Computers and Mathematics with Applications 63 (2012) 511–517

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Efficient object identification and localization for image retrieval using query-by-region[☆]

Yong-Hwan Lee[a], Bonam Kim[b,*], Heung-Jun Kim[c]

[a] Dankook University, Yongin, Republic of Korea
[b] Chungnam National University, Daejon 448-701, Republic of Korea
[c] Gyeongnam National University of Science and Technology, JinJu, Republic of Korea

ARTICLE INFO

Keywords:
Object localization
Correlogram back-projection
Object region detection
Query-by-subregion
Region-based image retrieval

ABSTRACT

Localizing an object within an image is a common task in the field of computer vision, and represents the first step towards the solution of the recognition problem. This paper presents an efficient approach to object localization for image retrieval using query-by-region. The new algorithm utilizes correlogram back-projection in the YCbCr chromaticity components to handle the problem of subregion querying. Utilizing similar spatial color information enables users to detect and locate primary location and candidate regions accurately without the need for further information about the number of objects. Comparing this new approach to existing methods, an improvement of 21% was observed in experimental trials. These results reveal that color correlograms are markedly more effective than color histograms for this task.

1. Introduction

Content-based image retrieval (CBIR) is the application of computer vision techniques to the image retrieval problem, specifically the search for specific digital images in large databases [1]. The two approaches commonly used for image retrieval are referred to simply as global-based image searches and region (or sub-image)-based image searches [2]. An important distinction between these approaches is that global-based methods enable whole image matching and consider how much of an image is relevant, while region-based methods focus primarily on specifying a region and on retrieving a large number of images with similar objects. Both methods are useful for image retrieval, but are best suited to queries of different types. Searching by global distinction is the preferred approach in cases where the user provides a whole image for query (i.e. query-by-example) [3], where queries take the form of "*show me more relevant images that look like this query image*". For instance, if a user is interested in finding panoramic shots of a soccer game, then relevant images can be retrieved from an image database by matching a user's input query image. In this case, global-based search methods work well because the user is not concerned with the precise position of colored objects or regions within the images. However, if the user is interested in finding something located in a specific part of an image (e.g., "*show me relevant images with a red flower on the right*"), global-based retrieval is unable to resolve spatially localized color regions from the global distribution and region-based image searches will be more successful. For both these techniques, the retrieval system must incorporate a function capable of performing the automated extraction and efficient representation of visual features.

There are two different kinds of tasks involved in this process: object-presence detection and object localization [4]. Object-presence detection seeks to determine whether one or more objects are present anywhere in the image. This is

[☆] A short version of this paper appeared at the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.
[*] Corresponding author. Tel.: +82 10 3213 1715.
E-mail address: kimbona@cnu.ac.kr (B. Kim).

0898-1221/$ – see front matter Crown Copyright © 2011 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.08.019

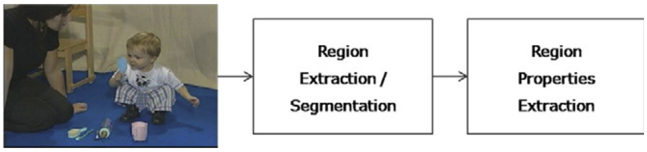


Fig. 1. General approach for image subregion querying.

called *image classification*, and is most useful for object-based image retrieval. Formally, it can be described as estimating the probability $P(O = 1|I)$, where $O = 1$ represents the presence of a particular object and I is the image. In contrast, object localization detects instances of a given category in the image, in many cases up to a bounding box. Formally, this is represented as estimating the probability $P(X = i|I)$, where $i \in \{1, \ldots, N\}$ is a set of possible occurrences of the object in the image. Once it has been determined that an image contains an object (e.g., a red flower), it becomes necessary to locate that object inside the image. This is referred to as the *localization problem* [4], and is defined as: for a given image M (dubbed the model, to distinguish it from a whole image query) and image I (the target image) such that $M \in I$, find the location in image I where query image M is present. The localization problem can be viewed as a special case of the image retrieval problem in the following manner. Let $I|_p$ denote the sub-image in I located at a position with the same size as M. The set of all sub-images $I|_p, \ldots, I|_I$ present in I constitutes the image database and M is the query image. The solution $I|_p$ to this retrieval problem gives p, the location of M in I.

