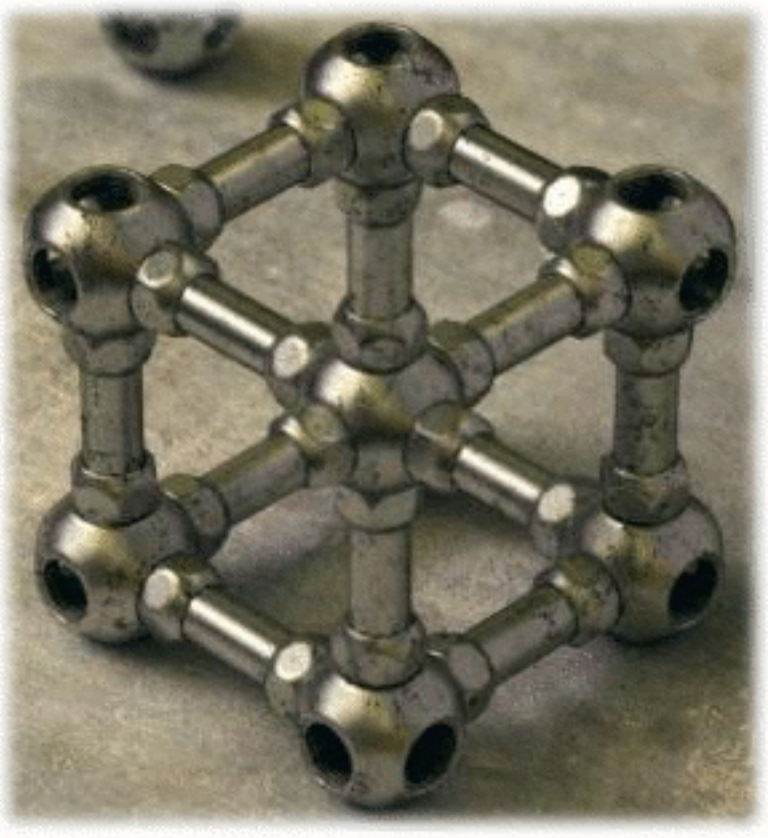


Object Class Detection in Omnidirectional Images [1]

Author(s):
Fábio R. Amaral
Anna H. R. Costa

[1]This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

Object Class Detection in Omnidirectional Images

Fábio R. Amaral and Anna H. R. Costa
Laboratório de Técnicas Inteligentes - LTI
Polytechnic School, University of São Paulo
São Paulo, Brasil
fabioramaral@usp.br, anna.reali@poli.usp.br

Abstract

In this paper we investigate the use of a generative model to the object recognition task in omnidirectional images. The purpose of our work is to find and classify objects typically found in indoor office environments (tables, chairs, etc) through the analysis of images obtained from an omnidirectional vision system. First, a set of generic features are obtained from the query image and clustered in appearance clusters. In training mode we make use of labeled features to compute the joint probability for the containing classes by matching those features to the appearance clusters. The recognition proceeds by matching the features extracted from the query image to the model, computing the likelihood used in the decision equation, through the Bayes rule. Results are presented for some first experiments.

1. Introduction

Object recognition is one of the most actively researched areas of computer vision. This ability can enlarge the possibilities and join useful tasks to the robots, in especial to mobile robots [13]. Objects can be used to represent targets or places and can help machines to be able to accomplish human tasks. In this research line, more generic approaches that deal with multiple classes of objects instead of specific ones are gaining more attention inspired for the task of humans' detection [11, 2, 7].

Multi-class object detection task has been receiving vary approaches. Some of them work sliding a window across the image, and applying binary classifiers to each window. Those classifiers, which discriminates between the class or the background, are trained using standard machine learning techniques, such as boosting [14] or support vector machines [12]. However, such classifiers are trained and run independently, and are unlikely to scale up to the detection of real world amount of object classes. Modern approaches are based on local features which can be shared between various classes of objects. They make use of collections of features or parts, where each part has a distinctive appearance and spatial position.

Local features are typically extracted from images and subsequently grouped into clusters to reduce the size of the feature space. This feature space can reach thousands in a single image. Another benefit acquired with clustering features is the generalization where multiple classes share the same feature information. Previous related works based on local features intended to detect and recognize specific objects previously learned. So they needed to hold all the features informations of those objects as a codebook. The same way, with those approaches, the computational cost in time and space might be unfeasible for real world applications.

