International Journal of Image and Graphics, Vol. 1, No. 3 (2001) 387–414

INTERACTIVE RETRIEVAL OF COLOR IMAGES

MARCEL WORRING* and THEO GEVERS†,‡

Intelligent Sensory Information Systems,
Faculty of Science, University of Amsterdam, Kruislaan 403,
1098 SJ Amsterdam, The Netherlands

Retrieval of color images has become an important application in recent years. We make a concise analysis of methodologies for interactive retrieval of color images. Two issues are of importance. First, the domain, which can be *broad* or *narrow*. Second, the search method, which can be *object search*, *target search*, *category search* or *associative search*. On the basis of these, we give complete guidelines for choosing and designing methods for interactive color image retrieval based on the domain and search goal characteristics.

Keywords: Color Indexing, Color Invariance; Similarity; Information Visualization; Query Space.

1. Introduction

Today, with the growth and popularity of the World Wide Web, a new application field is born through the tremendous amount of visual information, such as images and videos, which has been made accessible publicly. Apart from this, companies are starting to digitize all their images and videos. This growth is reflected in the many conference series on this topic that have started in recent years like the content-based workshops of the CVPR and ICCV, the SPIE storage and retrieval for image and video databases conference, and the Visual Information Systems series.

Color plays an important role in all aspects of visual data on the web. Aside from decorating and advertising potentials for web-design, color information has become a powerful tool in content-based image and video retrieval.

Various color-based image search schemes have been proposed based on various representation schemes such as color histograms, color moments, color edge orientation, color texture and color correlograms. These image representation schemes have been created on the basis of different color spaces. Which one to choose depends on the dataset and the search goals of the user. There is a need to get a better insight in the possibilities and limitations of the different representation schemes.

*E-mail: worring@science.uva.nl
†E-mail: gevers@science.uva.nl
‡This work was supported in part by the ICES MIA-project. The authors would further like to acknowledge the discussions with Arnold Smeulders on the topics in this paper.

Even with the best representation schemes, it is seldom possible to ignore the important role of interaction with the user. The user should play an active role in the retrieval process and the system should employ any information that can be provided by the interacting user. This requires an intimate interplay between the system and the user. Again, the proper choice of an interaction methodology depends on the search goals of the user. Hence, these search goals should be examined carefully.

The paper is organized as follows, in Sec. 2, the datasets and search goals of users are categorized on the basis of the query space framework. A taxonomy of color spaces based on the dataset and search goal categorization is put forward in Sec. 3. In Sec. 4, the different methods for indexing are analyzed. Finally, in Sec. 5, different interaction methodologies are described and put into the query space framework.

2. Color Image Retrieval

2.1. *Datasets and applications*

Color image datasets arise in many different applications in varying domains. To define methods for color image retrieval, it is important to consider carefully the classes of datasets one can encounter. Furthermore, methods depend on the search goals of the user.

From the dataset point of view, a distinction can be made between *narrow domains* and *broad domains*. Examples of datasets in narrow domains are pictures of 20th century architecture, images of flowers in a catalogue or images of paintings in a museum. In narrow domains, the images are typically derived under controlled circumstances. Characteristics of the imaging device are known and lighting conditions can be optimized, hence images are of high quality. This contrasts the variety in quality and devices encountered in broad domains. The broadest domain clearly being the world wide web.

When accessing a dataset, users can vary broadly in their goals of using the color image retrieval system. In general, we distinguish four major categories of search goals:

- *object search*: the search for a specific object
- *target search*: the search for a specific image
- *category search*: the search for one or more images from a specific category
- *associative search*: browsing through the collection with no other goal than interesting findings

When the user is performing an object search, he is likely to have the object at his disposal. The object can either be in its concrete form or it can be a picture of the object. In both cases, the goal is to verify whether there is an image in the database that contains the same object. The concern here is to find the object even if it is only partly visible, or recorded under different circumstances, e.g., seen from

a different viewpoint. When the image of the object is identified, the object should be localized in the picture.

In target search, the individual objects are not of primary importance, but the composition of the picture as a whole. The user has a mental model of the image, which should be communicated to the system to find the image. For this purpose, it is of great importance that the user can specify the colors he has in mind in an intuitive way.

