The development of a human-centered object-based image retrieval engine

Eva M. van Rikxoort[1,4], Egon L. van den Broek[2,4], and Theo E. Schouten[3]

[1] Image Sciences Institute, University Medical Center Utrecht
Heidelberglaan 100, 3584 CX Utrecht, The Netherlands
eva@isi.uu.nl http://www.isi.uu.nl/
[2] Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
egon@few.vu.nl http://www.few.vu.nl/~egon/
[3] Institute for Computing and Information Science, Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
T.Schouten@cs.ru.nl http://www.cs.ru.nl/~ths/
[4] Nijmegen Institute for Cognition and Information, Radboud University Nijmegen
P.O. Box 9104, 6500 HE Nijmegen, The Netherlands

Abstract. The development of a new object-based image retrieval (OBIR) engine is discussed. Its goal was to yield intuitive results for users by using human-based techniques. The engine utilizes a unique and efficient set of 15 features: 11 color categories and 4 texture features, derived from the color correlogram. These features were calculated for the center object of the images, which was determined by agglomerative merging. Subsequently, OBIR was applied, using the color and texture features of the center objects on the images. The final OBIR engine, as well as all intermediate versions, were evaluated in a CBIR benchmark, consisting of the engine, the Corel image database, and an interface module. The texture features proved to be useful in combination with the 11 color categories. In general, the engine proved to be fast and yields intuitive results for users.

1 Introduction

Humans differ in all imaginable aspects. This is no different for the characteristics of human vision. However, "the variance of human vision characteristics is much smaller than the gap between the characteristics of human vision and computer vision [1]". The latter is frequently called the semantic gap in computer vision and content-based image retrieval (CBIR) [2].

In order to bridge this semantic gap, the usage of appropriate prior knowledge is very important [3]. Ontologies, user preference profiles, and relevance feedback techniques were developed to utilize such knowledge. However, such methods require an enormous effort and consequently can only be applied in a limited domain [4]. We address the semantic gap from another angle, since we aim at developing techniques that are human-based and may lead to generic methods that was applied in an unlimited domain.

Our approach to improve the performance of CBIR systems is twofold: (i) we utilize knowledge concerning human cognition and (ii) we exploit the strength of image processing techniques. From this perspective, we aim to develop new image processing, classification, and retrieval techniques, which have low computational costs and provide intuitive results for users [5].

These techniques were inspired by human visual short-term memory (vSTM). Human vSTM can encode multiple features only when these features are integrated into a single object, defined by the same coherent boundary. Moreover, it has a storage limit between four items [6] and (at least) fourteen items [7]. Intrigued by the efficiency of human vSTM, we adapted a similar approach for our image analysis techniques.

In sharp contrast with human vSTM, in CBIR the features color and texture are most often analyzed over the complete images. However, with such an average description of images, a loss of information is present; i.e., characteristics of parts of images (e.g., objects) are lost. Moreover, most CBIR image processing schemes use large feature vectors; e.g., PBIR-MM (144 features: 108 color and 36 texture related) [8] and ImageRover (768 features) [9]. Since we aim to yield intuitive results for users [5] using computationally cheap methods, we mimicked the characteristics of the vSTM. Subsequently, we do not utilize complex shapes but applied a coarse segmentation algorithm, based on agglomerative merging [10], as described in Section 3. The content of the selected segments of images are compared with each other, using the highly efficient 11 color quantization scheme (Section 2) and the color correlogram (Section 2.1). This setup was tested in CBIR benchmark (see Section 4 and [2]) and adapted (see Section 5), resulting in a new CBIR engine. The performance of the final engine was measured (see Sections 6 and 7). Finally, in Section 8, a brief discussion can be found.

2 Color and Texture in 11 categories

As mentioned by Forsyth and Ponse [11]: "It is surprisingly difficult to predict what colors a human will see in a complex scene." However, it is known that humans use 11 color categories (red, green, blue, yellow, orange, brown, pink, purple, black, white, and gray) with processing color. These 11 color categories are considered as being universal and as being optimal [12]. Van den Broek, Schouten, and Kisters [13] developed a method to describe the complete HSI color space, based on a limited set of experimentally determined, categorized colors. This method provided a unique color space segmentation, based on the 11 color categories, which can be applied as an 11 color categories, quantization scheme. We adopted this 11 color quantization scheme (or color space segmentation). Hence, the color distribution of images was characterized by a color vector with only 11 color values.

Next to color, texture is an important feature for the human visual system [14]. Texture analysis can be done based on intensity differences, but nevertheless, color is important in texture recognition of colorful image material. With respect to color representation, Fujii, Sugo, and Ando [14] stated that "consider-

ing the effective computational strategy in our visual system, it is quite possible that not all the information carried out by the high-dimensional sensory representation is preserved for rapid judgments of natural textures." Taken this into account, the 11 color category quantization scheme should perfectly fit the job, and was, therefore, applied for colorful texture analysis.

