

Semantics-based Image Retrieval by Region Saliency

Wei Wang, Yuqing Song and Aidong Zhang

Department of Computer Science and Engineering,
State University of New York at Buffalo,
Buffalo, NY 14260, USA
{wwang3, ys2, azhang}@cse.buffalo.edu

Abstract. We propose a new approach for semantics-based image retrieval. We use color-texture classification to generate the codebook which is used to segment images into regions. The content of a region is characterized by its self-saliency and the lower-level features of the region, including color and texture. The context of regions in an image describes their relationships, which are related to their relative-salencies. High-level (semantics-based) querying and query-by-example are supported on the basis of the content and context of image regions. The experimental results demonstrate the effectiveness of our approach.

1 Introduction

Although the content-based image retrieval (CBIR) techniques based on low-level features such as color, texture, and shape have been extensively explored, their effectiveness and efficiency are not satisfactory. The ultimate goal of image retrieval is to provide the users with the facility to manage large image databases in an automatic, flexible and efficient way. Therefore, image retrieval systems should be armed to support high-level (semantics-based) querying and browsing of images.

The basic elements to carry semantic information are the image regions which correspond to semantic objects, if the image segmentation is effective. Most approaches proposed for region-based image retrieval, although successful in some aspects, could not integrate the semantic descriptions into the regions, therefore cannot support the high-level querying of images. After the regions are obtained, proper representation of the content and context remains a challenge.

In our previous work [7], we used color-texture classification to generate the semantic codebook which is used to segment images into regions. The content and context of regions were then extracted in a probabilistic manner and used to perform the retrieval. The computation of the content and context needed some heuristic weight functions, which can be improved to combine visual features. In addition, when retrieving images, users may be interested in both semantics and some specific visual features of images. Thus the content and context of image regions should be refined to incorporate different visual features. The salencies of the image regions [2], [3], [6] represent the important visual cues of regions, therefore can be used to improve our previous method. In this paper, we will introduce an improved image-retrieval method, which incorporates region saliency to the definition of content and context of image regions.

The saliencies of regions have been used to detect the *region of interest* (ROI). In [6], saliency was further categorized as *self-saliency* and *relative-saliency*. Self-saliency was defined as “what determines how conspicuous a region is on its own”, while relative-saliency was used to measure how distinctive the region appears among other regions. Apparently, saliencies of regions represent the intrinsic properties and relationships for image regions.

Our approach consists of three levels. At the pixel level, color-texture classification is used to form the semantic codebook. At the region level, the semantic codebook is used to segment the images into regions. At the image level, content and context of image regions are defined and represented on the basis of their saliencies to support the semantics retrieval from images. The three levels are illustrated in Figure 1.

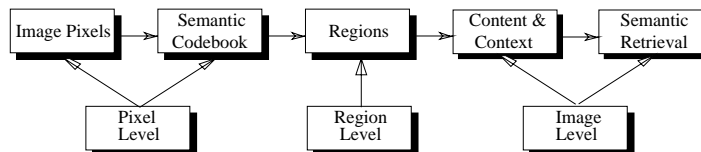


Fig. 1. System design of our approach.

The remainder of the paper is organized as follows. From Section 2 to Section 5, each step of our approach is elaborated. In Section 6 experimental results will be presented, and we will conclude in Section 7.

2 Image segmentation

2.1 Generation of the semantic codebook

We generate the semantic codebook by using color-texture classification to classify the color-texture feature vectors for pixels in the images into cells in the color-texture space. About the color-texture feature vectors for each pixel, the three color features we choose are the averaged *RGB* values in a neighborhood and the three texture features are *anisotropy*, *contrast* and *flowOrEdge*. *Anisotropy* measures the energy comparison between the dominant orientation and its orthogonal direction. *Contrast* reflects the contrast, or harshness of the neighborhood. *FlowOrEdge* can distinguish whether an 1D texture is a flow or an edge. (See [7] for detailed description of these values). The color and texture feature vectors are denoted as $Color_{FV}$ and $Texture_{FV}$, respectively. Therefore for each pixel, we have a six-dimensional feature vector where three dimensions are for color, and three for texture. The color-texture classification is performed in the following way: by $Color_{FV}$, the color space is uniformly quantized into $4 \times 4 \times 4 = 64$ classes; by $Texture_{FV}$, texture space is classified into 7 classes: one class for low contrast and edge, respectively; two classes for flow and three for 2D texture. Formal definition of these classes can be found in [7]. Because edges cannot reflect the color-texture characteristics of image regions semantically, we classify all the pixels, except those pixels corresponding to edges, to 6 texture classes

