Bag-of-Bags of Words model over irregular graph partitions for image retrieval

Yi Ren, Aurélie Bugeau, Jenny Benois-Pineau

Bag-of-Bags of Words model over irregular graph partitions for image retrieval

Yi Ren, Aurélie Bugeau, Jenny Benois-Pineau
University of Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France
Email: {yi.ren, aurelie.bugeau, jenny.benois-pineau}@labri.fr

***Abstract*—The paper presents a novel approach, named bag-of-bags of words (BBoW), to address the problem of Content-Based Image Retrieval (CBIR) from image databases. The proposed bag-of-bags of words model extends the classical bag-of-words (BoW) model. An image is represented as a graph of local features on a regular grid. Then irregular partitions of images are built using different graph cutting methods. Each graph is then represented by its own signature. Compared to existing methods for image retrieval, such as Spatial Pyramid Matching (SPM), the BBoW model does not assume that similar parts of a scene always appear at the same location in images of the same category. The extension of the proposed model to pyramid gives rise to a method we name irregular pyramid matching. The experiments demonstrate the strength of our method for image retrieval when the partitions are stable across an image category. The experimental results for Caltech101 benchmark show that our method achieves comparative results as SPM, and is globally more stable.**

I. Introduction

Recent methods in Content-Based Image Retrieval (CBIR) mostly rely on the bag-of-visual-words (BoW) model [1]. The idea, borrowed from document processing, is to build a visual codebook from all the feature points in a training image dataset. Each image is then represented by a signature, which is a histogram of quantized visual features-words from the codebook. Image features are thus considered as independent and orderless. The traditional BoW model does not embed spatial layout of local features in image signature. However, those information have shown to be very useful in tasks like image retrieval, image classification, and video indexing. Ren et al. [2] put forward a concept of grouping pixels into "superpixels". Leibe et al. proposed to adopt codebooks to vote for object position [3]. Lazebnik et al. [4] partitioned an image into increasingly fine grids and computed histograms for each grid cell. The resulting spatial pyramid matching method (SPM) clearly improves the BoW representation. Nevertheless, this method relies on the assumption that a similar part of a scene generally appears at the same position across different images, which is not always true.

Graphs are versatile tools to conveniently represent patterns in computer vision applications and have been vastly investigated. By representing images with graphs, measuring the similarities between images becomes equivalent to finding similar patterns inside series of attributed (sub)graphs representing them. Duchenne et al. [5] introduced an approximate algorithm based on graph-matching kernel for category-level image classification. Gibert et al. [6] proposed to apply graph embedding in vector spaces by node attribute statistics for classification. Bunke et al. [7] provided an overview of the structural and statistical pattern recognition, and elaborated some of these attempts, such as graph clustering, graph kernels and embedding etc., towards the unification of these two approaches.

In this paper, we present a new approach for Content-Based Image Retrieval (CBIR) that extends the bag-of-words (BoW) model. We aim at embedding color homogeneity and limited spatial information through irregular partitioning of an image into a set of pre-defined number of graphs. Each partition results from the use of graph partitioning methods, the node of the initial graph being positioned on a dense regular grid of pixels. We compare two graph partitioning methods, namely Graph Cuts [8] and Normalized Cuts [9]. The BoW approach is next applied to each of graph partition independently. An image is then represented by a set of graph signatures (BoWs), leading to our new representation called *bag-of-bags of words* (BBoW). As in the spatial pyramid matching approach [4], we also consider a pyramidal representation of our images. BBoW models are therefore computed at different resolutions, i.e. with different number of graphs. More precisely, a different number of (sub)graphs are considered at each level of the pyramid. The comparison of images in a CBIR paradigm is then achieved via comparison of the irregular pyramidal partitions. We call this pyramidal approach *Irregular Pyramid Matching* (IPM). Hence in this paper, we try to address a challenging question: will an irregular, segmentation-like partition of images outperform a regular partition (SPM)? Intuitively, it is invariant to the rotation and reasonable shift transformations of image plane. Nevertheless, what can be its resistance to the noise and occlusions? How will it compete with SPM when embedded into pyramidal paradigm?

The remainder of the paper is organized as follows. In section II we briefly introduce the notations and prerequisites. The proposed bag-of-bags of words model is discussed in section III and the irregular pyramid matching in section IV. Section V presents the experimental results and we conclude in section VI.

