur la division des corp materiels en parties. *Bulletin L'Acadmie Polonaise des Science IV*, C1. III (1956), 801–804.
16. Vijayanarasimhan, S., and Grauman, K. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008).
17. Wnuk, K., and Soatto, S. Filtering internet image search results towards keyword based category recognition. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), 1–8.
18. Yeh, T., and Darrell, T. Dynamic visual category learning. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), 1–8.
19. Zhou, X. S., and Huang, T. S. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems, vol.8, no.6,* (2003), 536–544.