In this paper, we examine how the back-projection method can be utilized to solve the localization problem for subregion image queries, and apply the block-oriented decomposition technique to the selection of the subregion. We then propose a simple but efficient algorithm for identifying the location of a known object in an image, where the object refers to the query image and corresponds to the model in the description of the localization problem. The new method is able to find not only the best matching region but also the candidate target region without the need for additional information about the number of objects in an image. This paper is an extension of the work first presented in [5]: here we provide a more thorough experimental comparison, and demonstrate much improved performance.[1]

The remainder of this paper is organized as follows. In Section 2, we present related works and background. In Section 3, we explain the proposed template matching method, which uses correlogram back-projection to detect the occurrence of the target object in an image. In Section 4, we present and discuss experimental results, and we conclude in Section 5 with a summary and suggestions for future work.

2. Related works and theoretical background

Before we introduce our proposed new method, it is useful to briefly review related work and the theoretical background on which this work is based. Image retrieval based on subregion querying is technically more difficult than approaches that utilize whole image querying. The process consists of two stages, the first of which is region extraction and segmentation, and the second, the extraction of region properties [6], as shown in Fig. 1.

In this paper, we focus on region extraction in order to address the localization problem and deal primarily with spatial color histograms, thus avoiding the intractable problem of region segmentation.

The main goal of region extraction is to identify the spatial boundaries of those regions that would be of most interest to a user, or would occupy an area within the image similar to that occupied by the smaller query image. There are several techniques that can be used for region extraction: (1) manual (or semi-automated) extraction, (2) fixed block segmentation, (3) color segmentation and (4) template matching [7]. The first of these, the manual extraction method, is not relevant for this study since the objective is to use an automated system to perform the extraction of the region of interest. The second approach utilizes a fixed block segmentation of the image, such as the region tessellation used in MPEG-7 [8]. Representing the feature content of small blocks independently increases the likelihood of obtaining matches between regions, although it may be difficult to select the optimum scale for the image block sizes. The third approach involves color segmentation and several papers have proposed algorithms based on the segmentation technique [9]. Image segmentation involves a complete partitioning of the image such that each image point is assigned to a single segment. This differs from region extraction, where an image point may be assigned to many regions or none. Conceptually, this is more suited to template matching, the fourth technique, than image segmentation because it supports an object-oriented representation of image content. The template matching process takes the small query image as a template, and moves this template over all the possible locations in the target image to find the best match.

Recent research on moving video, where moving objects are detected and tracked, has established a number of techniques [10] and several algorithms have been proposed for finding the location of an object in an image. For example,

[1] These improvements are due to two changes: first, a better algorithm is applied in the pre-processing stage to enhance time and memory efficiency; and second, image databases were prepared more carefully, ensuring sufficient objects in an image and eliminating errors in image collection by removing images without the object.

Lampert et al. in [11] proposed the use of an efficient sub-window search (ESS) algorithm based on the branch-and-bound procedure. They reported that ESS identified the global maximum of boundaries with a worst-case computational complexity of $O(n^2)$ for $n \times n$ images. An alternative algorithm suggested by Malki et al. in [12] adopted a region querying approach based on the use of a multi-resolution quad tree representation of the images. However, this cannot be directly compared to the algorithm proposed in the current paper, because they utilized non-overlapping fixed blocks with structured regions. Other researchers have used color templates and histograms for localizing objects [13,14]. Histogram based approaches can be useful for abstract representations of objects that are flexible enough to handle scaling and changes in viewpoint, but there is often significant information contained in the distribution of the color through the background. Thus, they tend to be too specific and can seldom adapt to small object shapes. In such cases, color co-occurrence histograms or correlograms offer a possible solution to the problem.