In object detection, frequently used clustering methods are k-means, because of its computational simplicity, and hierarchical agglomerative clustering which can be used directly to obtain a data structure for efficient volume search. The combination of both methods as proposed in [6] produces an approximate clustering solution similar to the exact solution but in a faster way. To summarize the approach we first apply k-means to partition the feature space. Then agglomerative clustering is applied within each partition. Finally, the agglomerative method is applied once more on all the cluster centers computed in the previous step. The tree produced in this stage, still holds in its leaves all the features of the training set. An approach proposed in [10] executes a pruning in a defined level of the tree to reduce the search effort. The product of the clustering stage is used as our object representation. This structure contains the appearance clusters in its leaves and for each cluster, an object class likelihood distribution is computed. That process of computing the likelihood distribution is defined as the learning stage.

Another point of our work is the use of omnidirectional vision systems. Because of the benefits acquired when having a vision system that can observe a large part of the scenery, omnidirectional cameras are being used more and

more in different applications, mainly in mobile robotics. A drawback in that vision system is the object deformation in the produced image. That deformation varies in function of the position of the object in the image and difficult the use of geometric models as base for the recognition. Figure 1 is an exemplar of the dataset.

Figure 1. Image obtained from an omnidirectional vision system

The combination of such technologies – local features, generative model and clustering methods – have already being exploited for the object recognition task, but not to omnidirectional vision systems, up to now. So that is the contribution of this work.

This paper is organized as follows. Section 2 describes the generative model used in the approach, in Section 3 the implementation is described including feature detection, clustering method, learning of the model and recognition. In Section 4 we present the results for the initial experiments. Finally, Section 5 presents the conclusions of the paper and points to some future works that are worth pursuing.

2. The generative model

In the literature, the approaches for the object recognition task have been categorized based on the model used in the process. They can be categorized as discriminative or generative.

For comparison, consider the scenario in which an input image, represented in a vector X of pixels or characteristics, needs to be classified as one of the K classes $k = 1, ..., K$. The best solution for this classification is given by the class k which maximize the posterior probability $p(k|X)$.

In discriminative classifiers, the decision boundaries between classes are modeled by computing the posterior probability $p(k|X)$ directly or learning the direct map from input X to the class labels. This classifiers are typically fast during execution and the correct prediction rate is high for well trained object classes.

On the other hand, generative classifiers learn a model of the joint probability $p(k, X)$, by learning the class prior probabilities $p(k)$ and the class-conditional densities $p(X|k)$ separately. The posterior probability $p(k|X)$ used for the decision is computed using Bayes rule, and then picking the most likely k.

$$p(k|X) = \frac{p(X|k)p(k)}{\sum_j p(X|k_j)p(k_j)} \tag{1}$$

This model grants to the system some characteristics: the ability to handle missing data or partially labeled data; the ability to handle composite objects (e.g. faces with glasses and/or hats); a new class can be added incrementally by learning its class-conditional density independently of all the previous classes. A drawback in this model is that the decision depends on an iterative process, slower then the discriminative process.

In our approach, a generative model based on [4] is used to make the classification. The likelihood $p(k|X)$ and the priors probabilities for the object classes are estimated during the learning stage. To explain the model we detail the recognition step. Given features F detected in a query image, appearance clusters A, we make a decision :

$$\frac{p(O_m|A)}{p(B|A)} = \frac{p(A|O_m)p(O_m)}{p(A|B)p(B)} \tag{2}$$

where $p(O_m)$ and $p(B)$ are priors probabilities of an object O_m and background B. We consider background in the equations as every other object but O_m. These priors probabilities can be estimated from the training set or, considered equal 1 for large sets. The likelihood $p(A|O_m)$ is given by

$$p(A|O_m) = \prod_i \sum_j p(a_j|O_m, f_i)p(f_i|O_m) \tag{3}$$

where $p(f_i|O_m)$ is a feature probability for the given object O_m, and $p(a_j|O_m)$ is the probability of an appearance cluster a_j for the same object O_m. These probabilities are the target in the training stage.

The likelihood value obtained from generative models is more reliable than the posterior obtained from discriminative models, since generative models try to represent the true density of the data. To improve the model, parameters like geometric distribution of the features, color descriptor, context, etc, can be used to compound the joint probability distribution.

3. Implementation

Our implementation follows the approaches proposed in [4, 10, 6]. The implementation has four main steps: feature detection, clustering, learning and recognition. The next Sections detail these steps.