II. Terminology

The input database $\Omega = (I, V)$ is composed of N RGB images $I = \{I_1, \ldots, I_N\}$ and of a visual codebook $V = \{V_1, \ldots, V_B\}$ of length B. The codebook is built using k-means over the SIFT [10] descriptor of all the dense grid points from the database. Let us denote by $G_j = (\mathcal{V}_j, \mathcal{E}_j, W_j)$ an *undirected weighted* graph constructed on the image I_j. The set $\mathcal{V}$ of vertices contains a regularly sampled subset of pixels $\mathcal{P}$ of the image and at the limit can contain all of them. The graph edges $\mathcal{E}$ connect these vertices with a 8-connected neighbourhood system. The adjacency matrix W

of size $|\mathcal{V}| \times |\mathcal{V}|$ is defined as:

$$W_{pq} = \begin{cases} w_{pq} & \text{if } p,q \in \mathcal{E} \\ 0 & \text{if } p,q \notin \mathcal{E}, \end{cases}$$

where w_{pq} represents the edge-based similarity between two vertices p and q. For each image I_j in the database, we aim to partition the graph G_j into K disjoint subgraphs $\{g_j^1, \ldots, g_j^K\}$, such that $\forall k \neq l, g_j^k \cap g_j^l = \emptyset$ and $G_j = \bigcup_{k=1}^{K} g_j^k$. We denote this K-way partitioning by $\Gamma_j^K = \{g_j^1, \ldots, g_j^K\}$.

The graph partitioning problem can be recast as a labeling problem. Given a set of vertices $\mathcal{V}$ and a finite set of labels $L = \{1, 2, \ldots, K\}$, for all node $p \in \mathcal{V}$, we are looking for the optimal label $l_p \in L$, such that the joint labeling $\mathcal{L} = \{l_1, \ldots, l_{|\mathcal{V}|}\} \in L^{|\mathcal{V}|}$ satisfies a specified objective function.

III. Bag-of-Bags of Words Model

Our BBoW model has been inspired by *bag-of-words model* [1] for image description and the idea that an irregular partition into a fixed number of graphs is a step in-between an arbitrary regular partition as in [4], and a segmentation of an image plane which can be strongly redundant. The imperfections of this intermediate partition should be compensated by a statistical nature of the original BoW model for each graph in the partition. Furthermore, we embed BBoW into multi-resolution schemes as *spatial pyramid matching* [4] for using image partitions in coarse to fine manner. For each image, construction of a bag-of-bags of words (see figure 1) can be decomposed into four main steps:

1) Select a reduced number of pixels $\mathcal{V}$ from the whole image.
2) Build an initial graph G.
3) Partition the graph G into K subgraphs.
4) Compute a signature for each subgraph.

The signature of a subgraph is a histogram obtained by assigning each feature of this subgraph to the closest visual word in the codebook. Hence, an image being composed of K subgraphs is characterized by a set of K histograms. In the following, we detail each step of the method.

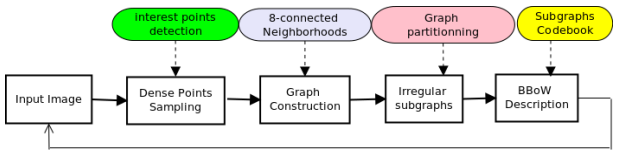


Fig. 1: Bag-of-bags of words pipeline

A. Graph nodes selection

We adopt a dense sampling strategy, *i.e.* sample the image at regular grid points. All the areas contribute equally to the final image representation. We choose 8-pixels spacing and 16-pixels patch size for the SIFT [10] features, called *Bag-of-Features Grid SIFT (BF-GSIFT)* [11].