The following paragraphs explain the conventional features and methods used to address these problems, namely the histogram, the correlogram and template matching. A histogram is a graphical display of frequencies that represents the total distribution in an image. This is formulated for color c_i in the image I as

$$H_{c_i}(I) = \text{prob}_{p \in I}[p \in I_{c_i}]. \tag{1}$$

Since histograms correspond to the probability of any pixels of color c_i occurring in an image, this feature does not take into account the spatial distribution of color across different areas of the image. A correlogram characterizes not only the color distribution of pixels but also the spatial correlation of pairs of colors (c_i, c_k). Thus a correlogram gives the probability of finding a pixel p_2 of special color c_k at a distance d for a pixel p_1 of given color c_i. This is formulated for color pair (c_i, c_k) as

$$C^d_{c_i,c_k}(I) = \text{prob}_{p_1 \in I_{c_i}, p_2 \in I}[p_2 \in I_{c_k} || p_1 - p_2| = d]. \tag{2}$$

With all the possible combinations of color pairs, the size of the correlogram is likely to be very large, so a formulation known as an *autocorrelogram* is generally used. An autocorrelogram provides the probability of capturing spatial correlations between identical colors only. The autocorrelogram is consequently a simplified subset of a correlogram and takes $C^d_c(I) = C^d_{c_i,c_k}(I)$. The autocorrelogram is then formulated as

$$C^d_c(I) = \text{prob}_{p_1 \in I_c, p_2 \in I}[p_2 \in I_c || p_1 - p_2| = d]. \tag{3}$$

A simple and effective way to implement template matching is to utilize histogram back-projection, first proposed by Swain and Ballard [15]. Here, the most likely location of a spatially localized color histogram within an image is found by back-projection onto the image of the quotient of the query histogram and the image histogram [16]. The histogram back-projection method can be described as follows. Given a query image M and an image I in the database, the ratio histogram R_h is defined as $R_h(m) = \min\left(\frac{H_M(m)}{H_I(m)}, 1\right)$, where m is the histogram bin, and H_M and H_I are the histograms of the model M and the image I. Each point in the image is assigned a likelihood $B[m, n] = R_h(I[m, n])$. A blurring mask D is generated with the disk radius r from Eq. (4):

$$D^r_{x,y} = \begin{cases} 1, & \text{if } \sqrt{x^2 + y^2 \le r} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Image B' is then blurred with mask D^r as $B'[x, y] = D^r[x, y] * B[x, y]$. The symbol $*$ denotes the two-dimensional convolution. Finally, the location of the model is given by $(x_p, y_p) = \max(B'[x, y])$. The point (x_p, y_p) is the location of the peak and its value corresponds to the most likely location of the model histogram within the image.

However, there is a major drawback in that the back-projection only compares like bins, and thus incorporates no spatial information. Therefore, false matches are likely to occur, especially when there are multiple similarly colored objects. In other words, histogram back-projection is insensitive to changes of image resolution or histogram resolution, and its complexity is given by $O(P_i + P_d \times P_b)$, where P_i, P_b and P_d are the number of pixels in the image I, the blurred image B and the convolution mask D, respectively.

3. The proposed algorithm

Since the histogram back-projection method has a weak point in its correlation, we extended the method to incorporate spatial information by applying a color correlogram to overcome this problem. Fig. 2 depicts the proposed subregion querying routine.

The process of the proposed method can be described as follows. Given the query image (model) M and the images I in the database, image pre-processing is performed in the first step. This includes color space conversion, separation into three components and bi-directional down-sampling. The model and the images in the database must be preprocessed by converting the color space into YCbCr, which is widely used in both images (e.g., JPEGs) and digital video. However, since the correlogram depends primarily on the computation of the spatial correlation between different colors presented in the image, and better performance can be achieved with more spatial information, this is liable to lead to problems with

