Image Retrieval by Regions: Coarse Segmentation and Fine Color Description

Conference Paper · March 2002

DOI: 10.1007/3-540-45925-1_3 · Source: DBLP

2 authors:

Julien Fauqueur
DIGIT

Nozha Boujemaa
National Institute for Research in Computer Sci…

Image Retrieval by Regions: Coarse Segmentation and Fine Color Description

Julien Fauqueur[1] and Nozha Boujemaa[1]

[1] INRIA, Imedia Research Group, BP 105, F-78153 Le Chesnay, France
{Julien.Fauqueur,Nozha.Boujemaa}@inria.fr,
WWW home page: http://www-rocq.inria.fr/imedia/
2

Abstract. In Content-Based Image Retrieval systems, region-based queries allow more precise search than global ones. The user can retrieve similar regions of interest regardless their background in images.
The definition of regions in thousands of generic images is a difficult key point, since it should not need user interaction for each image, and nevertheless be as close as possible to regions of interest (to the user).
In this paper we first propose a new technique of unsupervised coarse detection of regions which improves their visual specificity. The Competitive Agglomeration (CA) classification algorithm, which has the advantage to automatically determine the optimal number of classes, is used.
The second key point is the region description which must be finer for regions than for images. We present a novel region descriptor of fine color variability: the Adaptive Distribution of Color Shades. It is based on color shades adaptively determined for each region at a high resolution: 5 million of potential different colors represented against few hundreds of predefined colors in existing descriptors.
Successful results of segmentation and region queries are presented on a database of 2500 generic images involving landscapes, people, objects, architecture, flora. ...

1 Introduction

The primary functionality of a Content-Based Image Retrieval system is the global query-by-example approach, in which visual features are extracted from the entire image. But in many cases the user's goal is to retrieve similar regions rather than similar images as a whole. In a generic image database the search for similar regions using global features over images can be highly biased by the surrounding regions and background.

Region based query systems allow to select a region in an image and retrieve images containing a similar region. The two major points to consider are the definition of regions and their description.

A manual extraction of regions was proposed in [1] but is unviable for huge databases. Automatic region detection can be performed on-line using features back projection (see [2] and [3]), but they are inaccurate and time consuming at query phase. Off-line methods include systematic image subdivision into squares (see [4]) and image segmentation. This latter method was proposed in a couple of systems such as Blobworld [5] and Netra [6]. In Blobworld [5], segmentation is performed by classification with EM-algorithm which requires to have a prefined number of classes. A contour-based segmentation proposed in [7] and integrated in a CBIR system ([6] and [8]) provides an accurate segmentation but with very homogeneous regions. We can also cite the work of Wang [9] in SIMPLICIty, which performs a color segmentation of images to describe an image as a set of regions, but single region queries can't be performed. Existing region color discriptors are based on histograms determined on a predefined subsampling of a color space: uniform subsampling of Hsv into 166 bins in VisualSeek [2], uniform subsampling of Lab into 218 bins in Blobworld [10] or a 256 color codebook predetermined for a given database in Netra [8].

Our approach differs from the above by our conception of regions and the techniques for extracting and describing them. We think regions should integrate more intrinsic variability to provide a better characterization and their color description should not depend on a predefined color set. The key idea of coarse region detection and fine description: the relatively high visual variability inside regions is accurately described by the fine resolution of color shades, such that regions are really specific against eachother in the database.

The Competitive Agglomeration classification algorithm used for both segmentation and indexing will be detailed in the first section. In section 3, the coarse image segmentation for automatic region detection will be presented. Region indexing and matching are explained in section 4. Then tests and results will be presented and discussed in section 6. Finally conclusions are drawn.

2 CA clustering Algorithm

Competitive Agglomeration classification, originally presented in [11], has the major advantage to determine the optimal number of clusters. In [12] an application of this algorithm is proposed for image segmentation.
Using notations from [11] and [12], we call $\{x_j, \forall j \in \{1, ..., N\}\}$ the set of N data we want to clusterize and C the number of clusters. $\{\beta_i, \forall i \in \{1, ..., C\}\}$ denotes the set of prototypes to be determined. The distance between data x_j and prototype β_i is $d(x_j, \beta_i)$. Then CA-Classification is performed by minimizing the following quantity J:

$$J = J_1 + \alpha J_2, \ where \ J_1 = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^2 d^2(x_j, \beta_i) \ and \ J_2 = -\sum_{i=1}^{C} [\sum_{j=1}^{N} u_{ij}]^2 \quad (1)$$

where u_{ij} represents the membership degree of feature point x_j to prototype β_i.