For the analysis of texture, various methods are available, such as: statistical methods (e.g., co-occurrence matrices and autocorrelation features), geometrical methods (e.g., Voronoi tessellation features and structural methods), model based methods (e.g., random field models and fractals), and signal processing methods (e.g., spatial domain filters, Fourier domain filtering, Gabor models, and Wavelet models). Originally, they were developed for gray-value images but some of them were recently adapted to fit texture analysis on color images.

2.1 The color correlogram

For the current research, one of the most intuitive texture analysis methods was applied: the color correlogram as suggested by Huang, Kumar, Mitra, Zhu, and Zabih [15], which is constructed from an image by estimating the pairwise statistics of pixel color. In order to (i) provide perceptual intuitive results and (ii) tackle the computational burden, the 11 color scheme for quantization of color was chosen.

The color correlogram $C_{\bar{d}}(i,j)$ counts the co-occur-rence of pixels with colors i and j at a given distance $\bar{d}$. The distance $\bar{d}$ is defined in polar coordinates (d,α), with discrete length and orientation. In practice, α takes the values 0°, 45°, 90°, and 135°. The color correlogram $C_{\bar{d}}(i,j)$ can now be defined as follows:

$$C_{i,j}^{\bar{d}}(I) = \Pr(\mathrm{I}(p_1) = i \wedge \mathrm{I}(p_2) = j || p_1 - p_2| = \bar{d}), \tag{1}$$

where Pr is probability and p_1 and p_2 are positions in the color image I. Let N be the number of colors in the image, then the dimension of the color correlogram $C_{\bar{d}}(i,j)$ will be $N \times N$, which is in our scheme 11×11. This algorithm yields a symmetric matrix. One direction insensitive color correlogram can be defined for each distance (d) by averaging the four color correlograms of the different angles (i.e., 0°, 45°, 90°, and 135°).

From the color correlogram, a large number of textural features can be derived, such as: energy, entropy, correlation, inverse difference moment, inertia, Haralick's correlation, cluster shade, and cluster prominence, which characterize the content of the image. Based on previous research [16], the combination of entropy, inverse difference moment, cluster prominence, and Haralick's correlation, with distance $d = 1$ was used, resulting in a vector of four texture features.

3 Image segmentation

The purpose of image segmentation is to divide an image into segments or regions that are useful for further processing the image. Many segmentation methods have been developed for gray level images and were later extended to color images; see Cheng, Jiang, Sung, and Wang [17] for an overview of them.

3.1 Segmentation by agglomerative merging

Segmentation was applied by agglomerative merging, as described by Ojala and Pietikäinen [10]. Their algorithm is a gray-scale image algorithm but was extended to color images using a color texture descriptor. The algorithm was applied using the color correlogram as texture descriptor that was based on the 11 color quantization scheme.

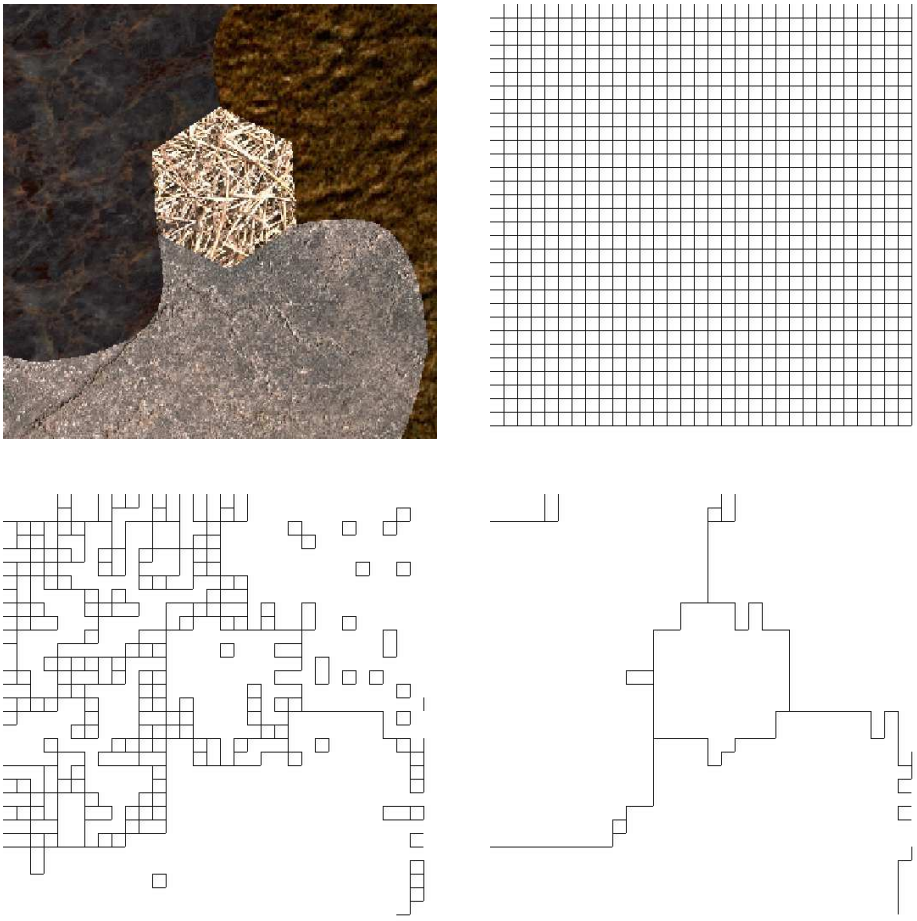


Fig. 1: The segmentation process, from left to right: The original image, division of the image in blocks of size 16×16, the regions after 800 iterations of agglomerative merging, and the final segments.