LowContrast, 2D₀, 2D₁, 2D₂, Flow₀, Flow₁. Corresponding to $T(T = 6)$ texture classes and $C(C = 64)$ color classes, we now split the color-texture space (denoted as CTS , excluding the pixels on the edges) to $C \times T$ cells.

We then define the major semantic categories based on the semantic constraints of the images in the database: $SC = \{sc_1, \dots, sc_M\}$. For each semantic category SC_i , certain number of images are chosen to be the training images such that the semantic codebook generated from them can represent as much as possible the color-texture characteristics of all images in the database belonging to the SC . The set of all pixels in the training images, except those in class *Edge*, is denoted as *TrainingPixels*. For each pixel in *TrainingPixels*, its feature vector will fall into one cell of CTS . After all the pixels in *TrainingPixels* are classified, for each cell in CTS we count the number of pixels in it. Only those cells whose number of pixels exceeds a threshold will be one entry of the semantic codebook of the that database. Therefore the size of semantic codebook will be less or equal to $C \times T$. In the following discussion, we use $SCDB$ to denote the semantic codebook for color-texture space, and suppose its size is N .

2.2 Image segmentation based on semantic codebook

We segment images by $SCDB$ in this way: for each image I , we extract the color-texture feature vector for each pixel q of I . For each feature vector, we find the cell in CTS where the pixel belongs to. If the cell is one of $SCDB$'s entries, q is replaced with the *index* of that entry; otherwise its value is set to $C \times T + 1$ to distinguish it from any valid entry of the semantic codebook. After the process, I becomes a matrix of indices corresponding to entries in $SCDB$. Because the number of valuable regions in an image is usually less than 5 in our experiments, we only choose 5 most dominant indices (referred as *DOMINANT*) and use *DOMINANT* to re-index the pixels with indices not present in *DOMINANT*¹. Finally we run the encoded images through a connected-components algorithm and remove the small regions (with area less than 200 pixels).

3 Representation of saliency for regions and semantic categories

As stated before, saliency features of the regions can be either self-saliency or relative-saliency. Self-saliency features are computed by the visual features of the regions. Relative-saliency features are computed by taking the value of a feature for the current region and comparing it with the average value of that feature in the neighboring regions. Gaussian and Sigmoid functions are used to map the feature values to the interval $[0, 1]$ for the purpose of normalization and integration of different saliency features. Self and relative-saliency features we are using include:

(1) *Self-saliency features of image regions:*

- *Size (S_{size}):* $S_{size}(R_i) = \max(\frac{A(R_i)}{A_{max}}, 1.0)$. Here $A(R_i)$ is the area of region R_i and A_{max} is a constant used prevent excessive saliency being given to very large regions. It

¹ re-index means to find the closest codebook entry in *DOMINANT*, not in the whole $SCDB$.

was shown in [3] that larger regions are more likely to have larger saliency. However a saturation points exists, after which the size saliency levels off. A_{max} is set to the 5% of the total image area. The value is in the interval [0, 1].