Minimizing J_1 separately is equivalent to perform an FCM classification [13] which determines C optimal prototypes and the fuzzy partition U given x_j and C using distance d. And J_2 is a complexity reduction term which garantees the cluster validity (see [12]). Therefore J is written as a combination of two opposite effect terms (J_1 and J_2). So minimizing J with an over-specified number of initial clusters simultaneously performs the data clustering and optimizes the number of clusters. α is the competition weight which should allow a balance between terms J_1 and J_2 in (1).

J is minimized recursively and at iteration k, weight α is written as :

$$\alpha(k) = \eta_0 \exp(\frac{-k}{\tau}) \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^2 d^2(x_j, \beta_i)}{\sum_{i=1}^{C} [\sum_{j=1}^{N} u_{ij}]^2} \tag{2}$$

As iterations go, α decreases so emphasis is first given to agglomeration process, then to classification optimization. α is fully determined by η_0 and τ.

During the algorithm spurious clusters are discarded. The convergence is decided when prototypes are stable.

The classification granularity is controlled by factor α through its magnitude η_0 and its decline strength with τ. The higher η_0 and τ, the higher α, so the more classes are merged. So for given classification granularity, CA determines the optimal number of classes. CA will be used at three steps in our work with different levels of granularity and different input data: first to perform image quantization, then to segment roughly the image by computing LDQC prototypes and then to finely describe regions with color shades.

3 Coarse region detection

Extracted regions should encompass a certain visual diversity to be visually characteristic, using a coarse segmentation. We want to stay beyond a too fine level of spatial and feature details. This choice is also motivated by the drawbacks of an oversegmentation which provides small and homogeneous regions:

- a small region is rarely visually salient in a scene
- a statistics-based description computed on a small region can't be accurate
- if all regions are homogeneous, it's harder to differentiate them from one another
- too many regions grow needlessly the database size

We'll define a *region of interest* as an area of connected pixels perceptually salient, i.e. covering a minimum surface in the image and presenting a certain visual "homogeneous diversity". To group pixels to form such regions, we want to perform a CA-classification of local color distributions of the image. This feature naturally integrates the diversity of colors in pixels neighbourhood.

The choice of the color set to compute local color distributions is crucial: it must be compact to gain speed in classification and be representative of a small pixel neighbourhood. If all original colors (an image can contain thousands of different colors) are kept, the classification will become computationally too expensive. Classic color histogram, computed on a uniform subsampling of a color space are too long (they contain useless empty bins). So we define the color set as the adaptive set representing the quantized colors of a given image obtained by color classification.

All neighbourhoods in the image give a set of Local Distributions of Quantized Colors (referred as *LDQC's*) which are classified. LDQC prototypes are back projected onto image, then small regions are either merged or discarded.

3.1 Image Color Quantization

Image colors are CA-classified as (L,u,v) triples using the Euclidean distance. The classification granularity was chosen such that big areas in images with a strong texture are at least represented by 2 color shades.

At classification convergence the color prototypes define the set C_{qc} of nqc color shades. Since CA determines automatically the right number of clusters, the number of color shades nqc will be representative of the image color diversity. Quantized image is obtained by back projecting color prototypes in the image.

3.2 Determination of LDQC prototypes in image

To determine all the LDQC's, we slide a window over pixels in the quantized image and evaluate the corresponding local distribution over the C_{qc} color set. Let's denote S_W the window surface and S_{TOT} the image surface. LDQC's are evaluated every wr pixels, where wr is the window radius, so that all pixels participate to the determination of the LDQC prototypes.