Category search focuses on the common characteristics of groups of images. Especially those characteristics that distinguishes this group from other groups are of great importance.

The final category to consider is associative search. As the system cannot predict what is of interest to the user, it is important that the user takes the lead in the search process. To that end, it is crucial that in the perception of the user, the effect of his choices are reflected in the progress made in the search space. This can only be the case if the image descriptions relate to the perception of the user.

2.2. *Query space: definition*

The search categories defined in the previous section require different retrieval methods. We now present a formalism called Query Space as was first put forward in Ref. 1. Here, it will be made specific for the goal of structuring different interactive color retrieval methods.

The basis for image retrieval is of course the active set I_Q of images in the database. This can be the whole database, but also in some view the user is given on the database.

Images are never retrieved on the basis of the full pixel array of color values. Therefore, all the images should be described in abstract way using a set of color features F_Q. The color features to select depend on the aim the user has. In Sec. 3, guidelines for the selection will be derived. The set of features span a high-dimensional space in which each image corresponds to a single point according to the feature values derived for this particular image.

Based on the feature set, the system must be able to compare images. Which images are similar and which are dissimilar? This is captured in a similarity function S as will be described in Sec. 4. Often, the similarity function is Euclidean distance in feature space. As similarity is context dependent, typically weighting coefficients are used for the individual elements of the feature vector,[2,3] which can be set by the user or system.

Finally, with each image or group of images, an interpretation can be associated. These could on the one hand be semantic labels. On the other hand, they can be related to the search goal. Due to the fact that images are sensory observations of the world, labels cannot be assigned with full certainty. Therefore, when a label is attached to an image or image group, a probability is stored for the image-label pair. The set of possible labels is denoted by Z.

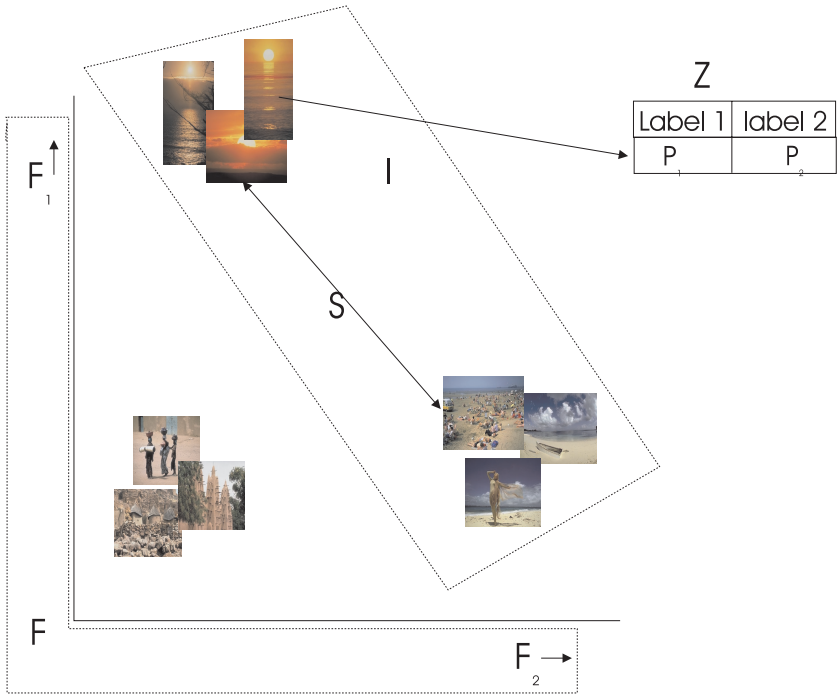

Fig. 1. Abstract representation of the query space. The image set I forms a true subset of the full set of images in the database. The feature space F is composed of two color features. Similarity S corresponds to Euclidean distance. Finally, with the images, two labels in Z are assigned with their associated probabilities P.

Given the above, query space is defined as:

Definition 1. The query space Q is the goal dependent 4-tuple $\{I_Q, F_Q, S_Q, Z_Q\}$.

The query space forms the basis for specifying queries, display of query results and for interaction, which will be described in Sec. 5. We now turn our attention to the proper definition of color features and similarity functions. An overview of the framework is presented in Fig. 1.