- *Eccentricity* (S_{ecce} , for shape): $S_{ecce}(R_i) = \frac{major_axis(R_i)}{minor_axis(R_i)}$. A Gaussian function maps the value to the interval [0, 1].
- *Circularity* (S_{circ} , for shape): $S_{circ}(R_i) = \frac{Peri(R_i)^2}{A(R_i)}$. Here $Peri(R_i)$ is the perimeter of the region R_i and a Sigmoid function maps the value to the interval [0, 1].
- *Perpendicularity* (S_{perp} , for shape): $S_{perp}(R_i) = \frac{angle(major_axis(R_i))}{\pi/2}$. A Sigmoid function maps the value to the interval [0, 1].
- *Location* (S_{loca}): $S_{loca}(R_i) = \frac{center(R_i)}{A(R_i)}$. Here $center(R_i)$ is the number of pixels in region R_i which are also in the center 25% of the image.

(2) *Relative-saliency features of image regions*: similar to [2], relative-saliency features are computed as:

$$(Relative - saliency)_{R_i}^{feature} = \sum_{R_i \in NR, i \neq 0} \frac{\|feature_{R_0} - feature_{R_i}\|}{feature_{R_i}}$$

Here NR refers to the neighboring regions. We use *Brightness* and *Location* as the features and compute *Relative brightness* and *relative location* as the relative-saliency features in the system. The values are mapped to interval [0, 1] using Sigmoid functions.

(3) *Combining self and relative-saliency to generate the saliency of the regions and semantic categories*: It is not necessary that all the saliency features will be used for each semantic categories. We choose particular saliency feature(s) to represent the most dominant visual features of each semantic categories, if possible. For instance, assume we have a category “water falls”. We will use *Eccentricity* and *Perpendicularity* as well as *Relative brightness* for the saliency features because usually “water falls” will be long and narrow, and can be approximately described as perpendicular to the ground. Another example is the category “flower”, the *Circularity* and *Location* as well as *Relative brightness* are used since usually flower will be round and in the middle of the images and brighter compared with surrounding scenes. A table in Section 6 lists the selection of the saliency features for all the semantic categories we have in the system.

Because all the self and relative-saliency values are normalized to the interval [0, 1], we can simply add them together as the *Saliency of the region* with regard to a certain semantic category, denoted as $Sal(R_i, SC_j)$, for the region R_i and the semantic category SC_j . For all the regions in training images that represent the semantic category SC_j , we calculate the mean and variance of their saliency and store them as the *Saliency of the semantic category*, denoted as μ_j and σ_j for the SC_j .

4 Representation of content and context of regions

By collecting the statistical data from the training images, we derive the logic to represent the content and context for all the images in the database for the semantics retrieval. We first generate the statistical data from the training images. For each entry e_i in the

semantic codebook $SCDB$, and each semantic category SC_j , we count the number of regions R in the training images such that (i) the pixels in R belong to e_i (e_i is a cell in the CTS); and (ii) the region R represents an object belong to the category SC_j . The result is stored in a table called *cell-category statistics table*. In addition, we count the times that two or three different categories present in the same training images. The result is stored in a table, called *category-category statistics table*.

Based on the cell-category statistics table, for each codebook entry of $SCDB$, we can calculate its probability of representing a certain semantic category. Let N be the size of $SCDB$, M be the number of SC , $i \rightarrow SC_j, i \in [0 \dots N - 1], j \in [0 \dots M - 1]$ denote the event that index of $SCDB$ i represents semantics described in SC_j . The probability of the event $i \rightarrow SC_j$ is $P(i \rightarrow SC_j) = \frac{T(i \rightarrow SC_j)}{T(i)}$, where $T(e)$ represents event e 's presence times in the training images.

Based on the category-category statistics table, we define the Bi-correlation factor and Tri-correlation factor, for representing the context of regions.

Definition 1 For any two different semantic categories SC_i and SC_j , we define the **Bi-correlation factor** $B_{i,j}$ as: $B_{i,j} = \frac{n_{i,j}}{n_i + n_j - n_{i,j}}$.

Definition 2 For any three different semantic categories SC_i, SC_j and SC_k , we define the **Tri-correlation factor** $T_{i,j,k}$ as: $T_{i,j,k} = \frac{n_{i,j,k}}{n_i + n_j + n_k - n_{i,j} - n_{i,k} - n_{j,k} + n_{i,j,k}}$. Here $n_i, n_{i,j}$ and $n_{i,j,k}$ are entries in the category-category statistics table. Bi and Tri-correlation factors reflect the correlation between two or three different categories within the scope of training images. If the training images are selected suitably, they reflect the relationship of pre-defined semantic categories we are interested in.