A suitable distribution distance must be used for the classification. L^p distances are widely used to measure similarity between color distributions computed over entire images but are not adapted to distributions computed over small pixel neighbourhoods. Indeed the distribution of a natural image is rather smooth and flatter than that of a small neighbourhood which presents a couple of peaks. Since there are few colors in a neighbourhood it is necessary to have a distance for LDQC which takes into account the inter-bin color similarity. This is what does the color quadratic form distance presented in [14]. Its expression is given for two distributions $\{x_i\}$ and $\{y_i\}$ evaluated on a set of nqc colors:

$$d_q(x,y)^2 = (x-y)^T A(x-y) = \sum_{i=1}^{nqc}\sum_{j=1}^{nqc}(x_i - y_i)(x_j - y_j)a_{ij} \tag{3}$$

where a_{ij} is the similarity between colors i and j, determined with the Euclidean distance in Luv space. This distance is used during classification to compare the LDQC histograms (we'll have $d = d_q$ in CA formulae (1) and (2)).

After classification, the segmented image is obtained by assigning to the S_{TOT}/wr^2 pixels the label of the LDQC prototype minimizing the quadratic distance to the LDQC around that pixel.
A maximum vote filter is applied to the image of labels to discard isolated pixels.

Window surface S_W defines the spatial level of details of the segmentation: the higher S_W and the bigger patterns we extract. wr was set to 8 pixels for a 500x500 image.

3.3 Adjacency information

The segmented image gives us a complete partition of the image into adjacent regions formed from the back projection of the LDQC prototypes. Very small regions correspond to salient areas detected by LDQC classification but are too small to constitute regions of interest, so they increase needlessly the total number of regions in the database. Besides, in complex scenes, they're often located at the frontier between two regions of interest or inside a region of interest. So they should be merged to improve the topology of regions of interest.

Region attributes (surface, color distribution) and region adjacency (list of neighbours) information are stored in a Region Adjacency Graph structure used to merge regions.

We want final regions of interest to be of minimum size $S_{Mmin} = 0.015 * S_{TOT}$ (i.e. 1.5% of the image surface). Below this threshold a region is merged to its closest visual neighbour if it has one and is discarded otherwise. Two small regions are said to be visually close if they have close mean quantized color distributions.

After merging process, remaining regions of size below S_{Mmin} are salient but too small, so they are discarded from the graph and not indexed.

The region extraction workflow is the following:

1. image quantization by CA-classification of color pixels
2. computation and CA-classification of LDQC's to obtain LDQC prototypes
3. determination of connected components and generation of the RAG
4. merge and discard regions

4 Region Indexing and retrieval

4.1 Fine color region description

Once regions are detected in a coarse way we have to finely describe their visual appearance. Existing region color descriptors are generally histograms evaluated on a few hundreds of bins obtained by a subsampling of the color space: a uniform subsampling in [2], [10] or a database-dependent subsampling in [8]. See the illustration of a 216 bin Luv histogram region description in left part of figures (1) and (2). Such a description forces the minimum distance between two colors to be high because the subsampling is fixed and because we only consider a few hundreds of colors among millions in a full color space. This low granularity of color description is suitable for complex *images* as they contain a wide range of different colors.

But regions are by definition more homogeneous than an image so their color description should be finer. To represent shades of any given hue, a high granularity color set must be found. A fine uniform subsampling of a color space raises the problems of numerous useless empty bins and heavy matching computation.

We want to select for each region an adaptive color set providing color shades which are relevant for the region. We should get a single color shade on a perfectly uniform region and many on a highly textured region. We decide to index regions with the distribution of their color shades determined with CA algorithm with a high classification granularity.

To achieve this, for *each region*, its color pixels in the *original image* are classified with low τ and η_0 to catch representative shades of colors. The optimal number of color shades found by CA is in itself an information about the region visual diversity. The color shade triples are determined from the whole Luv colorspace which contains 5.6 million colors while a classic color descriptor picks colors from around 200 given colors.

The descriptor index consists of the list of color shades as Luv triples with their respective percentage in the region. Top-right parts of figures (1) and (2) show examples of such descriptors.

Note: the image quantized colors determined in section 3.1 are unsatisfactory candidates to index regions for two reasons: they are determined with a too low granularity (suitable for a coarse segmentation) and all image color pixels are in competition which favours colors from big regions and bias the color prototypes determination.