At the initial state of the agglomerative merging algorithm, the images were divided in sub blocks of size 16×16 pixels. At each stage of the merging phase, the pair of blocks with the lowest merger importance (MI) was merged. This merger importance is defined as follows:

$$MI \ = \ p \times L, \tag{2}$$

where p is the number of pixels in the smaller of the two regions an L is defined as:

$$L = |I - I'| = \sum_{i,j=0}^{m-1} |C_{i,j}^{\bar{d}}(I) - C_{i,j}^{\bar{d}}(I')|, \tag{3}$$

where m is the number of bins used and $C_{i,j}^{d}(I)$ is the color correlogram of image I (see Equation 1), and $\bar{d}$ is set to 1 (see Section 2.1). The closer L is to zero, the more similar the texture regions are. The agglomerative merging phase continues until the experimentally determined stopping criterion (Y), given in Equation 4 is met:

$$MI_{stop} \ = \ \frac{MI_{cur}}{MI_{max}} < Y, \tag{4}$$

where MI_{cur} is the merger importance for the current best merge, MI_{max} is the largest merger importance of all preceding merges. The agglomerative merging phase is illustrated in Figure 1.

3.2 Parameter determination

In order to use the segmentation algorithm, the parameter Y from Equation 4 had to be determined. This was done using a small test set of texture mosaics. In addition, three variations of the Merger Importance (MI), as given by Equation 2, were evaluated: (i) the form as given in Equation 2, (ii) $\sqrt{p}$ instead of p in calculating the MI value, and (iii) not using the number of pixels at all. The third variant showed to work best. Using a sample set, the threshold Y (see Equation 4) was experimentally set on 0.6000.

With the introduction of the segmentation algorithm, all ingredients for an image description are defined: the color correlogram (see Section 2.1), the 11 color categories (see Section 2), and coarse image segmentation. Next, we will discuss the CBIR benchmark, which includes the CBIR engine, which uses the image description.

4 CBIR benchmark

In order to perform image retrieval using the image features discussed in the previous sections, a test environment or benchmark has been developed [2]. The three main components of this benchmark are: (i) The CBIR engine, (ii) an image database, and (iii) the dynamic interface module.

The CBIR engine calculates a feature vector for each image or image segment. Based on this feature vector, the distance between the query image and all other images is calculated by means of a distance measure. The result of this CBIR engine is a list of the top 100 most similar images to the query image. The most important parameters that can be set for the engine are: the distance measure and the feature vector.

Since the benchmark is modular, an image database of choice can be used. In principle, every database can be connected to the benchmark; the most common file-types are supported.

The dynamic interface module generates an interface in which the results can be presented. By way of a set of parameters, a range of options can be altered. For example, one can set the number of images presented for each query, the number of queries to be judged, and choose whether the presentation of the results is in random order or not.

For the present research, we have chosen as main settings: the intersection distance measure, the Corel image database, which is a reference database in the field of CBIR, and a presentation of the top 15 images retrieved in a 5×3 matrix, randomly ordered (see Figure 2).

