Self-Supervised Clustering for Codebook Construction: An Application to Object Localization

Arturo RIBES [a,1] Senshan JI [b] Arnau RAMISA [a] Ramon LOPEZ DE MANTARAS [a]
[a] *IIIA-CSIC, Campus Universitat Autònoma de Barcelona*
[b] *Departament de Matemàtiques, Universitat Autònoma de Barcelona*

Abstract. Approaches to object localization based on codebooks do not exploit the dependencies between appearance and geometric information present in training data. This work addresses the problem of computing a codebook tailored to the task of localization by applying regularization based on geometric information. We present a novel method, the Regularized Combined Partitional-Agglomerative clustering, which extends the standard CPA method by adding extra knowledge to the clustering process to preserve as much geometric information as needed. Due to the time complexity of the methodology, we also present an implementation on the GPU using nVIDIA CUDA technology, speeding up the process with a factor over 100x.

Keywords. Self-supervised clustering, agglomerative clustering, object localization

Introduction

Visual word representations have become a very popular and successful approach, adopted by many state of the art object recognition methods. This is mainly due to their capacity to enable feature sharing [15] and aggregating statistics in a local region of the feature space to build robust probabilistic models. Object recognition methods based on visual words rely in the construction of a codebook of appearance clusters which quantize some high-dimensional feature space. This codebook is later used to map any visual feature to a finite set of primitives, suitable for machine learning techniques that reduce the impact of the well-known curse of dimensionality.

Recently several works have addressed the introduction of extra knowledge into the codebook representations at different levels. In [2] it is done during the clustering process itself. The Information Bottleneck Method [13] is introduced in a partitional clustering scheme. First, an overdiscretized partition of the feature space is constructed and then, a radius-based clustering method is used to obtain a very discriminative representation with only a few visual words, adapted for bag-of-words representation. Nevertheless, the

[1] Corresponding Author: Arturo Ribes, IIIA-CSIC, Campus Universitat Autònoma de Barcelona, E-08193, Bellaterra, Catalonia, Spain; E-mail: aribes@iiia.csic.es.

This work was supported by the JAE-PREDOC grant from the CSIC and by the grant 2009-SGR-1434 of the Generalitat de Catalunya

contribution of [2] is a reduction of computational cost for a pixel-level labeling method, and not for improving its accuracy. Other approaches construct intermediate features by searching for visual and spatial configurations that occur frequently [10] or that are very discriminative for a given object class, either by its visual or spatial properties [18]. Semantic vocabularies are also constructed using manually annotated ground truth data, where meaningful labels are assigned to regions of the image (e.g. sky, building, etc.) [17]. In [9], specific vocabularies for particular concepts are derived from a universal one.

In [4], the codebook construction is split into three stages: Clustering by visual similarity, co-location, and finally co-activation. A radius-based clustering scheme – the Combined Partitional-Agglomerative (CPA) clustering – is used, changing the measure that estimates cluster similarity. In the first stage, a set of cluster representatives used for matching appearance features are constructed. The other two stages work at a semantic level, modeled by a Bayesian *part-of* network. In the co-location clustering stage, clusters that occur in different images in roughly the same locations are merged, building semantic sub-part clusters. Finally, the co-activation clustering merges sub-part clusters that occur in nearby locations in the same image, building part clusters, which provide the evidence for object presence. Very good results are achieved with this method, cutting down the number of nodes that need to be used for belief computation. For a more extensive comparison between compact codebook construction methods, the interested reader is referred to [16].

In this work, we contribute a novel method, the Regularized CPA clustering, which adds extra knowledge to the clustering process to preserve as much geometric information as needed. This method improves the standard CPA clustering method by introducing a regularization term based in spatial information, which provides the property of self-supervision to the clustering method. The resulting clusters have more discriminative power in estimating object locations in novel images than using the standard method.

The rest of this paper is structured as follows: in Section 1 the base object localization framework used in this work is explained. Section 2 reviews the main codebook construction methods used in the related literature. Next, in Section 3 the proposed method is described and in Section 4 the experimental setup and the obtained results are presented. Finally, in Section 5 we draw conclusions and propose the future work.