B. Construct initial weighted graph

The set of nodes $\mathcal{V}$ of our initial graph G contains all the points resulting from the dense sampling. The edges $\mathcal{E}$ are obtained by linking these points with a 8-connected neighbourhood system. As mentioned in the introduction, we have tested two different methods to achieve graph partitioning, namely Graph Cuts and Normalized Cuts. For both, the same weights on the edges are used:

$$w_{pq} = \exp\left(-\lambda \left(\bar{C}_p - \bar{C}_q\right) \Sigma^{-1} \left(\bar{C}_p - \bar{C}_q\right)^T\right), \qquad (1)$$

where $\bar{C}_p$ accounts for the mean color vector over a $n \times n$ patch centred on point p. We use the YUV color space that has independent color channels and allows to better deal with changes in lightning conditions. We denote $Y_p = Y(x,y)$ (respectively $U_p = U(x,y)$ and $V_p = V(x,y)$) as the color channel value at point $p = (x,y)$ in the whole image coordinate system. The mean color vector becomes $\bar{C}_p = (\bar{Y}_p, \bar{U}_p, \bar{V}_p)$ where:

$$\bar{Y}_p = \frac{1}{n^2} \sum_{u=-n}^{n} \sum_{v=-n}^{n} Y(x+u, y+v).$$

The covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_Y^2 & 0 & 0 \\ 0 & \sigma_U^2 & 0 \\ 0 & 0 & \sigma_V^2 \end{bmatrix}$$

is diagonal because of channels' independence. As in [12], the covariance is defined for each channel as:

$$\sigma^2 = \langle (Y_p - Y_q)^2 \rangle$$

where $\langle . \rangle$ denotes the expectation over all the edges $(p,q) \in \mathcal{E}$ of the graph G.

C. Graph Partitioning

Let us now present how the initial graph G is partitioned into K subgraphs. We investigated two different algorithms for that purpose. In the following, the details for both of them: Normalized Cuts and Graph Cuts are given.

1) Partitioning with Normalized Cuts: Normalized Cuts [9] is a well known method in data clustering and image analysis. The principle is to divide the graphs into subgraphs such that the similarity between nodes within a subgraph is greater than the similarity between nodes in separated subgraphs. The algorithm then optimizes two criteria measuring the total dissimilarity between the different partitions as well as the similarity within each partition. The Normalized Cuts method can directly be applied to partition a weighted graph G. In practice, we used the code from [9], [13].

2) Partitioning with Graph Cuts: Graph Cuts is another very famous method for graph partitioning in the computer vision community. Graph Cuts [8], [14], [15] are based on min-cut/max-flow optimization and are used to minimize energy functions containing a data term and a smoothness term. In our case, the energy reads:

$$\begin{aligned} E(\mathcal{L}) &= E_{data}(\mathcal{L}) + E_{smooth}(\mathcal{L}) \\ &= \sum_{p \in \mathcal{V}} E_d(p, l_p) + \sum_{(p,q) \in \mathcal{E}} E_s(p,q) \\ &= \sum_{p \in S} E_d(l_p) + \sum_{(p,q) \in \mathcal{E}} E_s(p,q). \end{aligned}$$

The first term is the data term. It is only applied to seed points $p \in S$, where $S = \{s_1, \ldots s_K\}$ is the set of seeds. Seed points are nodes that we select in the following way. The image is divided into K regular cells, K being the number of partitions we are looking for. The seed points are then the nodes that are

Fig. 2: Seed points. In red: the nodes of the graphs. In other colors: the seed points

the closest to the barycentres of these cells (figure 2). The goal of the data term is to fix the label of the seed points. The seed points are then used to impose hard constrains on Graph Cuts. For instance, the seed point s_i must end up with label i after the energy minimization process. This idea can be formulated as in [16]:

$$E_d(l_p) = \begin{cases} 0 & \text{if } p = s_i \ \& \ l_p = i \\ \infty & \text{otherwise.} \end{cases} \quad (2)$$

The smoothness term on the other hand, is directly linked to the weights of the graph, as defined in equation (1) :

$$E_s(p,q) = \frac{w_{pq}}{\|p-q\|^2}\big(1-\delta(l_p,l_q)\big). \quad (3)$$

where $\delta(.)$ is the Kronecker function.

D. Signature of subgraphs

As in the standard BoW approach, the signature of each subgraph $g_j, j = 1 \dots K$ is a histogram H_j. It is obtained by assigning the SIFT descriptor of each of its nodes $p \in \mathcal{V}_j$ to the closest word in the codebook computed on the whole database. As mentioned before, the codebook is computed from all the dense grid points in a training image dataset. In other words, we do not rely on our graph partitions to construct the visual dictionary.

IV. IPMK: Irregular Pyramid Matching Kernel

In this section, we explain how our bag-of-bags of words model can be used in an image retrieval system to match images.