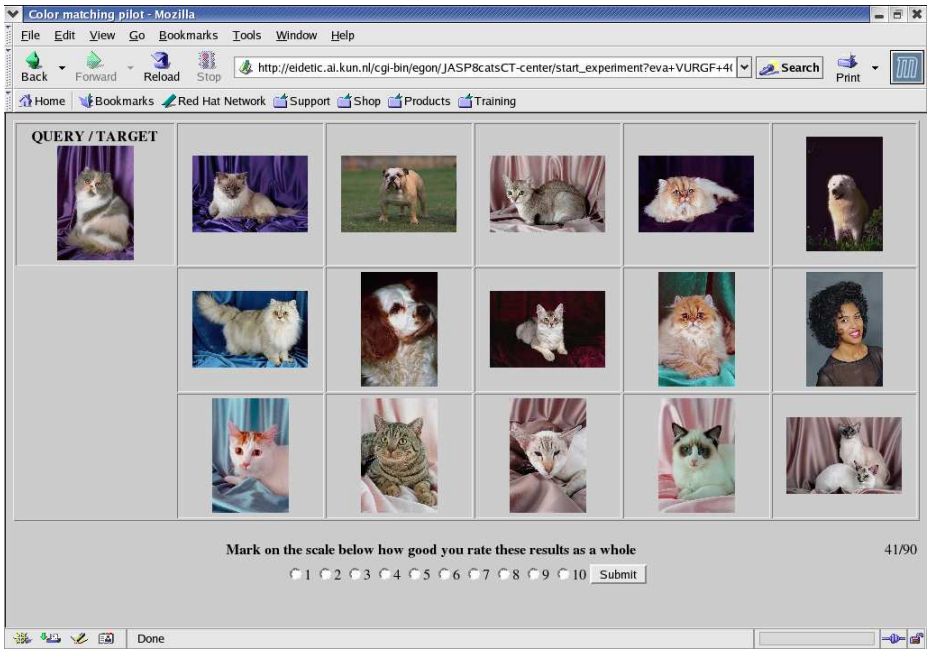


Fig. 2: A query image with retrieval results when using color and texture features for matching.

The histogram intersection distance (D) of Swain and Ballard [18] is used to calculate the difference between a query image(q) and a target image (t):

$$D_{q,t} = \sum_{m=0}^{M-1} \mid h_q(m) - h_t(m) \mid, \tag{5}$$

where M is the total number of bins, h_q is the normalized query histogram, and h_t is the normalized target histogram. This distance measure is developed for histograms but also works for texture feature vectors [19].

Three different feature vectors were used: (i) the histogram of the 11 color categories (see Section 2), (ii) the 4 texture features (see Section 2.1), and (iii) the color categories and texture features combined, resulting in a vector of length 15.

5 Phases of development of the CBIR engine

The final CBIR engine was developed in four phases. The final version of each of these phases can be found online; see Table 1 for the web-address of each of the 13 benchmarks, including the final benchmark. The results of each benchmark (in each phase) were judged by two experts, who each judged 50 random chosen queries on the quality of the retrieved images.

5.1 Phase 1

In the first phase of the development of the CBIR engine, the Corel image database (consisting of 60,000 images) was used as a test set. The segmentation algorithm, described in Section 3, was applied on each image in the database. Resulting segments were used for the CBIR engine if its area was more than or equal to 20% of the total area of the image; smaller ones were discarded.

People are, in most cases, interested in objects on the image [20]. Multiple objects can be present, not necessary semantically closely related (e.g., a person standing next to his car). So, one image can satisfy two unrelated queries (e.g., persons and cars). Hence, we have chosen to use each segment separately in searching the database of images.

In previous research on using texture based segmentation for CBIR, only one type of feature vector was chosen for the matching phase [19]. In a first attempt to apprehend the influence of texture in color image retrieval, three CBIR-engines were developed: a color-based, a texture-based, and a color&texture-based engine. With this approach we aim to evaluate the influence of texture features on the retrieval results.

Let us briefly summarize the results, as judged by the experts. The retrieval results of the color and of the color&texture-based engine were judged as being on an acceptable level. The results of the texture-based engine were very poor.

The inspection of the results revealed two problems: (i) The areas that exceeded the threshold of 20% did frequently form the background of the scene

presented on the image and (ii) Frequently, no area exceeded the threshold of 20%. These two problems indicate that often we were not able to detect objects in the images. Therefore, in Phase 2, we will try an alternative method for segmentation.

Table 1: The addresses of the 13 different benchmarks, using either color, texture, or a combination of both features. The * stands for `http://eidetic.ai.ru.nl/egon/`. The final benchmark is indicated as bold.

Phase	Color	Texture	Color and Texture
1	`*/JASP1`	`*/JASP2`	`*/JASP12`
2a	`*/JASP19c`	`*/JASP19t`	`*/JASP19`
2b	`*/JASP29c`	`*/JASP29t`	`*/JASP29`
3	`*/JASP8catsC`		`*/JASP8catsCT`
4	`*/JASP8catsC-center`		`*/JASP-final`

5.2 Phase 2

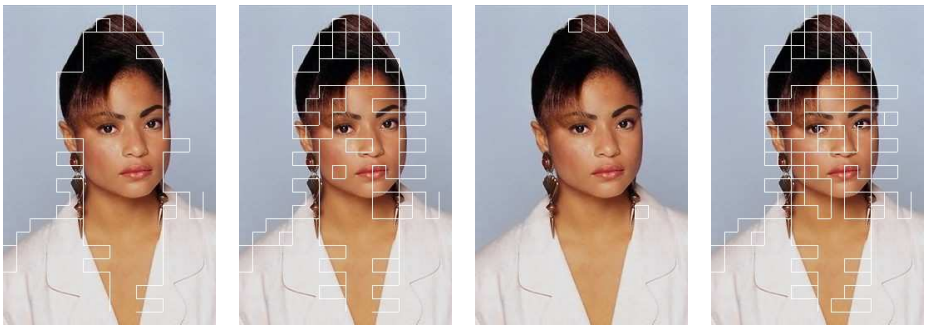
The making of most photos is initiated by the interest in certain objects. Therefore, the photographer will take care that an adequate presentation of the object(s) is present within the frame of the photo. In most cases, this means the object of interest is placed central in the photo. Thus, the central position of the image is of the utmost importance. This also holds for non-photo material: Imagine an image of a painting, of a sculpture, or of a cartoon. Also for this image material both the photographer as well as the artist who made the original, will place the object(s) in the center of the image.