3.1. Feature detection

For feature extraction, as most of the approaches, we rely on Lowe's "Scale Invariant Feature Transform" (SIFT) features [8]. In this approach keypoints are identified by finding peaks of a Difference-of-Gaussian (DoG) function applied to different scales of an image. Keypoints of an image are located in regions and scales where there is a high amount of variation. This means that these locations contain useful information for matching. In addition, since these keypoints are in the peak of DoG, minor changes in their surroundings do not greatly affect their locations. The SIFT descriptor of a feature contents its location, scale and a local key descriptor of the region around it. Performing Principal Components Analysis (PCA) on the SIFT descriptors we reduce their dimensionality from 128 to 40 and gain robustness to image deformations [15]. Besides the gain in efficiency, the dimensionality reduction also improves generalization properties of these features since the similarity between two descriptors is calculated over lower dimensionality vectors.

Different methods to find features in the image can be used. In [10] a Canny edge detection combined with Laplacian based automatic scale selection is used instead DoG. The Kadir and Brady method [5] used in [4, 3] finds circular regions in the image having the highest saliency based on maxima of the entropy scale-space of region histograms. In both cases, the SIFT descriptor is used as the identity to the features.

To show the matching accuracy for SIFT features, a test was made with two images from the same environment in different observation points. The image produced by the matching process is shown in Figure 2. We can observe that most of the features in the top image match to the object related in the bottom image.

Figure 4(b) shows an image with the sift features obtained by this process. The features are represented by vectors where their sizes and directions represents the scales and orientations in the image.

3.2. Clustering

The two main clustering methods used in related approaches are k-means and hierarchical agglomerative. We first apply k-means partitioning to the feature space. For each cluster produced, the agglomerative method is applied

Figure 2. Matching result

computing the distance between the features in the bottom nodes and merging the two closest until the last two nodes are merged. The node created in that merging carry the index of the merged nodes and based on their centers, the new center calculated. The average linkage criteria is applied to obtain the center of the node and the Euclidean metric is used in distance calculations. The agglomerative method is applied once more to merge the top nodes. The k value in the k-means process is defined based on [9] as :

$$k \approx (n/2)^{1/2} \tag{4}$$

where n is the number of features.

The tree now is pruned in a chosen level. The more levels has the tree more specific will be its bottom nodes, now called appearance clusters, because the proximity to the local features. The name appearance clusters is given due to the agglomeration of features whose descriptors are near in Euclidean space, in other words, they have the same visual appearance, even belonging to different objects.

In Figure 3 we show an example of the tree structure. The dashed line shows the pruning level. The selection of the number of levels depends on the feature distribution and

needs to be empirically defined. A level definition analysis is shown in Section 4.

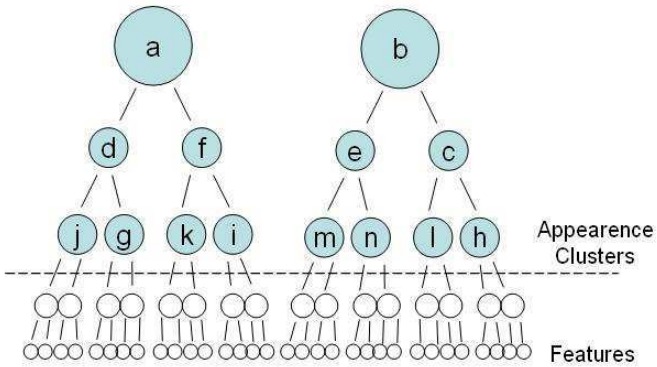


Figure 3. Appearance cluster representation

3.3. Recognition

Given a query image, the features are extracted and clustered as described above and a query tree is produced. The objective is to match the query features in the model to compute the likelihood function as describe in Section 2. The clustering stage is used upon the query features to speedup the process since described in [10], this process is approximately 200 times faster compared to exhaustive search. The likelihood is computed and finding local maxima in its distribution, we find initial hypothesis for objects. As we despise the geometric distribution of the objects while building the tree model, we are not able to return the exact space where the object is in the query image, but the bounding box that hold the features of the hypothesis object.

3.4. Learning

The process of learning an object category is unsupervised. As described in Section 2, the learning objective is to estimate the model $p(a_j|O_m)$ and $p(a_j|B)$. That way we first extract features F from all labeled training examples. We then build the appearance clusters by clustering the feature set with k-means and agglomerative as described above. After pruned, the tree is ready to receive the likelihood information. We match the features back to the appearance clusters centers. Matches are considered only within a threshold β, defined for each appearance cluster during agglomerative process. It represents the distance from the center of the cluster to the most far feature agglomerated in this cluster. Each feature that matches to a_j contributes to the probability estimate. A different feature set can be used in this stage.