4.2 Matching regions

For a given query region of color shades distribution X, similar regions are such that their distribution Y minimizes the distance between X and Y. Let's write distributions X and Y as pairs of color/percentage:

$$X = \{(c_1^X, p_1^X), ..., (c_{ncs_X}^X, p_{ncs_X}^X)\} \; and \; Y = \{(c_1^Y, p_1^Y), ..., (c_{ncs_Y}^Y, p_{ncs_Y}^Y)\} \tag{4}$$

and $a_{c_i^X c_j^Y}$ as the color similarity between c_i^X (i^{th} color of X) and c_j^Y (j^{th} color of Y).

Since color shades are finely determined the quadratic distance is again a good choice to take into account the inter-bin color similarity. The formula (5) gives the quadratic distance between two color distributions x and y evaluated on the *same* color set. But when measuring the distribution distance between two regions from two different images, the two color sets are different. So we will rewrite the expression of the quadratic distance to discard the distributions binwise differences.

Let's consider x as the extension of distribution X over the entire color space and y the extension of Y. The extension consists in setting bin values to zero for colors which are not color shades, so we have $d_q(x, y) = d_q(X, Y)$.

$$\begin{aligned} d_q(x,y)^2 =& (x-y)^T A (x-y) \\ =& x^T A x - x^T A y - y^T A x + y^T A y \\ =& x^T A x + y^T A y - 2 x^T A y \\ =& \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j a_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j a_{ij} - 2\sum_{i=1}^{N}\sum_{j=1}^{N} x_i y_j a_{ij} \end{aligned} \tag{5}$$

Then we finally have the following expression of the quadratic distance used to compare two color shades distributions X and Y evaluated on any color sets :

$$d_q(X,Y)^2 = \sum_{i=1}^{ncs_X}\sum_{j=1}^{ncs_X} p_i^X p_j^X a_{c_i^X c_j^X} + \sum_{i=1}^{ncs_Y}\sum_{j=1}^{ncs_Y} p_i^Y p_j^Y a_{c_i^Y c_j^Y} - 2\sum_{i=1}^{ncs_X}\sum_{j=1}^{ncs_Y} p_i^X p_j^Y a_{c_i^X c_j^Y} \tag{6}$$

The first term involves only the X distribution, the second the Y distribution and the last one the product of both and no more binwise difference is involved. Returned regions are sorted by growing quadratic distance d_q.

5 Tests

Our system was tested on IDS database provided by courtesy of *Images Du Sud Photo Stock* company. It contains 2500 generic images of flowers, portraits, landscapes, seascapes, architecture, people, fruit, gardens. Images size are between 400x400 and 600x600 pixels.

6 Results

6.1 Region Detection

A few segmented images are presented in figure (3). More examples can be seen at: `http://www-rocq.inria.fr/~fauqueur/ADCS/` .

Images for which an obvious segmentation could be decided are correctly segmented. More generally, images in the database are complex natural scenes and extracted regions present a coherent color diversity. The coarse segmentation proves its ability to integrate within regions areas formed with many shades of the same hue, strong textures, isolated spatial details, which make their specificity.

15248 regions were automatically extracted from the 2483 images (average of 6 regions per image). Segmenting an image took an average of 5.6s. Discarded regions (shown as small grey regions in examples) represent a very small percentage of image surfaces.

To compare color shades, a 4*4*4=216 uniform subsampling of the Luv colorspace was also tested to compute the local color distributions. Resulting regions were inaccurate and the histogram vectors were to long to classify.

6.2 Region Description

Top-right parts of figures (1) and (2) illustrate the fine granularity of the color shade representation and their fidelity to the original colors. In figure (3), the segmented images show the detected regions followed by the corresponding images formed by each region color shades used for their description. More examples of such images can be seen at:

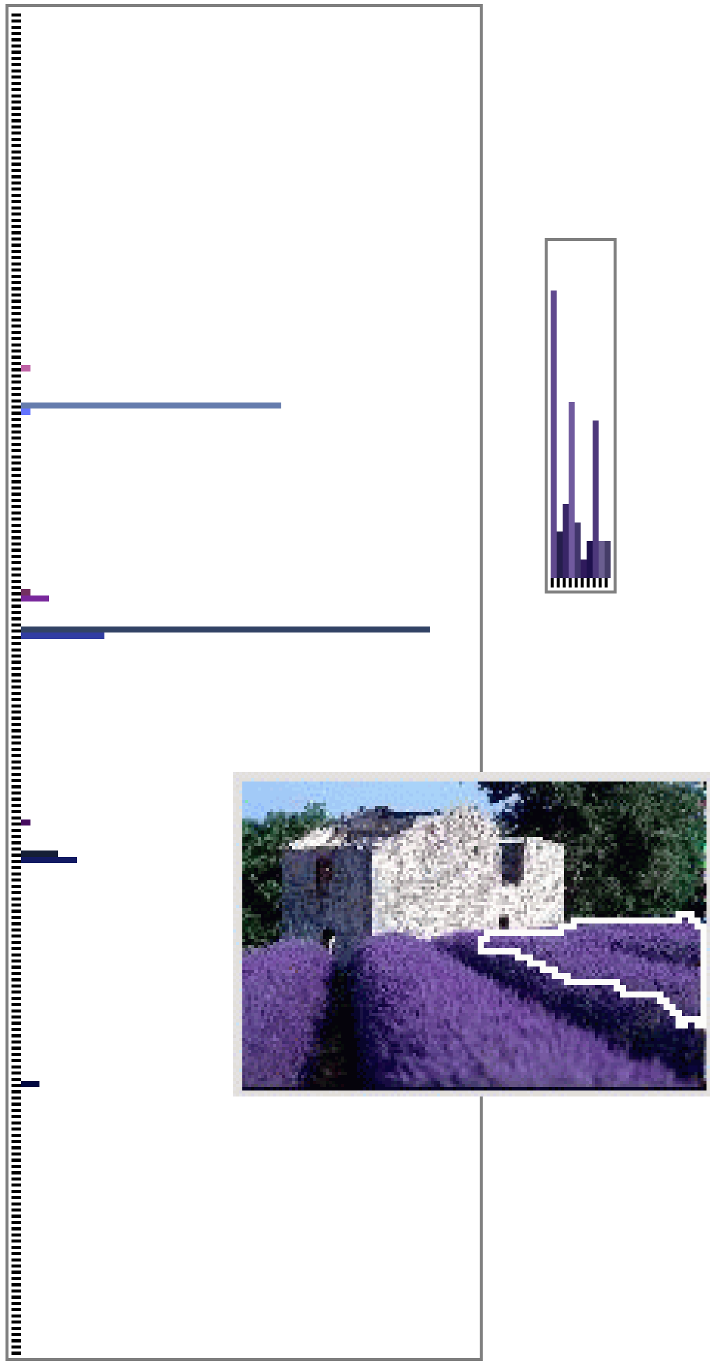

Fig. 1. Color description of the lavender region: with a classic 216 bin Luv distribution (left) and with the ADCS descriptor (top right). Because of the strong subsampling into 216 bins, wrong colors appear in the classic Luv distribution: blue shades rather than purple ones. The ADCS descriptor represents the purple color shades accurately and provides a more compact descriptor.
Note color bins in the ADCS distribution have no specific order.