Most images will present an object; but what to do with those images that present a scene (e.g., the sunrise on a photo or a landscape on a painting)? In such a case, the center of the image will not hold the object of interest but will hold a sample of the scene of interest. So, in one way or the other, the center of the image contains the most important information.

In order to investigate this hypothesis, we conducted a new research toward CBIR without image segmentation. We simply selected the center of the image. In order to do this, a grid of 3×3 grid cells was placed over the image. The center of the image was defined in two ways: (a) the center grid cell (see Figure 4b) and (b) both the center grid cell ans the cell below the center grid cell (see Figure 4c).

We were still interested in the influence of color, texture, and their combination (see Section 4). Hence, for each of the center definitions, three CBIR engines were developed, making a total of six CBIR engines developed in this phase (see also Table 1). The six engines retrieved their images from the complete Corel image database.

Similar to Phase 1, the engines relying on texture features solely performed poor. With that, the evidence was strengthened that texture solely is not useful



(a) (b) (c) (d)

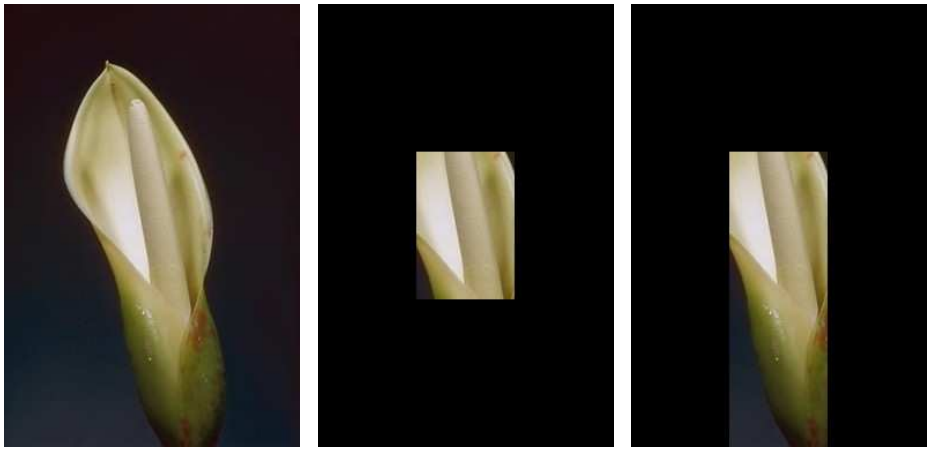
Fig. 3: Segmentation of images with several parameters: (a) The correct parameter for its class (0.700). (b) The generic parameter as used in phase 1 (0.600). (c) The parameter of the class cats (0.800). (d) The parameters of the class dinos (0.500).

for CBIR. Hence, in the next phases of development, texture features on its own will no longer be used. For the color and color&texture-based engines, the center image approach proved to be successful. According to the experts, the $\frac{1}{9}$ approach performed slightly better than the $\frac{2}{9}$ approach. However, the results were still far from satisfying.

5.3 Phase 3

In this Phase, we aim to tackle the problems of segmentation, due to the variety of images in image classes. In order to tune the segmentation algorithm, the parameter Y (see Equation 4) had to be set separately for each class of images used. Except from tuning the parameters, the segmentation is similar to the segmentation in Phase 1. In this phase, similarity based on color and a combination of color and texture were used. Both engines were applied on seven classes of the Corel image database (i.e., cats, dogs, food, flowers, women, waterfall, and dinos), resulting in a database of 900 images. For each of these seven classes, the segmentation algorithm was applied using its own parameter.

As expected, tuning the segmentation algorithm for each class separately improved the retrieval performance substantially. The effect of tuning the segmentation algorithm for each class separate is illustrated in Figure 3. Furthermore, including texture features in the engine, improved the retrieval, compared to the retrieval results of the engine using color solely. However, the results were still not fully satisfactory; therefore, in phase 4, a combination of phase 2 and phase 3 is applied.



(a) (b) (c)

Fig. 4: (a) The original image. (b) The $\frac{1}{9}$ center grid cell of the image as used for analysis. (c) The $\frac{2}{9}$ center grid cells of the image as used for analysis.

5.4 Phase 4: The final CBIR engine

Since both Phase 2 and Phase 3 provided promising results, we chose to combine both approaches: both the selection of the center of the image and the tuning of the segmentation for each class of images are utilized.

The procedure is as follows: (i) the image is segmented, (ii) the center grid cell is selected, and (iii) the region with the largest area within the segmented center grid cell is selected for analysis. So, for each image only one region represents the complete image. We assume that this region represents the object, which is the subject of the image. This process is illustrated in Figure 5.