1. Generalized Hough Transform Object Localization Approach

Approaches to object localization based on the Generalized Hough Transform (GHT) [5,6,7] work by accumulating votes coming from each local descriptor or visual word present in the image in a pose parameter space, in which later one can search for local maxima to obtain feasible object hypotheses. The main steps of the process are depicted in Figure 1. The initial stages are similar to those of the standard histogram-based bag of features approach [1], as we also use local features and a codebook of visual words. After the appearance codebook is built, we compute the geometric distributions of visual words on objects. Here we use a star-shaped model, as in [5,7], which assumes that feature occurrences are independent given the object centroid. The geometric space is parametrized in three dimensions, two for locations and the scale, representing the offset from feature location to object centroid, thus, a feature-centric view. In our experiments

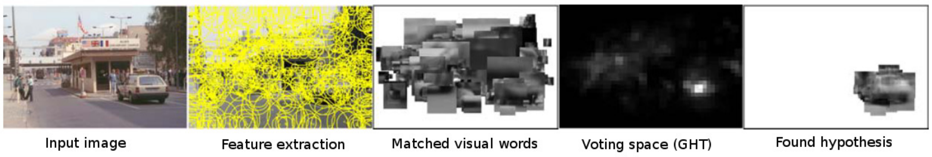


Figure 1. Steps of the standard object localization approach using the GHT.

we used scale- and rotation-invariant features, so offsets are rotated back to the canonical orientation to obtain 3-D normalization. In [5], geometric distributions are modeled as point samples with a constant weight, while in [7] the geometric space is discretized with a grid of the same size as the GHT used in the detection stage, allowing to compute a weight on each cell of the grid as the log-likelihood score of object presence given feature presence. In our experiments we evaluated both approaches.

In order to localize an object in a novel image, local features are extracted and matched to the appearance codebook to obtain the visual words. Then, each of these visual words casts votes in the GHT based on its geometric distribution. After all votes have been cast, the local maxima in the voting space forms the initial set of hypotheses. As typically a high number of false positives is found in this set, a refinement stage is necessary. In [5] good results are obtained with a scale-adapted version of the Mean-Shift algorithm to refine the hypotheses, followed by a final MDL verification stage. In contrast, [7] learns an optimal threshold based on a MAP estimate by modelling positive and negative hypothesis scores as Gaussian distributions. In our experiments, we took this last approach for simplicity, although the reported results for MDL show good improvements. However, this work focuses in the clustering process, which is not related to the choice of the hypotheses validation method. This pipeline has been widely adopted by the research community, with many variations that address particular issues, mainly dealing with multiple views of the objects [12], the huge amount of false positives that arise from this bottom-up approach and enhancing the model with Bayesian inference capabilities [4].

2. Codebook Construction Methods

Our work focuses on optimizing the codebook learning process, tailoring it to the task at hand by introducing information from the geometric distribution of the features. Good clustering methods make the object localization schema more stable and less codebook entries have to be activated during the matching stage, as shown in [3]. Next we will briefly review the two main clustering techniques (partitional and agglomerative) used for codebook learning in recent related literature.

From the two families of clustering schema, the most frequently used methods are k-Means, its hierarchical variant and Gaussian Mixture Models (GMM) as partitional methods and Single- or Average-link Clustering as agglomerative methods, being the last one also known as Unweighted Pair Group Method with Arithmetic Mean (UPGMA). Partitional methods are based on the EM pattern, where each data point is assigned to the closest cluster and the cluster representatives are recomputed, repeating the process until the convergence criteria are satisfied. The drawbacks of partitional methods are the predefined number of clusters, the boundary artifacts, its dependence on a good initialization of cluster representatives (although good initialization methods for k-Means have been proposed) and that the methods are typically biased to put more clusters in very populated regions of the feature space [16].

Agglomerative methods, on the other hand, start with as many clusters as training data points and, at each iteration, the two closest clusters are merged until the convergence criterion is met. The usual stopping criterion is a threshold on the maximum distance between the closest pair of clusters, which determines how compact our clusters will be. Besides, the clustering trace can be saved and revisited to generate clusterings with different thresholds at almost no computational cost. The main bottleneck of agglomerative clustering is the distance matrix computation, which has squared time and space complexity. With some minor modifications, the space complexity can be reduced to linear storing only the closest neighbor for each cluster. In [3] a method to reduce the computational complexity is proposed, which consists in maintaining a table of cluster Reciprocal Nearest Neighbors (RNN) and, after each merge, update only the table for neighbors of the merged clusters. Furthermore, the distances between clusters can be efficiently determined using its mean and variance, which can be incrementally computed. However, this method works only if the clustering criterion fulfills the reducibility property:

$$d(c_i, c_j) \leq inf(d(c_i, c_k), d(c_j, c_k)) \Rightarrow inf(d(c_i, c_k), d(c_j, c_k)) \leq d(c_i \cup c_j, c_k) \tag{1}$$

where c_i,c_j and c_k are clusters and $d(c_j, c_k)$ is a distance measure. This has been proven to be valid for the Average-link criterion using Euclidean distances but, as it will be shown, it is not valid for the regularized inter-cluster distance used in our experiments. With this modifications, agglomerative clustering is suitable for clustering large sets of data. In [3] is shown that the run time of k-Means exceeds that of agglomerative clustering using RNN for more than ten or twenty thousand data points. Furthermore, clusters resulting from agglomerative methods have lower variance, and therefore less ambiguity when matching novel features to codebook entries.

3. Object Localization with a Regularized Codebook

Our proposed object localization system is based on the Generalized Hough Transform based approach for object localization described in section 1, but with some improvements, mainly in codebook creation. The feature detection stage relies in the Harris-Laplace detector, which is known to give state of the art results in object categorization. Extracted patches are described in a rotation- and scale-invariant frame using the well known SIFT descriptor. After all features have been extracted from training images, we construct the codebook using the method proposed in this section. Finally, for matching new SIFT descriptors to codebook entries, hard-assignment is used: We assign the identity of the closest cluster prototype. In [5] authors show that soft-assignment in recognition mode does not provide an accuracy improvement, but it does for learning. We expect that using our regularized clustering scheme, soft-assignment will not be necessary even in the learning stage. Due to space limitations, we describe in detail only the regularized codebook construction method, and readers interested in a more comprehensive explanation of our complete method are referred to [11].

3.1. Combined Partitional-Agglomerative clustering

In [7], the codebook generation is efficiently computed using k-Means clustering of Gabor jets and color histograms. However, as stated before, it has been shown that agglomerative clustering gives better codebooks in terms of cluster compactness. However, agglomerative clustering is very demanding in memory and computation if our data set is bigger than about $20,000$ samples. This is a problem if we want to cluster hundreds of thousands of features, as happens with object category recognition with multiple classes. In order to address this limitation, we have adopted the schema known as Combined Partitional-Agglomerative (CPA) clustering. It consists in applying a partitional method (e.g. k-Means) to the whole data set, obtaining k partitions of no more than $20,000$ samples. Next, agglomerative clustering is applied to each partition independently using a low distance threshold, so we end up having k relatively small sets of clusters. Then, all the k sets are combined and agglomerative clustering is applied again, this time using the distance threshold that we would normally use. The resulting cluster representatives form our codebook.

3.2. Regularized CPA clustering

A careful reader may have noticed that using the clustering schema described above, the stopping criterion used – the threshold on inter-cluster distance – may be too weak for obtaining a good representation suitable for the object localization task. To solve this, we propose to bring extra knowledge into the clustering process to obtain a more discriminative discretization of the feature space. As we saw in the object detection pipeline, appearance patches are extracted from the image, matched to a codebook and then the geometric distributions of activated codebook entries are used to accumulate evidence of object presence in a given location on the image. In the optic of scale-space theory, it is desirable a trade-off between the accuracy of the geometric distributions associated with codebook clusters and the aperture of its receptive field in the appearance space.

The information bottleneck method has been recently used in many successful approaches ranging from feature selection [14] and codebook refinement for bag-of-words representations [2]. This principle maximizes the following Lagrangian:

$$I(C;Y) - \lambda * I(C;X) \tag{2}$$

where X is the random variable we want to code, Y is a relevant variable and C is the compressed variable, the codebook. In our application, we want that the codebook representatives lose as much information as possible from the initial feature set, that is, we want the clustering process to maximize the aperture of the receptive fields in SIFT appearance space, while preserving a fraction – governed by the λ parameter – of the information shared between each codebook entry and its spatial distribution relative to object centroid. In the standard procedure, the best merge is the pair of clusters which have the minimum inter-cluster distance. This is very simple and works well in the initial stages of clustering, where C is still very close to X (Eq. 2) and the optimal merge is obvious. At some point of the iterative merging procedure, selecting the best merge becomes harder, as there are many pairs of clusters with roughly the same inter-cluster distance. If the task is to make good predictions for object locations, it would be a bad choice to merge two similar clusters in appearance space but very different semantically, e.g. a cluster representing eye shapes and another for mouth shapes. With the aim of evidencing this effect, we computed the distance between the spatial center of the hyper-rectangle encompassing all the clusters and the density center. If the distribution is uniform, then the density center should coincide with the spatial center. The measure is defined as:

$$degree_{uni} = ||x_{spatial} - x_{density}||_2$$

where $x_{spatial} = \frac{x_{min}+x_{max}}{2}$ and $x_{density} = \frac{1}{N}\sum x_i$