Our strategy directly follows the one from Spatial Pyramid Matching [4]. Let us assume that we compute a sequence of subgraphs at different resolutions $r = 1 \dots R$ of an image. For each image I_i, we end up with a set $H_i^r = \{H_{i,1}^r, \dots, H_{i,K_r}^r\}$ of K_r histograms at each resolution r. Example of subgraphs and their histograms are visible in figure 3. As with SPM, at each resolution the number of partitions is given by $K_r = 2^{2r}$. In order to compare two images I_i and I_j, we use the histogram intersection function [17]:

$$\mathcal{I}\big(H_{i,k}^r, H_{j,k}^r\big) = \sum_{b=1}^{B} \min\big(H_{i,k}^r(b), H_{j,k}^r(b)\big), \quad (4)$$

where B in the codebook size.

In our method we cannot directly apply this function to match histograms of a pair of images. At the end of the partitions steps, for two images, the two histograms $H_{i,k}^r$ and $H_{j,k}^r$ may not correspond one to the other. Indeed, after applying a graph partitioning method, the subgraph attributed

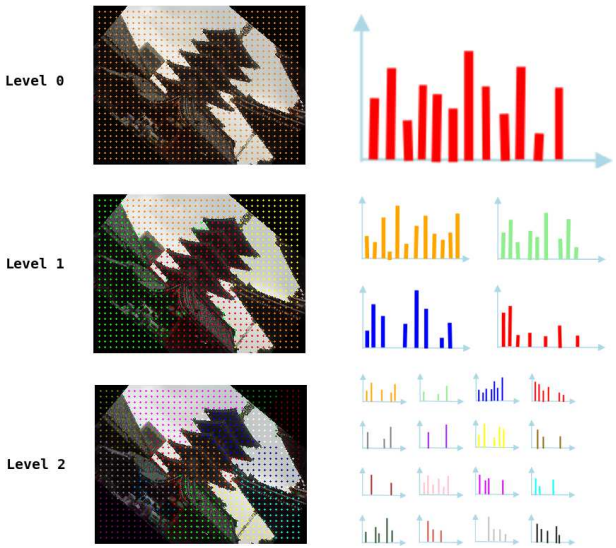


Fig. 3: A schematic illustration of the bag-of-bags of words representation at each resolution of the pyramid. At level 0, the decomposition consists of just a single label, and the representation is equivalent to the classical bag-of-words. At level 1, the image is subdivided into four partitions, leading four features histograms, and so on.

to label k can be at any position in the image or represent any object of the image. The spatial arrangement is lost if an image in a database undergoes rotation for instance. To overcome this issue, we first need to reorganize the labels of our histograms. For this purpose, we reorganize the labels between the two sets of histograms $\{H_{i,k}^r\}_{k=1,\dots,K_r}$ and $\{H_{j,k}^r\}_{k=1,\dots,K_r}$. We call this step *bipartite subgraphs matching*.

To rearrange the labels, we are facing an assignment problem which requires the use of combinatorial algorithm. In our method, we rely on the Hungarian algorithm to minimize the discrete optimal transport between the two set of histograms. The cost matrix $D_{i,j}^r$ between the pair of images I_i and I_j at level r reads:

$$D_{i,j}^r = \begin{bmatrix} d_{i,j}^{11} & \dots & d_{i,j}^{1K_r} \\ & \vdots & \\ d_{i,j}^{K_r 1} & \dots & d_{i,j}^{K_r K_r} \end{bmatrix}. \quad (5)$$

where

$$d_{i,j}^{k,l} = \sum_{b=1}^{B} |H_{i,k}^r(b) - H_{j,l}^r(b)|$$

is the L_1 distance between $H_{i,k}^r$ and $H_{j,l}^r$, which is equivalent to the *histogram intersection* kernel. The Hungarian algorithm finds the minimum cost assignment by associating each label k in image I_i to one label $k' = f_i(k,r)$ in image I_j thus the pairs of histograms to compare are identified.