The results of both the color and the color&texture-based engine were promising. The color&texture-based engine performed better than the engine based on color solely. So, finally a successful setup was found and the final CBIR-engine was defined. In order to validate the success of the engines, we wanted to conduct a more thorough analysis of the retrieval results. This process of validation is described in the next two sections.

6 Measuring performance

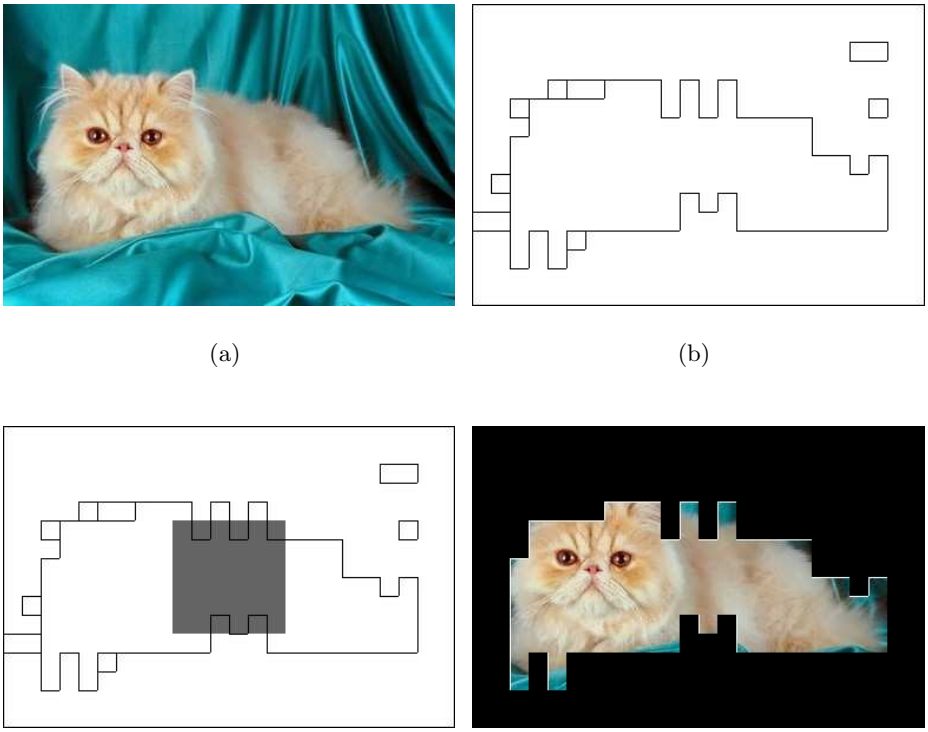
6.1 Recall and precision

Two methods of validation can be applied in CBIR; both adapted from the field of Information Retrieval. Given a classified database with labeled images, recall and precision of the retrieval results can be determined. Recall signifies

the percentage of relevant images in the database that are retrieved in response to the query. Precision is the proportion of the retrieved images that is relevant to the query.

In this experiment, it is not possible to determine recall of the system because the number of relevant image are not known beforehand. A similar problem is present when querying the Internet. However, in both cases the precision of the system can still be determined.

In most CBIR research, precision is determined automatically, provided a well annotated database. However, with such an approach a problem arises with the Corel image database as it is used. The classification is done with only one keyword. As a result separate categories (i.e., categories labeled by different keywords) can have considerable overlap.



(a) (b)

(c) (d)

Fig. 5: (a) The original image. (b) The segments in the image (c) The grid. (d) The final region.

In order to tackle this problem with automatic determination of precision, we utilized a manual determination of precision. Recently, this approach was successfully applied [5, 2]. Users were asked to judge the retrieved images as either related to the query image or as not related.

To facilitate the manual determination of precision, the benchmark was utilized. The users were asked to judge the images retrieved by comparing them to the query image. The judgment was binary: either an image was judged as appropriate and selected, or an image was judged as inappropriate and not selected. For each query, the top 15 images, as determined by the CBIR engine, were presented to the users. To facilitate a rapid judgment of the query results, the query images were pre-defined; i.e., the user did not have to search for and select a query image, a random selection of query images was already taken from the database. For each query, we can then define the precision of the presented images. The number of 15 retrieved images is a compromise. It is low enough to allow all images of a query to be presented on one screen in a size that is suitable for judgment. This optimizes the speed of judgment and thus maximizes the number of queries that can be judged.

6.2 Semantic and feature precision

In everyday life, search-engines are judged on their semantic precision; i.e., do the results have the same meaning as the query? However, two possible problems arise: (i) the query is ill defined or (ii) the search engine's algorithm are not able to interpret the query correct. The interest in this distinction lays in whether the user or the engine can be blamed.

In CBIR the same problems arise. However, since the field of CBIR is young relative to that of (text-based) Information Retrieval and its techniques are not fully grown, the problems have a larger impact on the judged semantic precision. However, it is not yet possible to search on semantics; it is done through the features that correlate strongly with semantic categories.