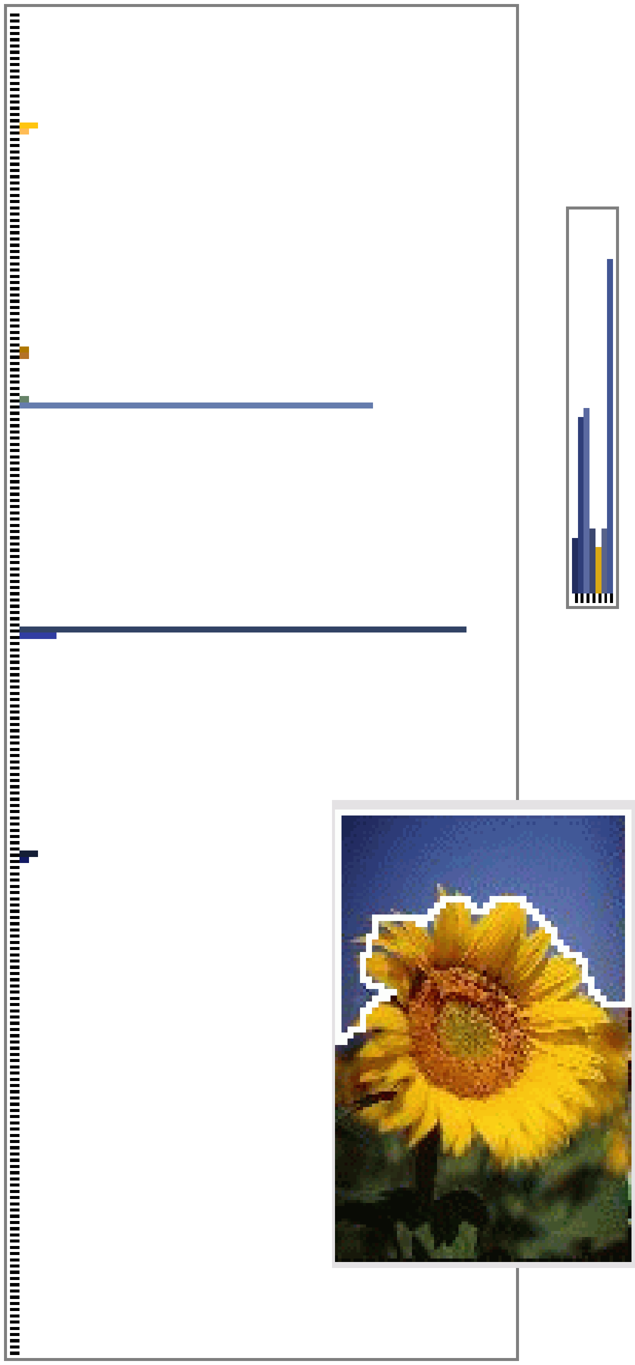

Fig. 2. Color description of the sky region: with a classic 216 bin Luv distribution (left) and with the ADCS descriptor (top right). Distribution comparison: both distributions represent real colors but the ADCS has a finer dynamic of blue shades and still in a more compact descriptor.

http://www-rocq.inria.fr/~fauqueur/ADCS/ .

The global appearance of these quantized images shows the precision of the ADCS region color descriptor.

A total of 261219 color shades from the Luv space were used to index the 15248 regions (average of 17 colors per region). 168912 of these colors were unique (to be compared to the couple of hundreds of fixed bins in a classic histogram). Extracting an ADCS index from a region took around 0.5s .

Since an average of 17 colors is used to represent a region, we can determine the number of bytes needed to store an ADCS index: for one region, it contains: the number of color shades, the list of color shades (as Luv triples) and the population of each shade, i.e. $1 + 17 * (3 + 1) = 69$ bytes. This makes an ADCS index around three times more compact than a classic color histogram.

Fig. 3. First: original images. Second: images of regions with mean color. Third: mages of regions with color shades used for indexing. Non-indexed regions are shown with random color pixels.

6.3 Retrieval

Region queries are done by exhaustive comparison with the 15248 regions and average query time is 1.3s.

Hundreds of region queries in our system always returned regions which presented a perceptually similar color distribution for various kinds of regions: uniform or textured, containing different hues. Regions described by many color shades returned regions with many color shades and conversely for single-colored regions. We observed that the number of color shades is also an exploited information about the color diversity of a region.

Screenshots in figures (4) and (5) show the result of a query on a lavender region. ADCS descriptor is used in figure (4) and, in figure (5), classic 216 bin Luv histogram matched with the L^1 distance. We can observe that classic histogram didn't top-ranked regions with colors as similar as with color shades.

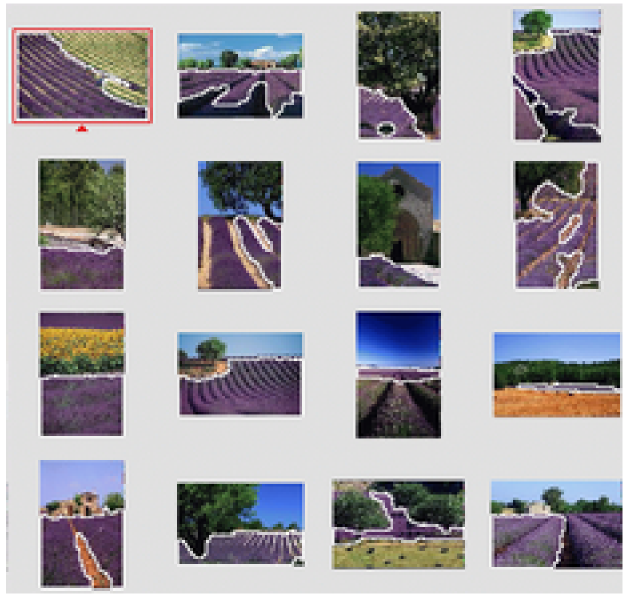

Fig. 4. Retrieval from top-left lavender region using ADCS.