Armed with the statistical data we have generated from training images, we can define and extract content and context of regions for all the images in the database. Assume we have an image I with number of Q regions. The $SCDB$'s codebook indices of regions are $C_i, i \in [0 \dots Q - 1]$, and regions are represented by $R_i, i \in [0 \dots Q - 1]$, and each C_i is associated with i_N possible semantic categories $SC_{i_{j(i)}}, i_{j(i)} \in [0 \dots N - 1], j(i) \in [0 \dots i_N - 1]^2$. For I , let P_{all} be the set of all possible combinations of *indices* of semantic categories represented by regions $R_i, i \in [0 \dots Q - 1]$. We have

$$P_{all} = \{(0_{j(0)}, \dots, i_{j(i)}, \dots, Q - 1_{j(Q-1)}) \mid \forall i, P(C_i \rightarrow SC_{i_{j(i)}}) > 0, i \in [0 \dots Q - 1], \\ i_{j(i)} \in [0 \dots N - 1], j(i) \in [0 \dots i_N - 1]\}$$

Note there are totally $\prod_{i=0}^{Q-1} i_N$ possible combinations for P_{all} , therefore P_{all} has $\prod_{i=0}^{Q-1} i_N$ tuples of semantic categories indices, with each tuple having Q fields.

Corresponding to each tuple $\kappa \in P_{all}$, for $Q \geq 2$ we have $B(\kappa) = \sum_{p,q \in \kappa, p < q} B_{p,q}$, $\kappa \in P_{all}$, and for $Q \geq 3$ we have $T(\kappa) = \sum_{p,q,r \in \kappa, p < q < r} T_{p,q,r}$, $\kappa \in P_{all}$. Here p, q, r are the indices belonging to tuple κ , $B(\kappa)$ represents the sum of Bi-correlation factors of different semantic categories with regard to tuple κ , and $T(\kappa)$ represents the sum of Tri-correlation factors of different semantic categories with regard to tuple κ .

² N is the size of $SCDB$

Definition 3 We define **Context** $C_{Score}(\kappa)$ of I as: $C_{Score}(\kappa) = \frac{1}{Norm(\kappa)}(B(\kappa) + \beta T(\kappa))$, $\kappa \in P_{all}$ here $Norm(\kappa)$ is the normalization function with tuple κ , β is the weight for $T(\kappa)$, since $T(\kappa)$ will be more effective in distinguishing contexts of images than $B(\kappa)$ ³. We normalize the C_{Score} because several region indices may point to the same semantic category, we need to guarantee that removal the redundant semantic category will not influence the effectiveness of C_{Score} .

Definition 4 We define **ProbScore** $P_{Score}(\kappa)$ of I as: $P_{Score}(\kappa) = \sum_{i=0}^{Q-1} w(R_i, SC_{i_j})P(C_i \rightarrow SC_{i_j})$, $i_j \in \kappa$, $\kappa \in P_{all}$ $P_{Score}(\kappa)$ represents the probability score corresponding to tuple κ , here $w(R_i, SC_{i_j})$ is the weight function with regard to region R_i and semantic category SC_{i_j} , it can be determined by using the saliency of the region and the semantic category:

$$w(R_i, SC_{i_j}) = \frac{1}{\sigma_{i_j} \sqrt{2\pi}} e^{-\frac{(Sal(R_i, SC_{i_j}) - \mu_{i_j})^2}{2\sigma_{i_j}^2}}$$

where $Sal(R_i, SC_{i_j})$ is the saliency of region R_i with regard to the semantic category SC_{i_j} , μ_{i_j} and σ_{i_j} are the saliency of the semantic category SC_{i_j} .

Definition 5 We define the **TotalScore** T_{Score} of image I as $T_{Score} = Max\{P_{Score}(\kappa) + \gamma C_{Score}(\kappa) \mid \kappa \in P_{all}\}$ where $Max\{t\}$ represents the maximum value of value t .