3. Color Taxonomy for Image Retrieval

The choice of color features is of great importance for the purpose of proper image retrieval. It induces the equivalence image classes to the actual retrieval algorithm. However, no color system can be considered as universal because color can be interpreted and modeled in different ways. Each color system has its own set of color models, which are the parameters of the color system. Color systems have been developed for different purposes: (a) display and printing processes: *RGB*, *CMY*; (b) television and video transmission efficiency: *YIQ*, *YUV*; (c) color

standardization: XYZ; (d) color uncorrelation: $I_1I_2I_3$; (e) color normalization and representation: rgb, xyz; (f) perceptual uniformity: $U^*V^*W^*$, $L^*a^*b^*$, $L^*u^*v^*$; and (g) intuitive description: HSI, HSV. With this large variety of color systems, the inevitable question arises, which color system to use for which kind of image retrieval application. To this end, criteria are put forward to classify the various color systems for the purpose of content-based image retrieval.

In this section, the aim is to provide a taxonomy on color systems according to the following criteria:

- is the color system device independent (broad/narrow domain)
- is the color system perceptual uniform (category/associate search)
- is the color system intuitive (query specification)
- is the color system robust against varying imaging conditions (object/target search)
  - — invariant to a change in viewing direction
  - — invariant to a change in object geometry
  - — invariant to a change in the direction of the illumination
  - — invariant to a change in the intensity of the illumination
  - — invariant to a change in the spectral power distribution (SPD) of the illumination.

The first criterion is focussed on the independence of the color system on the underlying imaging device. This is required when the images in the image database arise from broad domains, where recordings are made by different imaging devices such as scanners, cameras, digital videos and webcam-recorder (e.g., images on Internet). In contrast, for narrow domains, images are usually recorded on controlled circumstances by the same imaging device. The second criterion states that the color system should exhibit perceptual uniformity, meaning that numerical distances within the color space can be related to human perceptual differences. This is important when images are to be retrieved, which should be visually similar such as stamp, trademark and painting databases. Perceptual uniformity is needed for category and associate search. Thirdly, the color system should be composed of color models, which are understandable and intuitive to the user. This is required for proper and easy query specification, where, for example, a color picker is used to select the proper color range of interest. Moreover, to achieve robust and discriminative image retrieval, color invariance is another important criterion. Especially for object search, where the goal is to find images containing the same object(s) as shown by the query image. In general, images and videos are taken from objects from different viewpoints. Two recordings made of the same object from different viewpoints will yield different shadowing, shading and highlighting cues changing the intensity data fields considerably. Moreover, large differences in the illumination color will drastically change the photometric content of images

even when they are taken from the same object. Hence, a proper retrieval scheme should be robust to the imaging conditions discounting the disturbing influences of a change in viewpoint, object pose and illumination.

The color system taxonomy, proposed in this section, can be used to select the proper color system for a specific application based on whether images come from broad domains and which search goals are at hand (object/target/category/associate search). For example, consider an image database of textile printing samples (e.g., curtains). The application is to search for samples with similar color appearances. When the samples have been recorded under the same imaging conditions (i.e., camera, illumination and sample pose), a perceptual uniform color systems (e.g., $L^*a^*b^*$) is most suitable. When the lightning conditions are different between the recordings, a color invariant system is most appropriate eliminating the disturbing influences such as shading, shadows and highlights.

3.1. *Color features F*

The mostly used model is the *grey*-value or *intensity*-feature, which is obtained by a standard grey-value camera or can be calculated from the original R, G and B tristimulus values from the corresponding red, green and blue images provided by a CCD color camera (e.g., NTSC):

$$grey(R, G, B) = 0.299R + 0.587G + 0.144B\,. \tag{1}$$

Of course, grey-value images are dependent on the imaging device because two different cameras (i.e., filters) will yield different grey-value images for the same scene. Moreover, grey is heavily influenced by the viewing direction, object geometry, direction of the illumination, intensity and color of the illumination.