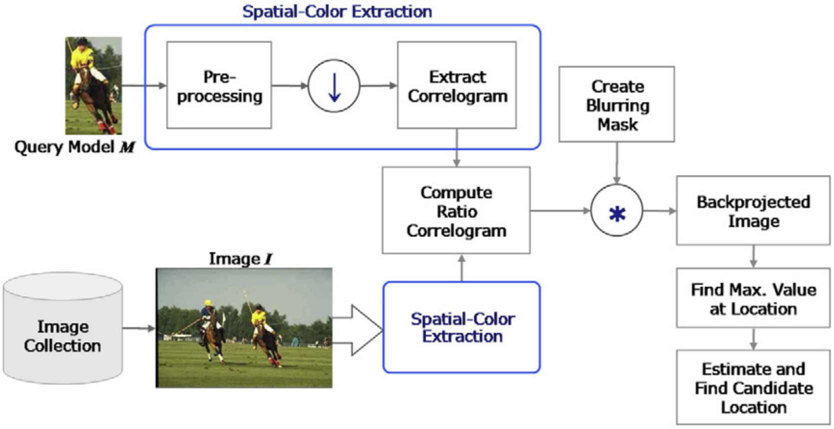
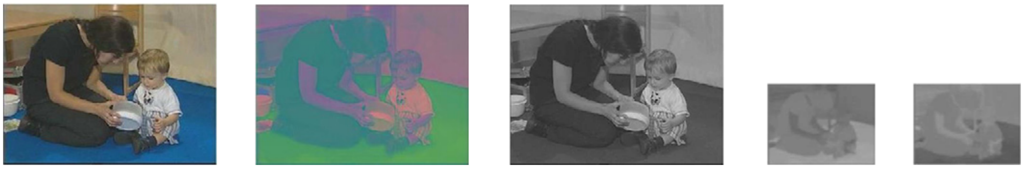


Fig. 2. Diagram of the proposed subregion querying method.



(a) Example image in RGB color space.

(b) Image in YCbCr.

(c) Y component.

(d) Cb component with 4:1:1.

(e) Cr component with 4:1:1.

Fig. 3. Example image and its converted YCbCr images; (a) the example image shown in RGB, (b) corresponding to the YCbCr image, (c) the Y component, and ((d), (e)) the Cb and Cr components with 4:1:1 down-sampling.

higher computational costs. To avoid this, the new model utilizes 4:1:1 YCbCr down-sampling, implementing horizontal and vertical sub-sampling for chrominance planes. Fig. 3 shows an example of the same image in RGB and YCbCr color spaces.

Then, the correlograms of the image and the model are calculated using Eq. (6), and the correlogram ratio R_c is computed using Eq. (7).

$$C_I^d(c_i) = \frac{|\{p(x,y)|I(x,y)=c_i;\ I(x \pm d, y \pm d) = c_i\}|}{|\{p(x,y)|I(x,y)=c_i\}|} \tag{5}$$

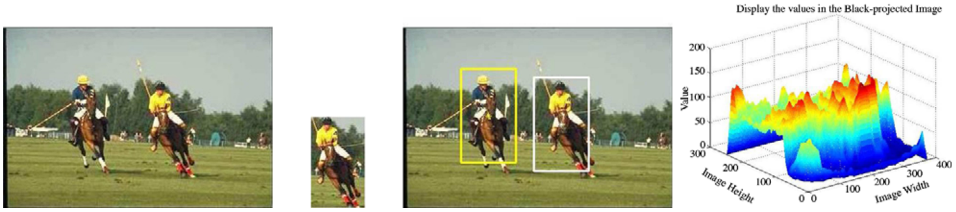
where c_i is the distinct value of the color and d is the fixed distance of correlation. The correlogram of image C_I^d which comprises pixels $p(x, y)$ is re-formulated from the definition of Eq. (3).

$$R_c(m) = \min\left(\frac{C_M^d(m)}{C_I^d(m)}, 1\right) \tag{6}$$

where m is the correlogram of the quantized value, and C_M^d and C_I^d are the correlograms of model M and image I with distance d, respectively. In Eq. (6), the value R_C indicates the probability that an image pixel with color j belongs to an occurrence between distances of the model in the image.

We then place the image color at every pixel, along with its probability of being part of the model, thus forming the back-projection image B_c as $B_c[m, n] = R_c(I[m, n])$. A blurring mask D_c is also generated, as shown previously in Eq. (4). Finally, the location of the maximum value in the back-projected image B_c' is identified after it has been convolved with mask D_c. This convolution sums the confidence values over local areas, and the location of the maximum in the result is the place where the model has detected the target object in the image. The algorithm used to select the location of the maximum value and the candidate location with the threshold in the back-projected image is as follows.

```
function $pLocs$ = FetchPeakLoc($imImage$) {
  $imSize$ = GetImageSize($imImage$);
  $imImageSeq$ = Generate1DSequence($imImage$);
  $locMax$ = Find($inImageSeq$ == Max($inImageSeq$));
  $peakV$ = $locMax - 0.1$;
  $peakLoc$ = [Floor($peakV/imSize(1) + 1$, Ceil(Mod($peakV$, $imSize(2)$))];
  $pLocs$ = AddPoint($peakLoc$);
```



(a) Image. (b) Model. (c) Result view.