This contingency calls for regularization, so in our method we added such a term based on the distance between the underlying geometric distributions linked to the pair of clusters to be merged. In our experiments, we used a metric based on a symmetrized estimate of the Kullback-Leibler divergence between the two geometric distributions.

Now we have explained *how* we obtain good merge candidates, it remains to explain *when* we need to compute them. Ideally this should be done from the start, but there are some problems with that: First, KL divergence estimate is based in the k-NN framework, so we need enough samples to get a good probability density estimate; second, at the initial stages of clustering the merges are evident, so only with appearance distance is enough; and last, but most important, the regularized inter-cluster distance does not fulfill the reducibility property that RNN needs, so we need to recompute the distances at each iteration, making the regularized clustering prohibitively slow if applied too early. The solution we propose goes through defining two parameters: The first parameter is the applicability threshold T_{reg}, which dictates the minimum inter-cluster appearance distance necessary to start computing the regularization term, and the second is the minimum number of geometric samples a distribution needs to contain to be considered as object and not background. The optimal choice for the first parameter is determined experimentally, and follows a better explanation of the second. A cluster is labeled as being either object or background depending on the number of geometric samples to compute a spatial distribution that it contains. This is related to cluster precision in object-background terms. Note that by geometric samples we mean features found inside a ground truth bounding box belonging to an object while training, which are the only ones used to estimate object centroids. With this concept in mind, we can distinguish three kinds of merges:

- Object vs. object. This is the only case where we can actually compute the geometric distance between corresponding spatial distributions.
- Background vs. background. In this case, computing the geometric distance is not meaningful, but still we have to define a value for it.
- Object vs. background. This case is the most problematic, as can have two possible interpretations: either it is a real background cluster, so we do not want to merge them, or it is an object cluster that still has not accumulated enough geometric samples. As in the previous case, we need to define a value for the geometric distance.

For second and third kind of merges, the most straightforward solution is assigning values based on the distribution of geometric distances for pairs of object clusters, which can actually be computed. For the background-background case, the average geometric distance is used as we do not want this type of merge to be penalized. The object-background case, however, needs to be penalized as it would otherwise decrease cluster

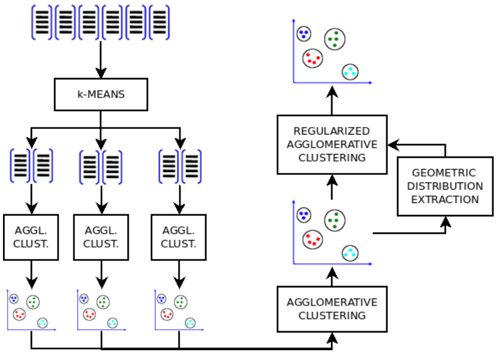

k-MEANS
AGGL. CLUST.
AGGL. CLUST.
AGGL. CLUST.
AGGLOMERATIVE CLUSTERING
REGULARIZED AGGLOMERATIVE CLUSTERING
GEOMETRIC DISTRIBUTION EXTRACTION

Figure 2. Regularized CPA clustering scheme.

precision. In our experiments, we assigned in this case a value equal to the mean geometric distance between any object-object pair of clusters plus three standard deviations. We apply the heuristic that merging an object cluster with a backgorund one is as bad as merging a very different pair of object clusters. Figure 2 illustrates the procedure. We tested two measures of distance between a pair of geometric distributions: The χ^2 distance between the two histograms, and another metric based on the Kullback-Leibler divergence.