7 Conclusions

The key idea is to detect visually specific regions of interest and match them with the fine descriptor to improve the retrieval results. We presented a scheme for coarse automatic image segmentation and fine color description to perform region-based queries in a generic image database.

The novel segmentation scheme detects regions which are potential regions of interest for the user (they are visually salient in the image) and at the same time specific from one another in the database (they encompass a visual "homogeneous diversity").

The new ADCS signature provides a representation of region color variability with more accuracy than existing descriptors.

References

[1] Del Bimbo and Vicario E., "Using weighted spatial relationships in retrieval by visual contents," *IEEE workshop on Image and Video Libraries*, June 1998.

[2] S.F. Chang J.R. Smith, "Visualseek: A fully automated content-based image query system," in *ACM Multimedia*, 1996, pp. 87–98.

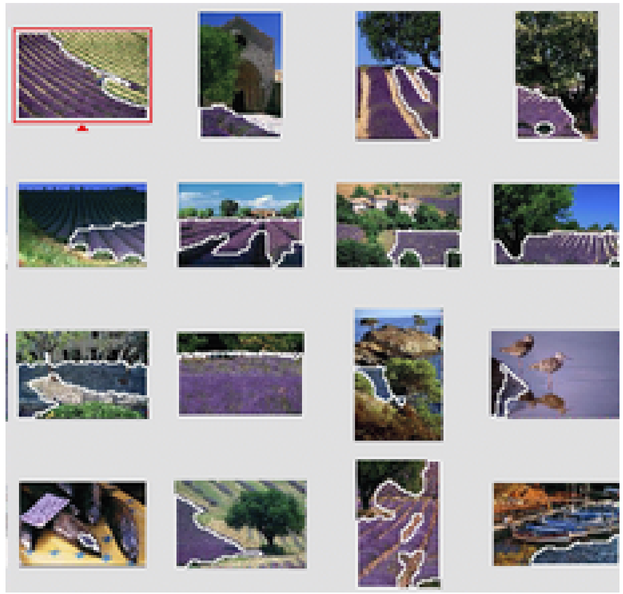

Fig. 5. Retrieval from top-left lavender region using classic 216 bin Luv histogram.

[3] B. Moghaddam, H. Biermann, and D. Margaritis, "Defining image content with multiple regions of interest," *CBAIVL*, 1999.

[4] J. Malki, N. Boujemaa, C. Nastar, and A. Winter, "Region queries without segmentation for image retrieval by content," in *Visual Information and Information Systems*, 1999, pp. 115–122.

[5] Belongie S., Carson C., Greenspan H., and Malik J., "Color- and texture-based image segmentation using em and its application to content-based image retrieval," *Proc. Int. Conf. on Computer Vision (ICCV'98)*, 1998.

[6] Deng Y. and Manjunath B., "An efficient low-dimensional color indexing scheme for region-based image retrieval," *ICASSP Proceedings*, 1999.

[7] Ma W. and B. Manjunath, "Edgeflow: A framework of boundary detection and image segmentation," *CVPR Proceedings*, pp. 744–749, 1997.

[8] Wei-Ying Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," *Multimedia Systems*, vol. 7, no. 3, pp. 184–198, 1999.

[9] Jia Li James Z. Wang and Gio Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *PAMI*, 2001.

[10] C. Carson, M. Thomas, and S. Belongie, "Blobworld: A system for region-based image indexing and retrieval," 1999.

[11] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, vol. 30, no. 7, pp. 1109–1119, 1997.

[12] Boujemaa N., "On competitive unsupervized clustering," *ICPR*, 2000.

[13] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Functions*, Plenum, New York NY, 1981.

[14] J. Hafner H. Sawhney W. Aquitz M.Flickner and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *PAMI*, 1995.