Fig. 4. Example of successful matching in region extraction with the proposed method; (a) target image, (b) query model, (c) result of finding the object location, and (d) three-dimensional view of the peak values corresponding to the back-projected image shown in Fig. 4(c).

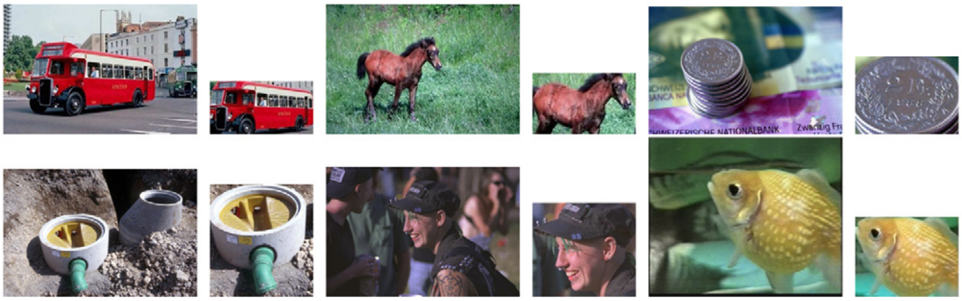


Fig. 5. Sample pairs of images and corresponding query models.

```
    if (peakLoc − anyLocations) < threshold
        then AddPoint(anyLocations);
    return pLocs;
}
```

The returned structured values (pLocs) are the positions where the model occurs in the whole image. This algorithm speeds up the computation by using a 1D array instead of a 2D image, reducing the complexity to $O(n)$ from $O(n^2)$.

Fig. 4 shows an example of finding the object location with the proposed algorithm. In the Fig. 4(c), the right white rectangle indicates the exact location with the highest peak value, and the left yellow one presents the candidate location with the second-highest peak value.

To evaluate the performance of the proposed algorithm, let variable $\mathrm{loc}(M, I)$ be 1 if the returned location is within reliable tolerance of the actual location of model M in the image I, and the variable be 0 if the location is outside the tolerance. A successful matching threshold is assigned to the center point of the model within a quarter of the centered model size. Then, given a series of queries $M_1, M_2, \dots, M_n$ and corresponding images $I_1, I_2, \dots, I_n$ (where n is the number of images in the database), the success ratio of the proposed method is given by Eq. (7).

$$\text{success ratio} = \frac{\sum_{i=1}^{n} \mathrm{loc}(M_i, I_i)}{n}. \tag{7}$$

4. Experiments and results

To evaluate the performance of the proposed object location scheme, we selected three datasets of images. Two of the datasets were Corel photo gallery[2] and MPEG-7 CCD (common color datasets[3]), both of which are widely used in the field of image retrieval. The third set included natural photos obtained from the website www.freeimages.co.uk. Each collection has images with various resolutions (e.g., 384 × 256, 640 × 420, 768 × 512 and 1600 × 1200) and various kinds of images, including humans, flowers, buses, fruits, structures, materials, and so on. For the location problem, query models and images containing any objects in the image were chosen. Images containing non-objects (such as water) were removed from the dataset, and the final database consisted of 2200 images. To identify the model location, each of the images was manually segmented to derive ground truth sets. Fig. 5 shows sample pairs of images and their corresponding query models.

Table 1 compares the performance achieved through changing the variables of (1) the down-sampling rate in YCbCr and (2) the correlogram distance (variable d in Eq. (5)). The result of 4:4:4 down-sampling slightly improves the success ratio, by

[2] http://wang.ist.psu.edu/docs/related/.

[3] http://abacus.ee.city.edu.hk/mpeg7/.

Table 1
Comparison of the success ratio and average processing time according to the variable condition.