Definition 6 We define the **Content** of image I as the semantic categories corresponding to T_{Score} . By computing the maximum value of $P_{Score}(\kappa) + \gamma C_{Score}(\kappa)$ over all tuples in P_{all} , we find the semantic categories that best interpret the semantics of the regions in image I as the *Content* of I . We store the *Content*, T_{Score} and the each region's *SCDB* codebook index and saliency as the final features for I . Note that for those regions whose codebook indices are invalid corresponding to semantic codebook, we will mark its semantic category as "UNKNOWN".

5 Semantics retrieval

Both semantic keyword query and query-by-example are supported in our approach. According to the submitted semantic keywords (corresponding to semantic categories), the system will first find out those images that contain all of the categories (denoted as set RI), then rank the documents by sorting the T_{Score} stored for each image in the database. For those images belonging to RI that has "UNKNOWN" category, its T_{Score} will be multiplied by a diminishing factor. When user submits the query-by-example, if the query image is in the database, its *Content* will be used as query keywords to perform the retrieval, otherwise it will first go through the above steps to obtain the *Content* and T_{Score} . When users are interested in retrieving images with not only same semantics, but also the similar visual features as the query, Euclidean distance of saliency value will be calculated between the query image Q and image I in RI :

$$d(I, Q) = \sqrt{\sum_{i=0}^{W-1} (Sal(I_{R_i}, SC_j) - Sal(Q_{R_i}, SC_j))^2}$$

³ In our experiments, we select $\beta = 10$.

Here assume Q has W regions and I 's region I_{Ri} has same semantic category SC_j with Q 's region Q_{Ri} . $d(I, Q)$ indicates the distance of saliency between I and Q and therefore can be used to fine tune T_{Score} .

6 Experiments

To test the effectiveness of our approach in retrieving the images, we conduct experiments to compare the performance between our approach (with and without using the saliency features) and the traditional CBIR techniques including Keyblock [1], color histogram [5], color coherent vector [4]. The comparison is made by the precisions and recalls of each method on all the semantic categories. In the following table, we define 10 semantic categories and list the saliency features adopted for each semantic category as ‘‘Y’’, otherwise as ‘‘N’’. Abbreviations represent the saliency features: S–Size, E–Eccentricity, C–Circularity, P–Perpendicularity, L–Location, RB–Relative brightness, RL–Relative location.

Category	Explanation	No. in <i>db</i>	S	E	C	P	L	RB	RL
SKY	Sky (no sunset)	1323	N	N	N	N	Y	N	Y
WATER	Water	474	N	N	N	N	Y	N	Y
TREE_GRASS	Tree or Grass	778	Y	N	N	N	N	Y	N
FALLS_RIVER	Falls or Rivers	144	N	Y	N	Y	N	Y	N
FLOWER	Flower	107	N	N	Y	N	Y	Y	N
EARTH_ROCK	Earth or Rocks or								
_MOUNTAIN	Mountain composed of	998	Y	N	N	N	N	N	Y
ICE_SNOW	Ice or Snow or								
_MOUNTAIN	Mountain composed of	204	Y	N	N	N	N	N	Y
SUNSET	Sunset scene	619	N	N	N	N	N	Y	N
NIGHT	Night scene	171	Y	N	N	N	Y	N	N
SHADING	Shading	1709	N	N	N	N	N	N	N

We use an image database with name *COREL* and size 3865. The *COREL* images can be considered as scenery and non-scenery. Scenery part has 2639 images consisting images containing the defined semantic categories, while non-scenery part has 1226 images including different kinds of textures, indoor, animals, molecules, etc. We choose 251 training images from scenery images as training images and form the semantic codebook with size 149. For each semantic category SC_i , we calculate and plot the precision-recall of our approach in the following way. Let *RETRIEVELIST* denote the images retrieved with SC_i . Suppose *RETRIEVELIST* has n images. We calculate the precisions and recalls of first $\frac{n}{30}$, $\frac{2n}{30}$, ..., and n images in *RETRIEVELIST*, respectively.