Now that bipartite subgraphs matching has been achieved, we can directly apply the Spatial Pyramid Matching scheme by using the pyramid match kernel:

$$\begin{aligned} \kappa(I_i, I_j) &= \mathcal{I}\big(H_{i,k}^R, H_{j,k'}^R\big) + \sum_{r=0}^{R-1} \frac{1}{2^{R-r}}\big(\mathcal{I}^r - \mathcal{I}^{r+1}\big) \\ &= \frac{1}{2^R}\mathcal{I}\big(H_{i,1}^0, H_{j,1}^0\big) + \sum_{r=1}^{R} \frac{1}{2^{R-r+1}}\mathcal{I}\big(H_{i,k}^r, H_{j,k'}^r\big). \end{aligned}$$

The weight $\frac{1}{2^{R-r}}$ before the histogram intersection function, which is proportional to number of subgraphs, allows to give more influence to penalize higher resolutions, reflecting the fact that higher levels (*i.e.* smaller resolutions) localize the features more precisely.

V. Experimental evaluation

In this section, we present the experimental evaluation on Caltech101 benchmark [18] and the people-playing-musical-instrument (PPMI) [1] dataset [19].

The Caltech101 dataset includes 101 distinct object such as faces, watches, ants, pianos, etc... and a background category for a total of 102 categories. This benchmark acts as a common standard for comparison of different algorithms without bias due to different datasets. We follow the same setting as in [4]. We use exactly the same grid points as generated from the code provided by the authors of [4] to build the initial graphs. The dictionary is built over 30 images per class and the retrieval is run on the rest of the images.

The PPMI dataset emphasizes on understanding subtle interactions between humans and objects. For twelve musical instruments, it contains images of people either playing or just holding the instrument. The twelve binary classes are used in our experiments. For each class, 100 normalized PPMI+ images are randomly selected for training and the remaining 100 images for testing.

A. Parameters evaluation

We start by testing the parameters' influence of our method for image retrieval. Two parameters have been evaluated: the patch size n defined in section III-B and the parameter λ from the edges weight (equation 1). The mean average precisions on the two datasets are presented in tables I and II. For efficiency, these experiments were conducted on smaller databases containing only 4 test images per category. These results highlight that these parameters do not have much influence on the quality of results using either Graph Cuts (GC) or Normalized Cuts (NC). For the following experiments, we therefore decided to use $n = 5$ and $\lambda = 5$.

TABLE I: Influence of parameter λ (equation 1) on image retrieval. For each value, the mean average precision is given. For this experiment, the patch size is set to $n = 5$.

	Mean Average Precision for the IMPK approach with 3 levels			
	$\lambda = 0$	5	20	100
NC	0.386079	**0.389737**	0.386185	0.380554
GC	0.373073	**0.382115**	0.377872	0.375660

TABLE II: Influence of parameter n (section III-B) on image retrieval. For each value, the mean average precision is given. For this experiment, $\lambda = 5$.

	Mean Average Precision for the IMPK approach with 3 levels			
	n = 3	5	7	9
NC	0.380546	0.388987	**0.389456**	0.388229
GC	0.379723	**0.382115**	0.377215	0.375741

[1] http://ai.stanford.edu/ bangpeng/ppmi.html

B. Image retrieval evaluation

The performance of our algorithm is measured by computing the mean Average Precision (mAP) for each category, as described in TREC-style evaluation[2]. We also add the computation of the mAP and the standard deviation for the whole dataset. We compare with SPM [4] and with the bag-of-words approach which simply corresponds to the results at level 0. Tables III and IV present this global mAP for the two datasets at each level of the pyramid. SPM globally outperforms our method, but we outperform BoW (level 0) both with Graph Cuts (GC) and Normalized Cut (NC). When looking closer to the results, we can nevertheless observe that the pyramidal approach is of no need for SPM on the dataset used. Indeed, the results at level 2 are better than the ones with the pyramid approach. These results have been obtained with the codes from the authors of SPM. On the contrary, the pyramidal approach is a good strategy in our case, as it improves the retrieval performance.

TABLE III: Retrieval performance (mAP and standard deviation) on Caltech-101 dataset.

	Level 0 (BoW)	Level 1	Level 2	Pyramid
NC	0.117 $\pm$0.18	0.107 $\pm$0.16	0.125 $\pm$0.19	0.129 $\pm$0.20
GC	0.117 $\pm$0.18	0.099 $\pm$**0.15**	0.114 $\pm$**0.18**	0.122 $\pm$**0.18**
SPM	0.117 $\pm$0.18	0.144 $\pm$0.21	0.162 $\pm$0.23	0.157 $\pm$0.22

TABLE IV: Retrieval performance (mAP and standard deviation) on PPMI dataset.