Histogram Measuring Geometric space is divided into small bins. Let H_x, H_y be two histograms for geometric distributions associated with cluster X and Y. Chi-square distance between H_x and H_y is used:

$$D_{\chi^2}(H_x, H_y) = \sum_i \frac{(H_x^i - H_y^i)^2}{H_x^i + H_y^i} \tag{3}$$

Kullback-Leibler Divergence Let P, Q be two continuous distributions, the discrete form of Kullback-Leibler divergence from P to Q is defined as:

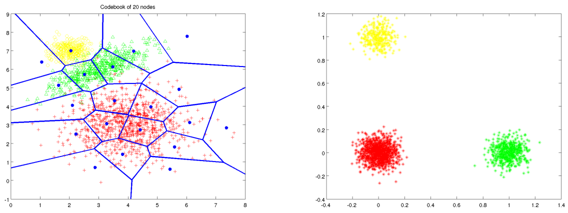
$$D(P|Q) = \sum_i p(x_i) \log \frac{p(x_i)}{q(x_i)} \tag{4}$$

which is computed using the kNN framework, as proposed in [8]. As $D(P|Q)$ is not symmetric, to make KL divergence a metric, we define a new distance which is the symmetrized KL divergence:

$$D_{SKL}(P, Q) = D(P|Q) + D(Q|P)$$

4. Experimental Results

In this section we explain the experiments done in order to demonstrate the feasibility of the proposed method. For this purpose, we designed a toy problem as a proof of concept, consisting of three *semantic classes* (Fig. 3). Although appearance clusters have some overlap and different sizes and extent, geometric clusters provide more information in order to better discriminate between good and bad merges, so we should rely on that to



(a) (b)

Figure 3. Synthetic dataset for the toy problem. The colors represent three different semantic classes. (a) Appearance distribution with Voronoi diagram for $T_{reg} = 20$ clusters. (b) Geometric distribution.

guide the clustering. With no regularization, the clustering makes a very bad merge at 12 clusters, decreasing the cluster precision, so we should have stopped. Using $\lambda = 1$, this point is at 8 clusters. Finally, with $\lambda = 8$, we can reach 6 clusters without losing cluster precision. This experiment shows that our regularization method produces more stable results in terms of preserved spatial information. For more details, the reader is referenced to [11].

For the rest of our experiments, we have used the TUD Motorbikes dataset [5]. The test set contains 115 images collected from the World Wide Web. Each image contains one or more motorbikes at different scales, usually partially occluded and in front of difficult backgrounds. We used the same training set as in [5], which consists of 153 images with uniform background of motorbike side views. The computation of geometric distribution distances is very demanding computationally. Given the advances in parallel computation techniques like CUDA technology and current GPUs, we can run algorithms with speedups superior to 100x. Our implementation took only 5 hours to cluster $260,000$ SIFT features, while a CPU-only implementation would need hundreds of hours, making it unfeasible to use. In order to see how the geometric distributions evolve as the clustering progresses, we computed the average entropy for the clusters each N iterations. First, the entropy rises because clusters are in formation. After a plateau, where we suppose that mostly background clusters are being merged, the entropy starts decreasing very fast, which gives evidence of the structure that is being consolidated, as many similar object clusters are being merged. Finally, we see that the entropy rises until we finish, suggesting that the clustering process should have stopped. After some initial test to determine the interesting parameter ranges, we used $T_{reg} = 10,000$ clusters, the regularization parameter $\lambda \in \{1, 2, 4, 8, 16, 32, 64\}$ and codebook sizes of 500, 1000, 2000, 4000 and 7000 visual words. The GHT bin size was also optimized by cross-validation. We used the overlap ratio between predicted and ground truth bounding boxes, which should be at least 50% to accept an hypothesis. As we did not implement any of the common robust hypotheses validation and refining algorithms common in the literature, we mainly focused on the impact that our method has on improving recall, and analyzed precision separately. Besides, the intention is to show how the codebook size affects false positives and true positives.

Figure 4 shows the recall results for different codebook sizes. As can be seen, using a codebook of 7000 clusters the results are virtually the same, except for high regularization factors. This was expected given the reduced range of appearance distances compared to later stages of clustering, e.g. with 500 or 1000 clusters. More interestingly, when codebooks are smaller, from 4000 to 500 clusters, we can see how our proposed regularization scheme clearly improves the results, with up to 9% higher recall with the 500 clusters codebook. Using high regularization factors help maintaining good recall for 1000-2000 cluster codebooks. Intermediate regularization factors have a more stable behavior and obtain good results with only 500 clusters. On the other hand, the increased aperture in appearance space affects precision results when using the regularized codebook – specially for smaller codebook sizes – as shown in Figure 5. Due to time constraints, resolving this drawback by incorporating an state of the art hypothesis verification stage is left for future work.