Since traditional CBIR approaches accepts only query-by-example, we have to solve the problem of comparing the approach of query-by-semantics with query-by-example. Let us take Keyblock as the example of traditional CBIR to show how we choose query sets and calculate the precision-recall for these methods. Suppose user submits a semantic keyword query of semantic category of *Sky*. There are total of 1323 images in

COREL containing *Sky*. For each image containing *Sky*, we use it as a query on *COREL* to select top 100 images by Keyblock, and count the number of images containing *Sky* in the retrieved set. Then we sort the 1323 images descendingly by the numbers of *Sky* images in their corresponding retrieved sets. Let the sorted list be *SKYLIST*. Then we select the first 5% of *SKYLIST* as query set, denoted as *QUERYSET*. Then for each *COREL* image *I*, we calculate shortest distance to *QUERYSET* - {*I*} by Keyblock⁴. The *COREL* images are sorted ascendantly by this distance. Top 1323 *COREL* images are retrieved and we calculate and plot the precision-recall of Keyblock on *Sky*, as we did for our approach.

The average precision-recall on all semantic categories is shown in Figure 2. We can see our method outperforms traditional approaches and retrieval performance improves when saliency features are used.

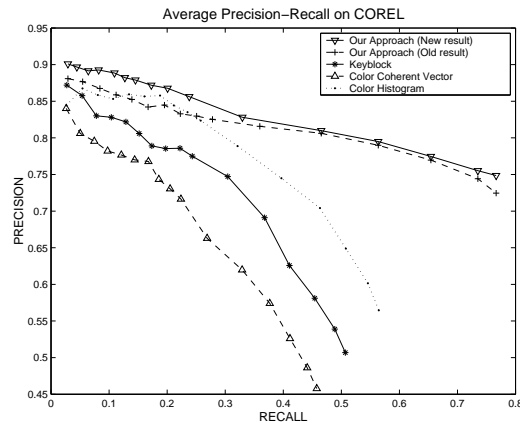


Fig. 2. Average Precision-recall on all the SCs

7 Conclusion

The saliency of image regions, which describes the perceptual importance of the regions, is used for the semantics-based image retrieval. It helps refine the content and context of regions to represent the semantics of regions more precisely. The experimental results show that our approach outperforms the traditional CBIR approaches we compared with.

⁴ images in the *QUERYSET* will have distance zero to *QUERYSET*. Thus the query images will be automatically be retrieved as the top 5% images, which is unfair when making comparison.

8 Acknowledgment

This research is supported by the NSF Digital Government Grant EIA-9983430. The authors thank the anonymous reviewers for their valuable comments to the paper.

References

1. L. Zhu, A. Zhang, A. Rao and R. Srihari. Keyblock: An approach for content-based image retrieval. In *Proceedings of ACM Multimedia 2000*, pages 157–166, Los Angeles, California, USA, Oct 30 - Nov 3 2000.
2. J. Luo and A. Singhal. On measuring low-level saliency in photographic images. In *Proc. IEEE Comp. Vision and Pattern Recognition*, pages Vol. 1 pp 84–89, 2000.
3. W. Osberger and A. J. Maeder. Automatic identification of perceptually important regions in an image. In *Proc. IEEE Int. Conf. Pattern Recognition*, 1998.
4. Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *Proceedings of ACM Multimedia 96*, pages 65–73, Boston MA USA, 1996.
5. M.J. Swain and D. Ballard. Color Indexing. *Int Journal of Computer Vision*, 7(1):11–32, 1991.
6. T. F. Syeda-Mahmood. Data and model-driven selection using color regions. In *Int. J. Comp. Vision*, pages Vol. 21 No. 1. pp 9–36, 1997.
7. W Wang, Y Song, and A Zhang. Semantics retrieval by content and context of image regions. In *Proc. of the 15th International Conference on Vision Interface (VI'2002)*, Calgary, Canada, May 27-29, 2002.