	Level 0 (BoW)	Level 1	Level 2	Pyramid
NC	0.109 $\pm$0.031	0.104 $\pm$0.022	0.106 $\pm$0.025	0.108 $\pm$0.027
GC	0.109 $\pm$0.031	0.104 $\pm$0.022	0.106 $\pm$0.024	0.108 $\pm$0.027
SPM	0.109 $\pm$0.031	0.115 $\pm$0.036	0.121 $\pm$0.043	0.118 $\pm$0.041

We now propose to look into more details to the results for certain categories. A comparison of the mAP on sixteen categories from Caltech-101 dataset is presented on figure 4. The result highlights the fact that for a few number of categories SPM has a much higher retrieval performance than our method. These categories explain the importance difference in the global performance on all the database (as described previously). Nevertheless for most of the categories, we are as good or better than SPM with either Graph Cuts or Normalized Cuts. In order to give a deeper understanding on why our method suits for some catagories and not for the others, let us now look at the graph partitions obtained by the graph cutting methods. Figure 5 presents the results of the partitioning at level 2 (16 partitions) with either Normalized Cuts (NC) or Graph Cuts (GC). The first two lines show images from two different categories for which our method clearly outperforms SPM. For the minaret category, IPMK with Normalized Cuts gives the best results. As can be seen on the first row of figure 5, our method is able to represent the minaret with only one or two partitions. The other partitions in the background are stable in the sense that almost the same partitions can be found in each image. The second row presents the Graph Cuts results on four images from the beaver category. Once again, it can be observed that the partitions are stable. The four last rows on the other hand represents two categories for which our

[2] http://trec.nist.gov/

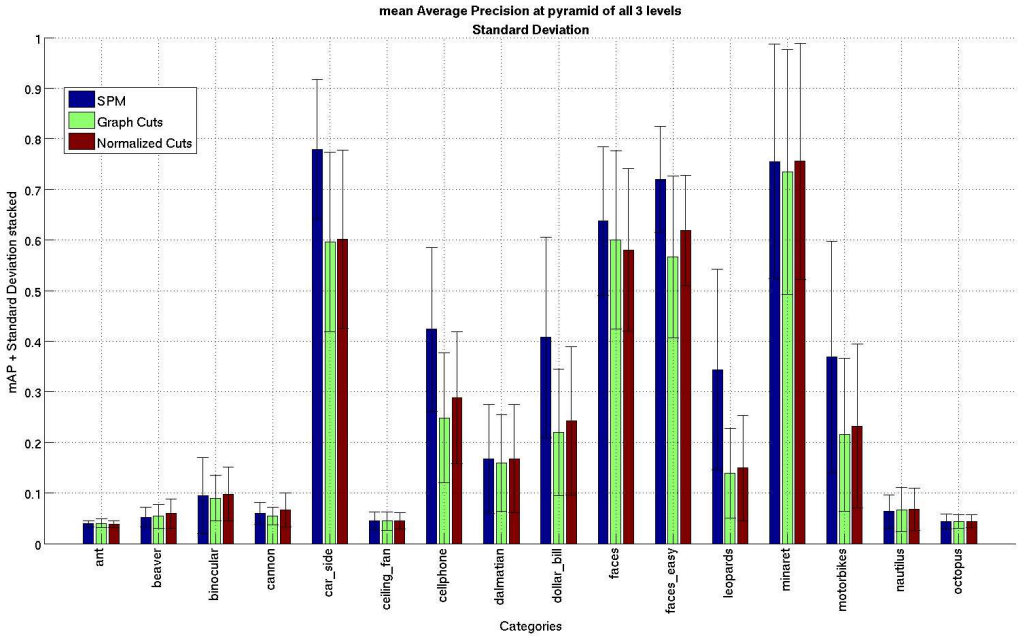


Fig. 4: The mean Average Precision for 16 typical categories in Caltech101

method fails. The partitions obtained almost do not exhibit any consistency across the images from the same category. The resulting histograms are therefore each time very different and we can obviously not expect any good retrieval.

VI. Conclusion and perspectives

In this paper, the bag-of-bags of words model over irregular image partition with graph cut methods and its pyramidal extension, the irregular pyramid matching kernel have been introduced. Graph cuts methods are used to separate an image graph which is a dense grid of points into several subgraphs, each then being described by a bag-of-words model. Our approach incorporates color and limited spatial information into image representation for image retrieval. The obtained results are encouraging, especially when the partitions obtained are stable across all the images from the same category. Nevertheless, this stability is not always insured with this preliminary work. We will now be focusing on improving the weights of the graph in order to reach more stability.