Experimental parameters		Success ratio (%)	APT[a]
Down-sampling rate	4:1:1	86.73	3.110
	4:2:2	86.77	3.614
	4:4:4	87.05	4.668
Correlogram distance	1	86.73	3.110
	3	86.91	3.548
	5	87.00	4.048
	7	87.09	4.814

[a] APT stands for average processing time per image (s/image).

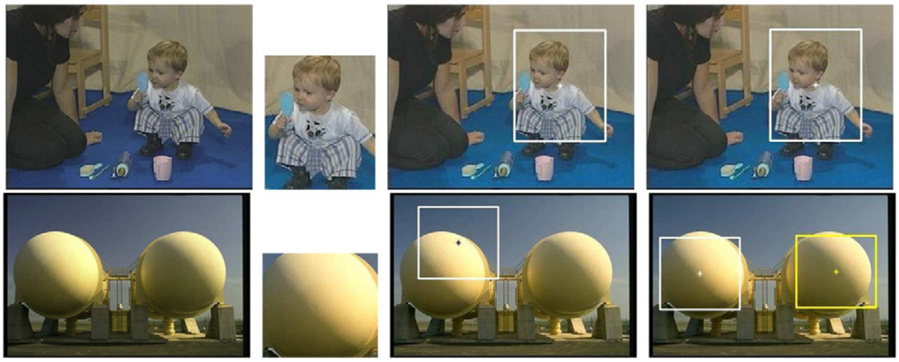


Fig. 6. Example of successful matching results of subregion querying: (a) target images, (b) models, (c) results of the histogram back-projection, and (d) results of the proposed method.

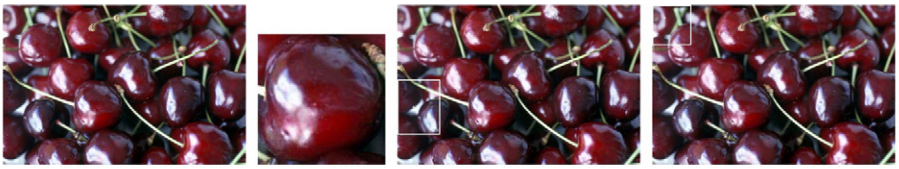


Fig. 7. Example of subregion querying with false matching.

Table 2
Results for subregion querying.

Method	Success ratio	APT[a]
Histogram back-projection [14]	1453/2200 (66%)	2.297
Proposed method	1908/2200 (87%)	3.110

[a] APT stands for average processing time per image (s/image).

0.32%, compared to that for 4:1:1 sampling (corresponding to detecting 7 additional images over the entire dataset of 2200 images). However it spent considerably more processing time—around 50% more (increasing from 3.110 to 4.668 s). The success ratio is also increased slightly with larger distance value (from 1 to 7), although this again increased the processing time by over 50%. On the basis of these results, it was determined that 4:1:1 down-sampling and correlation distance $d = 1$ are best suited to the task of object detection.

Both the standard histogram back-projection and the new method proposed here based on the use of correlograms were evaluated. Figs. 6 and 7 show examples of subregion querying with successful matching and with false matching, respectively.

As the second example in Fig. 6 shows, two locations that appeared to offer suitable matches for the primary location and candidate region were found with the proposed method, compared to the single location found for the first example. The unreliable determination of the localization was also observed with the new method, as shown in Fig. 7, probably due to the continuous occurrence of similar patterns with the object. As Fig. 8 shows, it is not always possible to find the location of values in the back-projected image and identify the maximum value because of blurring with similar objects in the same image.

Table 2 shows the results for the 2200 queries. Experiments were conducted on a 2.66 GHz Intel Core2 Duo CPU PC with 2 GB of RAM running Windows XP Professional OS. The algorithms were implemented in Matlab 7.0.1.