Frequently, users do not understand the results a CBIR query provides, when they are naive to the techniques behind the CBIR engine. For example, the query image can contain a dog with brown hair. The CBIR engine can return other dogs with brown hair (e.g., see Figure 2, but also cats with a brown coat and women with much brown hair. From a semantic point of view, the latter latter two results are incorrect; however, from a feature point of view, one can perfectly understand them.

We asked a group of eight users, who participated in judging the two CBIR engines, to judge the engines twice: once on semantic precision and once on precision based on features. In the next section, we will discuss the results of both of these judgments, for both engines, in general and for each class separately.

7 Results

In Section 5.4, the final color and color&texture-based engines were introduced. They use 11 color and 4 texture features of the center segment of each image. Since one of our main interests was whether or not texture features contribute to the correct classification and retrieval of images, both engines had to be judged by users.

In addition, in the previous section we have explained our interest in the difference between semantic precision and feature-based precision. For the latter judgments, the eight participating users were instructed to judge the retrieved images on the similarity with the query image, based on the patterns present (e.g., grass, hair, clouds) and on the color distributions.

These two differentiations result in four different situations in which precision of retrieval had to be determined. In total, the eight users judged 640 queries (20 per person per situation) and so provided a manually determined precision. The precision was determined over the top 15 matches of the queries, by selecting the images that are considered to be correctly retrieved (see also Section 6.1).

For each situation we determined the average number of selected images and with that the precision of the engine for each situation (see Table 2). Both the precision on feature level ($p < 0.0286$) and the precision on semantic level ($p < 0.0675$) is higher for the color&texture-based engine (feature: 8.51; semantic: 6.91) than for the color-based engine (feature: 7.39; semantic: 6.14).

In other words, no matter from which perspective the engines were judged, texture increased the precision of the retrieval performance. In addition, note that when the engines were judged on semantics significantly less images were selected then when judged on image features (color: $p < 0.0036$ and color&texture: $p < 0.0016$; see Table 2).

We will now present the average and standard deviation of the number of selected images for each of the seven classes separate, for each of the four situations (see Table 3). A large variance between the classes becomes apparent. The average number of images selected, for the seven classes, in the four situations, ranges from 2.20 (food; color-based, semantic) to 11.89 (dinos; color&texture-

Table 2: The average number of images selected when judging feature and semantic precision. The p values (determined by a two-tailed Student's t-test) indicate the difference between using only color features and using color and texture features as well as the difference between when judging feature-based or on semantic precision.

	color	color-texture	p value
feature	7.39	8.51	0.0286
semantic	6.14	6.91	0.0675
p value	0.0036	0.0016	

Table 3: The average number of images selected (i.e., indicating the precision) and the standard deviation (between brackets), for both engines (color and color&texture) on both feature and semantic precision.

Class	Color-based		Color&Texture-based	
	Feature	Semantic	Feature	Semantic
dinos	10.14 (5.04)	8.90 (4.99)	11.89 (4.11)	11.30 (4.54)
flowers	7.14 (3.92)	4.75 (2.12)	7.05 (5.08)	4.05 (2.11)
food	6.81 (3.11)	2.20 (2.14)	5.56 (4.57)	2.85 (3.36)
women	6.31 (4.16)	5.20 (2.98)	8.40 (5.24)	5.60 (2.64)
waterfall	11.27 (2.64)	7.05 (1.76)	11.46 (2.75)	7.90 (2.22)
cats	6.10 (4.03)	8.10 (3.39)	8.80 (4.94)	8.85 (3.62)
dogs	5.66 (2.54)	6.48 (2.50)	7.45 (5.06)	7.35 (2.57)

based, feature). Additionally, within most classes a considerable variability is present, as indicated by the standard deviations presented in Table 3.

Please note that all classes used are object-classes, except the class food. This class represents a concept on another level of semantics. The class food contained, for example, images of: plates with food on it, a champagne glass, people eating, and a picnic setting with a boat in a lake as background.

A class as heterogeneous as food, is impossible to classify with a high semantic precision. This is sustained by the poor results: 2.20 (color) and 2.85 (color& texture) images selected per query. In addition, the class food was the only class for which the use of texture substantially reduced the precision of retrieval. For the class flowers texture did decrease the precision of retrieval as well, but to a lower extent. For all other classes texture proved to be a useful feature for CBIR.

In general, for most classes an acceptable precision was achieved; for some queries even excellent (e.g., see Figure 2). However, the performance differed considerably between the classes and between the queries within these classes.

8 Discussion

The present paper provided an overview of the development cycle of new object-based CBIR techniques. These were evaluated in a CBIR benchmark, which provided the Corel image database and an interface module, for the engines developed. In order to provide intuitive results for users based on computationally cheap generic techniques, we mimicked human visual processing characteristics, utilizing the 11 color categories, four texture features derived from the color correlogram, and image segmentation by agglomerative merging. A central region from the image was chosen, such that it had a high probability to represent the object, which is the subject of the image. With a feature vector of 15 elements (i.e., the 11 colors + 4 texture features) and a segmentation algorithm based on the 11 color categories, the techniques introduced are very cheap.