Acknowledgment

This work was partially supported by CNRS (Centre national de la recherche scientifique) & Region of Aquitaine Grant.

References

[1] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *ICCV*, 2003, pp. 1470 –1477.

[2] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV*, 2003.

[3] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *In ECCV workshop on statistical learning in computer vision*, 2004.

[4] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[5] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in *ICCV*, 2011.

[6] J. Gibert, E. Valveny, and H. Bunke, "Graph embedding in vector spaces by node attribute statistics," *Pattern Recognition*, vol. 45, no. 9, pp. 3072 – 3083, 2012.

[7] H. Bunke and K. Riesen, "Towards the unification of structural and statistical pattern recognition," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 811 – 825, 2012.

[8] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 23, no. 11, pp. 1222–1239, 2001.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 22, no. 8, pp. 888–905, 2000.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. journ. of comp. vis. (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[11] T. Tuytelaars, "Dense interest points," in *CVPR*, 2010, pp. 2281–2288.

[12] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[13] T. Cour, S. X. Yu, and J. Shi, "Normalized cut segmentation code," 2004.

[14] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 26, no. 9, pp. 1124–1137, 2004.

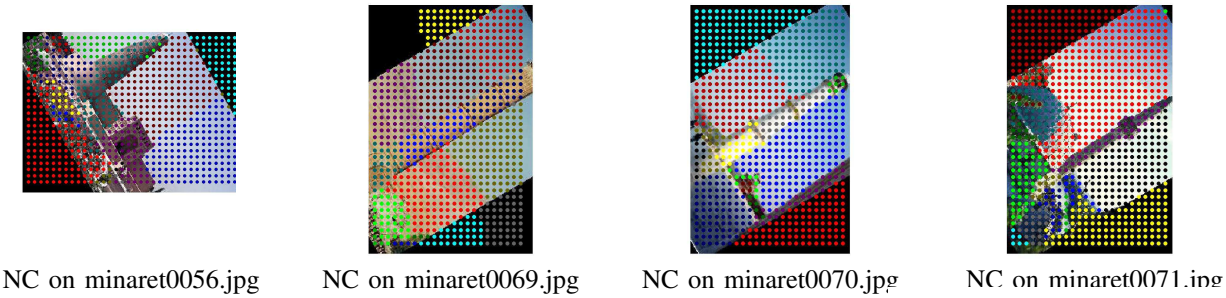
[15] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 26, no. 2, pp. 147–159, 2004.

[16] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *ICCV*, 2001.

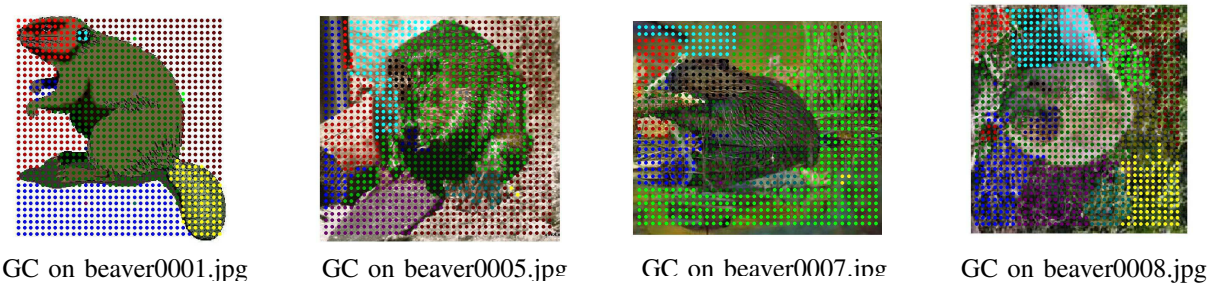
[17] F.-D. Jou, K.-C. Fan, and Y.-L. Chang, "Efficient matching of large-size histograms," *Pattern Recognition Letters*, vol. 25, no. 3, pp. 277–286, 2004.

[18] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," *Workshop on Generative-Model Bas. Vis.*, 2004.