The parameter to be calculated in this stage, as defined in Section 2, is the likelihood of the appearance cluster a_j given a object O_m. This probability is given by

$$p(a_j|O_m) = \sum_i p(a_j|O_m, f_i)p(f_i|O_m) \tag{5}$$

We make use of threshold based matching where $p(a_j|f_i)$ is a binary decision where the feature match or not the appearance cluster, and $p(a_j|O_m)$ would be a ratio of the number of features that match to cluster a_j to the total number of matches.

There are two ways to compute the likelihood, the threshold based and a function of similarity. The function of the similarity between the labeled feature and the appearance cluster is used in the approach [10]. For comparison, the similarity function can have the following model:

$$p(a_j|f_i) = \frac{1}{Z} \exp\left(-\frac{\|a_j - f_i\|^2}{\beta}\right) \tag{6}$$

Z is chosen such that $\sum_j p(a_j|O_m) = 1$. The constant β is a threshold when matching the f_i to the cluster a_j.

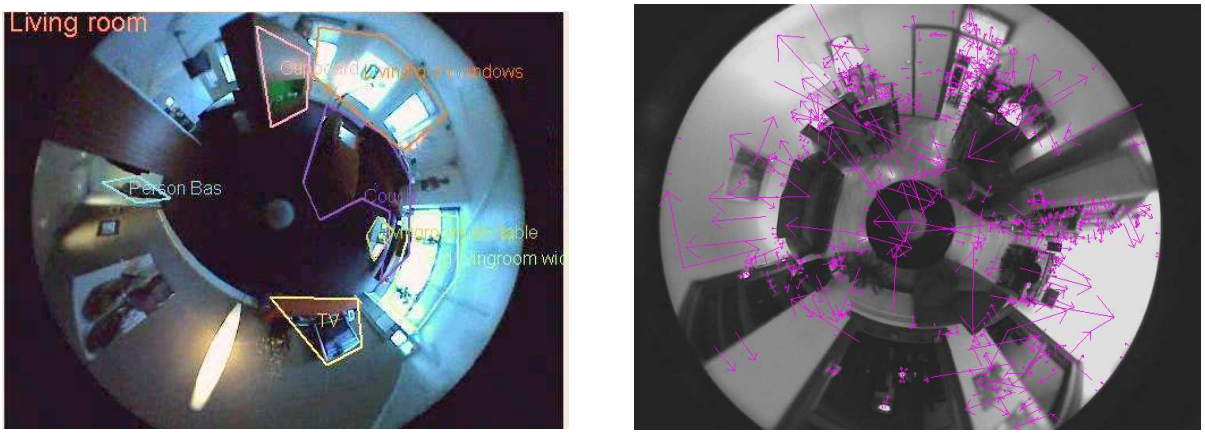
4. Experimental results

The presented approach was tested using a real world annotated data set, obtained from the project "From Sensors to Human Spatial Concept" [1]. The image sequence was captured by an omnidirectional camera using a hyperbolic mirror at 7.5 fps. However, only the odd numbered images were annotated. The complete data set used is composed by 1401 images. For the training set, we use one to each ten images resulting in a set of 140 images. The color information is not used in our approach, only the intensity information is used, in other words, a gray scale representation of the color image. To enhance the image quality a simple histogram equalization was performed.

The target objects in our dataset are armchair, bookcase, cabinet, computer desk, couch, wardrobe and tv. The Figure 1 is an exemplar of the dataset and Figure 4(a) has the annotation example used in training stage. Some of the target objects can be found in these images.

The background – in other words, everything but the target objects – is responsible for about 65% of the features in the feature space. Furthermore, these features belong to a plenty of other objects that fits the appearance clusters and in many cases, their probabilities are higher than the objects that belongs to the cluster. This way, the background recognition reaches 99% of success but disturb the other objects recognition. To overcome this issue, the background features can be partially rejected in learning stage.

Our first observations were made upon the labeled features. We match these features to the model and analyze the



(a) Annotated image

(b) Features detected and represented in the image

Figure 4. Image examples

result individually. Another way is to analyze the clusters of features belonging to the same object in the query image. That way the error rate must be lower.

One study we made cares about the tree level, which might be empirically defined. We notice that low values cause higher error rates analyzing the individual features. Tree levels near the original (tree level before pruning), cause lower error rates for individual features but a reduction in generalization. This happens in our tests because the objects for each class are the same in the whole dataset, for example, the couch in every image is the same. So it is acceptable that when we math the features from couch to the model with the original collected features during training stage, the matching result is higher. On the other hand, when a different couch features are match to the model, the error rate is higher. In this case we need more generic nodes in the bottom of the tree. So the importance of the pruning level. The Figure 5 shows that analysis for the couch object.