The final color&texture-based engine proved to have a good precision. However, the engine is not generic applicable since it needs to be fine-tuned for dif-

ferent classes of images. This is due to the different background scenes against which the images in the Corel image database are photographed. So, the amount to which the objects differ in texture from their background is variable. This variability in texture differences between classes is the reason the parameters have to be fine-tuned for each object class.

In Section 6.2, we discussed the difference between feature and semantic precision. This is of interest since often the claim is made that a CBIR engine retrieves images based on semantic properties, while actually retrieval is based on image features that correlate with semantic categories. Feature precision was significantly higher than semantic precision for both the color-based engine and the color&texture-based engine. These results indicate that, when the retrieval results were not semantically relevant, they were intuitive to the users. Especially, heterogeneous image classes proved to be a problem for semantic precision, which was illustrated by the class food. We do not expect that images of such classes can be adequately classified or retrieved from a database using an object-based approach.

This paper describes the development of an efficient OBIR engine that provides good retrieval results. Its efficiency is founded on principles inspired by human perception. Moreover, it provides intuitive results for its users. Hence, an important step is made toward bridging the semantic gap present in CBIR.

Acknowledgments

The Dutch organization for scientific research (NWO) is gratefully acknowledged for funding the ToKeN Eidetic project (nr. 634.000.001). Further, we thank the reviewers for their detailed comments on the original manuscript.

References

1. Shirai, Y.: Reply performance characterization in computer vision. Computer Vision, Graphics, and Image Processing - Image Understanding **60** (1994) 260–261
2. Broek, E., Kisters, P.M.F., Vuurpijl, L.G.: Content-based image retrieval benchmarking: Utilizing color categories and color distributions. Journal of Imaging Science and Technology **49** (2005) [in press]
3. Rousson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based feature space. In: Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., Madison, Wisconsin (2003) 699–704
4. Zhang, R., Zhang, Z.M.: A robust color object analysis approach to efficient image retrieval. EURASIP Journal on Applied Signal Processing **4** (2004) 871–885
5. Müller, H., Müller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Performance evaluation in content-based image retrieval: Overview and proposals. Pattern Recognition Letters **22** (2001) 593–601
6. Wilken, P., Ma, W.J.: A detection theory account of visual short-term memory for color. Journal of Vision **4** (2004) 150a

7. Rensink, R.A.: Grouping in visual short-term memory [abstract]. Journal of Vision **1** (2001) 126a
8. Lai, W.C., Chang, C., Chang, E., Cheng, K.T., Crandell, M.: Pbir-mm: multimodal image retrieval and annotation. In Rowe, L., Merialdo, B., Muhlhauser, M., Ross, K., Dimitrova, N., eds.: Proceedings of the tenth ACM international conference on Multimedia. (2002) 421–422
9. Sclaroff, S., Taycher, L., la Cascia, M.: Imagerover: A content-based image browser for the world wide web. In Picard, R.W., Liu, F., Healey, G., Swain, M., Zabih, R., eds.: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries. (1997) 2–9
10. Ojala, T., Pietikäinen, M.: Unsupervised texture segmentation using feature distribution. Pattern Recognition **32** (1999) 477–486
11. Forsyth, D.A., Ponse, J.: Computer Vision: A modern approach. Pearson Education, Inc., Upper Saddle River, New Jerssey, U.S.A. (2002)
12. Derefeldt, G., Swartling, T., Berggrund, U., Bodrogi, P.: Cognitive color. Color Research & Application **29** (2004) 7–19
13. Broek, E., Schouten, T.E., Kisters, P.M.F.: Efficient color space segmentation based on human perception. ([submitted])
14. Fujii, K., Sugi, S., Ando, Y.: Textural properties corresponding to visual perception based on the correlation mechanism in the visual system. Psychological Research **67** (2003) 197–208
15. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In Medioni, G., Nevatia, R., Huttenlocher, D., Ponce, J., eds.: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (1997) 762–768
16. Broek, E., Rikxoort, E.: Evaluation of color representation for texture analysis. In Verbrugge, R., Taatgen, N., Schomaker, L.R.B., eds.: Proceedings of the 16th Belgium-Netherlands Artificial Intelligence Conference, Groningen - The Netherlands (2004) 35–42
17. Cheng, H., Jiang, X., Sung, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recognition **34** (2001) 2259–2281
18. Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision **7** (1991) 11–32
19. Yao, C.H., Chen, S.Y.: Retrieval of translated, rotated and scaled color textures. Pattern Recognition **36** (2003) 913–929
20. Schomaker, L., Vuurpijl, L., de Leau, E.: New use for the pen: outline-based image queries. In: Proceedings of the 5th IEEE International Conference on Document Analysis, Piscataway (NJ), USA (1999) 293–296