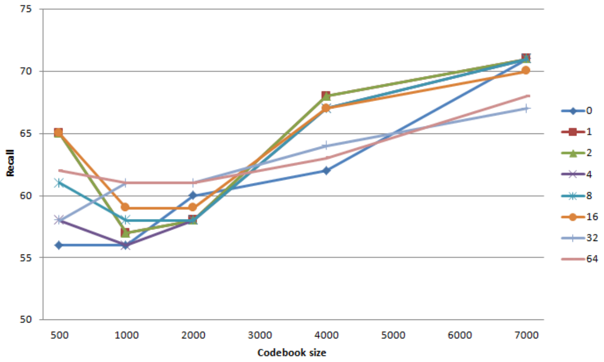


Figure 4. Recall results with different vocabulary sizes. Each line represents a regularization factor.

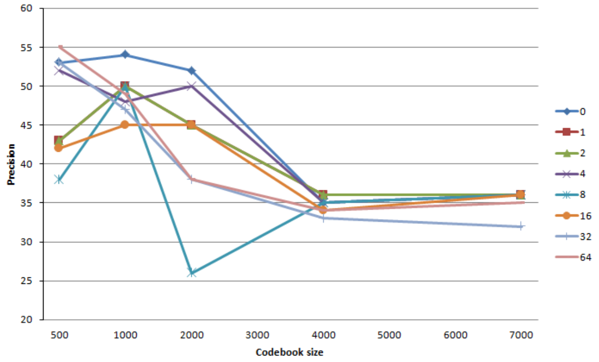


Figure 5. Precision results with different vocabulary sizes. Each line represents a regularization factor.

5. Conclusion and Future Work

In this work, we contribute a novel self-supervised radius-based codebook construction method which uses visual word spatial information to regularize the agglomerative clustering process, which allows to retain more information in the geometric space than the standard agglomerative clustering methods. This geometric information is used to hypothesize where objects are in novel images. We are looking forward to implement the whole object recognition framework in CUDA, as it is highly parallelizable and we expect to obtain serious speedups.

The experimental results with the motorbikes dataset are quite promising, as we obtained a significantly better recall than using a non-regularized codebook. It would be interesting to automatically adjust the regularization factor to the vocabulary size, as small vocabularies typically achieved better performance and stability with higher regularization factors.

Finally, we plan to make more thorough experimentation with intermediate vocabulary sizes. Also, we will make use of a state of the art object localization framework, using our regularized codebooks there to see if those methods also benefit as much as ours.

References

[1] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[2] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *10th European Conf. on Comp. Vis.*, pages 179–192, 2008.

[3] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *Proceedings of British Machine Vision Conference*, 2006.

[4] B. Leibe, A. Ettlin, and B. Schiele. Learning semantic object parts for object categorization. *Image Vision Comput.*, 26(1):15–26, 2008.

[5] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int. J. Comput. Vision*, 77(1-3):259–289, 2008.

[6] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the Int. Conf. on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[7] E. Murphy-Chutorian and J. Triesch. Shared features for scalable appearance-based object recognition. In *WACV-MOTION '05: 7th IEEE Workshops on Application of Comput. Vision - Vol 1*, pages 16–21, Washington, DC, USA, 2005. IEEE Computer Society.

[8] F. Perez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 1666 –1670, july 2008.

[9] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1243–1256, 2008.

[10] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool. Efficient mining of frequent and distinctive feature configurations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th Int, Conf, on*, pages 1–8, 2007.

[11] A. Ribes. Effective object localization with regularized codebook construction. Master's thesis, IIIA-CSIC (Univ. Autonoma de Barcelona), 2010.

[12] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. volume 2, pages 1589 – 1596, 2006.

[13] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[14] K. Torkkola. Feature extraction by non parametric mutual information maximization. *J. Mach. Learn. Res.*, 3:1415–1438, 2003.

[15] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 29:854–869, 2007.

[16] J. C. van Gemert, C. G. M. Snoek, C. J. Veenman, A. W. M. Smeulders, and J-M. Geusebroek. Comparing compact codebooks for visual categorization. *Comp. Vis. Image Underst.*, 114(4):450–462, 2010.

[17] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *Int. J. Comput. Vision*, 72(2):133–157, 2007.

[18] Y-T. Zheng, M. Zhao, S-Y. Neo, T-S. Chua, and Qi Tian. Visual synset: Towards a higher-level visual representation. *IEEE Conf on Computer Vision and Pattern Recognition*, 0:1–8, 2008.