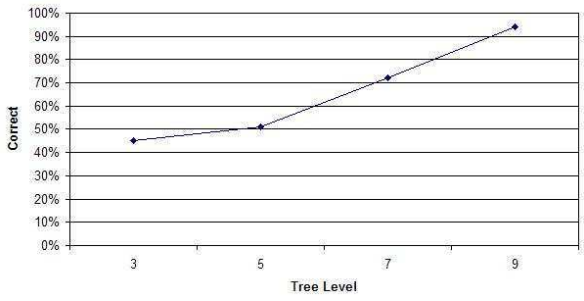


Figure 5. Tree level analysis

The use or not, of background information in the training stage, was also analyzed. With the tree level fixed in 5 and using the background information in the training stage, the error rate for the objects is 62% and for the background features is 1%. Now, not using the background information, the error rate is 49%. When using the background information, the top probable object in the appearance cluster distribution is background. That is the reason for the higher error rate.

Object	Features	Recall
Cabinet	1418	9%
Wardrobe	1022	26%
Couch	537	59%
Bookcase	5360	57%
Armchair	570	24%
Desk	1562	32%
TV	122	29%

Table 1. Objects recall

In table 1 we show the target objects for recognition, their amount of extracted scale invariant features and the recall percentage in our first tests. For this test we set the tree level in 5, which is the mean value in depth of most the trees produced in clustering stage. We notice in this result that objects with more variability of features as a couch or a bookcase have higher recognition values. Objects that can be confused with others have lower recognition values. In recognition process they are also confused with similar objects with higher likelihood. To overcome this issue, other variables representing different characteristics in the objects, like color, geometric details or texture, can be used

in a joint probability distribution. As more details we can extract from the objects and map in the generative model, as accurate can be our results.

In further tests, the lower error rate achieved was 6%. However the analysis made in the level of the features only partially reflects the analysis in object level. An object can be located in an image with 3 strongly matched features as shown in [8], so even with low rates of recognition in the feature level, we can have high rates in object recognition. This object level analysis is being developed for the progress of the research.

5. Conclusions and future work

With the presented approach we show our first attempts to use a set of techniques to make the detection of object classes in images obtained from omnidirectional vision systems. The used techniques are commonly applied to the desired task with successful results. We observe that the generative model has a simple implementation but some particularities need exclusive attention, as the definition of the method for the cluster-feature likelihood calculation. The tree model used to represent the object classes is a powerful tool and also deserve some attention while defining the tree level and other parameters.

An important point for discussion is the use of background features in the model. Our experiments show that their presence during the training stage can cause a confusion in the probabilistic model, but changes in the way we treat them might correct this problem. In our implementation we deal with the background as being an object as the others.

Future works include tests with image segmentation to previously identify possible object regions in the image. Also improvements in the object model as well as in recognition and learning stages shall be done. The goal is to produce a real time system to equip mobile robots for new researches. For that purpose, we are also concerned with the computational costs in time and space. So tests and improvements also need to be done with this goal.

References

[1] O. Booij, Z. Zivkovic, and B. Kröse. From images to rooms. *Journal of Robotics and Autonomous Systems*, 55:411–418, 2007.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:886–893, 2005.

[3] G. Dorkó and C. Schmid. Object class recognition using discriminative local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2004.

[4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:II–264–II–271, 2003.

[5] T. Kadir and M. Brady. Scale, saliency and image descriptions. *International Journal of Computer Vision (IJCV)*, 45(2):83–105, 2001.

[6] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. *British Machine Vision Conference (BMVC)*, 2006.

[7] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:878–885, 2005.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2(60):91–110, 2004.

[9] K. Mardia, J. Kent, J. Bibby, et al. *Multivariate analysis*. Academic press London, 1979.

[10] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:26–36, 2006.

[11] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *European Conference on Computer Vision (ECCV)*, 3021:69–82, 2004.

[12] C. Papageorgiou1 and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision (IJCV)*, 38(1):15–33, 2000.

[13] A. Selvatici, A. H. R. Costa, and F. Dellaert. Object-based visual slam: How object identity informs geometry. *Workshop de Visão Computacional (WVC)*, 1:82–87, 2008.

[14] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision (IJCV)*, 2001.

[15] R. S. Yan Ke. Pca-sift: A more distinctive representation for local image descriptors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:506–513, 2004.