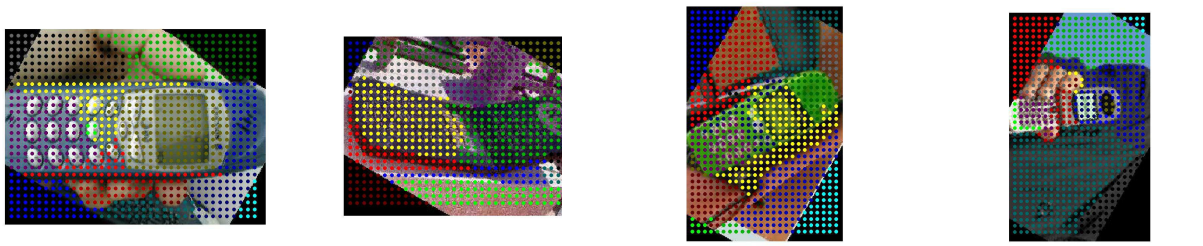
[19] B. Yao and L. Fei-Fei, "Grouplet: A structured image representation for recognizing human and object interactions," in *CVPR*, 2010.



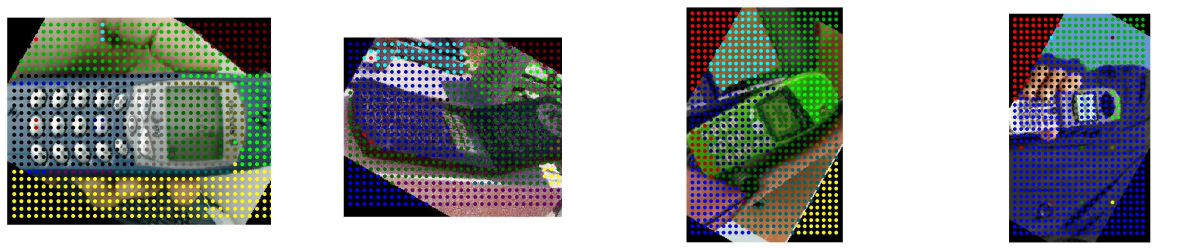
NC on minaret0056.jpg NC on minaret0069.jpg NC on minaret0070.jpg NC on minaret0071.jpg



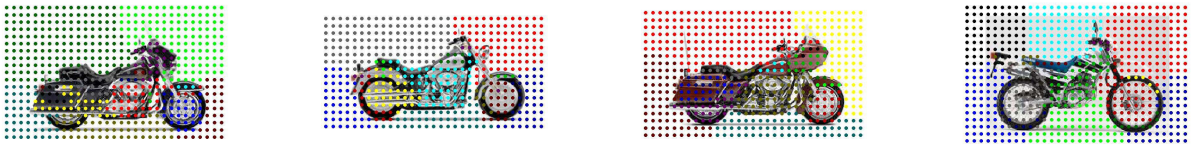
GC on beaver0001.jpg GC on beaver0005.jpg GC on beaver0007.jpg GC on beaver0008.jpg



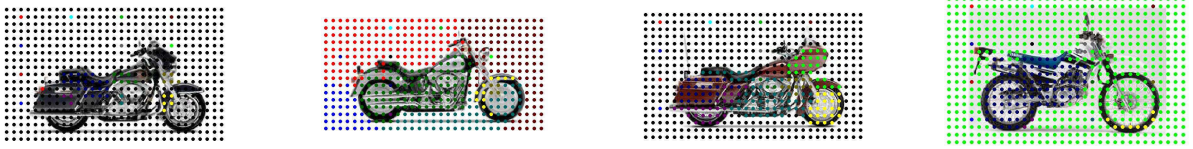
NC on cellphone0004.jpg NC on cellphone0007.jpg NC on cellphone0039.jpg NC on cellphone0041.jpg



GC on cellphone0004.jpg GC on cellphone0007.jpg GC on cellphone0039.jpg GC on cellphone0041.jpg



NC on motorbikes0005.jpg NC on motorbikes0007.jpg NC on motorbikes0009.jpg NC on motorbikes0059.jpg



GC on motorbikes0005.jpg GC on motorbikes0007.jpg GC on motorbikes0009.jpg GC on motorbikes0059.jpg

Fig. 5: Graph partioning with $K = 16$ results on several images from the Caltech-101 benchmark. GC: Graph Cuts, NC: Normalized Cuts.