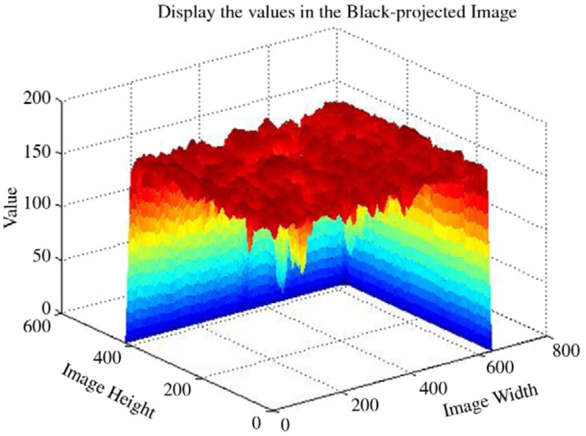


Fig. 8. Three-dimensional view of the peak values corresponding to the bottom image of Fig. 8.

The experimental results show that the proposed method achieved a success ratio of over 87%, markedly better than the 66% achieved by the histogram back-projection method for identifying the location of the query model in the image, an improvement of 21% over the comparative approach. Furthermore, our proposed new approach benefits from a noticeable reduction (0.814 s) in the average processing time compared to the results reported previously [5].

5. Conclusion

In this paper, we propose an efficient block-oriented detection algorithm based on the use of correlogram back-projection to solve the image subregion querying and object localization problem. The proposed approach is capable of identifying and locating objects in the primary region, as well as the candidate region, with no information about the object count. As shown by the experimental results, the proposed method significantly improved the performance when finding the location of a subregion in a whole image. Furthermore, in order to overcome the computational time problems afflicting the previous study, 4:1:1 YCbCr components were adapted to reduce the number of pixels in the image and the model.

This study builds on recent research on spatially localized features and subregion-based image retrieval. The key contribution of this paper is that a different way of treating color spaces and a histogram measure, which involves information on spatial color, are applied in object localization. This approach opens up new opportunities for improving the performance of sub-image retrieval.

References

[1] A.H. Halawani, A. Teynor, L. Setia, G. Brunner, H. Burkhardt, Fundamentals and applications of image retrieval: an overview, Datenbank-Spektrum 18 (2006) 14–23.

[2] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences, and trends of the new age, ACM Computing Surveys 40 (2) (2008) 1–60.

[3] M.S. Lew, N. Sebe, C. Djeraba, R. Jain, Content-based multimedia information retrieval: state of the art and challenges, ACM Transactions on Multimedia Computing, Communications, and Applications (2006) 1–19.

[4] K. Murphy, A. Torralba, D. Eaton, W. Freeman, Object detection and location using local and global features, Lecture Notes in Computer Science 4170 (2006) 382–400.

[5] Y.-H. Lee, B. Kim, H.-J. Kim, Efficient object localization for query-by-subregion, in: International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011.

[6] J. Huang, S.R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Spatial color indexing and applications, International Journal of Computer Vision 35 (3) (1999) 245–268.

[7] J.R. Smith, Integrated spatial and feature image systems: retrieval, analysis and compression, Ph.D. Thesis, Columbia University, USA, 1997.

[8] M.-S. Ryu, S.-J. Park, C.S. Won, Image retrieval using sub-image matching in photo using MPEG-7 descriptors, Lecture Notes in Computer Science 3689 (2005) 366–373.

[9] K.S. Deshmukh, G.N. Shinde, An adaptive color image segmentation, Electronic Letters on Computer Vision & Image Analysis 5 (4) (2005) 12–23.

[10] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38 (4) (2006) 1–45.

[11] C.H. Lampert, M.B. Blaschko, T. Hofmann, Beyond sliding windows: object localization by efficient subwindow search, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[12] J. Malki, N. Boujemaa, C. Naster, A. Winter, Region queries without segmentation for image retrieval by content, Lecture Notes in Computer Science 1614 (1999) 115–122.

[13] B. Deutsch, C. Grabl, F. Bajramovic, J. Denzler, A comparative evaluation of template and histogram based 2D tracking algorithms, in: German Pattern Recognition Symposium, 2005.

[14] M. Wirth, R. Zaremba, Flame region detection based on histogram backprojection, in: Canadian Conference Computer and Robot Vision, 2010.

[15] M.J. Swain, D.H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.

[16] D. Koubaroulis, The multimodal neighborhood signature for modeling object color appearance and applications in computer vision, Ph.D. Thesis, University of Surrey, UK, 2001.