The RGB color system represents the (R)ed, (G)reen and (B)lue color. In general, the R, G and B color features correspond to the primary colors, where $R = 700$ nm, $G = 546.1$ nm and $B = 435.8$ nm. Similar to grey-value, the RGB color system is not perceptual uniform and is device-dependent. Therefore, RGB should not be used for image retrieval for images from broad-domains. Further, RGB depends on the imaging conditions such as viewing direction, object geometry, direction of the illumination, intensity and color of the illumination. Hence, using RGB values for image retrieval causes severe problems when the query and target image are recorded under different imaging conditions. In conclusion, RGB is only suitable for object or target search from narrow domains (i.e., the same imaging device) under exactly the same imaging conditions.

The rgb color system is defined as follows:

$$r(R, G, B) = \frac{R}{(R + G + B)}\,, \tag{2}$$

$$g(R, G, B) = \frac{G}{(R + G + B)}\,, \tag{3}$$

$$b(R,G,B) = \frac{B}{(R+G+B)}\,. \tag{4}$$

These color models are called normalized colors or chromaticity coordinates, because each of them is calculated by dividing R, G and B by their total sum. Because the r, g and b chromaticity coordinates depend only on the ratio of R, G and B (i.e., factoring luminance out), they have the important property that they are not sensitive to shading, surface orientation, illumination direction and illumination intensity.[4] However, normalized colors are still device dependent. Moreover, rgb become unstable and meaningless when the intensity is small.[5] In conclusion, rgb is well suited for object search in broad-domains (i.e., under varying imaging conditions but with the same SPD of the light source). However, images should be recorded by the same camera.

For standardization of colorimetric measurements, in 1931, the international lighting commission (CIE) recommended the XYZ-color system. Any perceived color can be described mathematically by the amounts of these three color primaries. The luminance is determined only by the Y value. Because the XYZ system is a linear combination of R, G and B values, the XYZ color system inherits all the dependencies on the imaging conditions from the RGB color system. Note that the color system is device-independent as the X, Y and Z values are objective in their interpretation. The following conversion matrix is based on the RGB NTSC color coordinates system:

$$X(R,G,B) = 0.607R + 0.174G + 0.200B\,, \tag{5}$$

$$Y(R,G,B) = 0.299R + 0.587G + 0.114B\,, \tag{6}$$

$$Z(R,G,B) = 0.000R + 0.066G + 1.116B\,. \tag{7}$$

Further, the corresponding chromaticity coordinates are given by:

$$x(X,Y,Z) = \frac{X}{(X+Y+Z)}\,, \tag{8}$$

$$y(X,Y,Z) = \frac{Y}{(X+Y+Z)}\,, \tag{9}$$

$$z(X,Y,Z) = \frac{Z}{(X+Y+Z)}\,. \tag{10}$$

Similar to rgb, this system cancels intensity out yielding independence on surface orientation, illumination direction and illumination intensity.[4] In conclusion, xyz is well suited for object search for broad domains with varying imaging conditions (color invariant but not color constant) and different imaging devices (device-independent).

Further, the CIE introduced the $U^*V^*W^*$ color system to obtain perceptual uniformity. The color model W^* is based on the scaling of luminance. The luminance of a color is determined only by its Y value. To scale luminance between 0

(black) and 100 (white), the scaling method starts with black and selects a *just noticeable brighter* grey-value. Taking this just noticeable brighter grey-value, the next just noticeable brighter grey-value is selected. This process continues until white is reached. The other two color features solve the problem of large difference of the axis-diameters of the ellipses in the chromaticity diagram. Colors, which are not noticeable different for a particular color are lying on the ellipses, and all colors, which are (just) noticeable different are lying outside the ellipses. The system is visual uniform, because a luminance difference corresponds with the same noticed luminance difference and the ellipses in the adjusted chromaticity diagram have constant axis-diameters. However, the U^* and V^* color models become unstable and meaningless when intensity is small.[6] Further, the $U^*V^*W^*$ color system depends on viewing direction, object geometry, highlights, direction of the illumination, intensity and color of the illumination. Another perceptual uniform system, proposed by CIE, is the $L^*a^*b^*$ color system. The color feature L^* correlates with the perceived luminance and corresponds to W^* of the $U^*V^*W^*$ color system. Color feature a^* correlates with the red–green content of a color and b^* reflects the yellow–blue content. The color system is device-independent and perceptual uniform. However, similar to $U^*V^*W^*$, the $L^*a^*b^*$ color system is still dependent on viewing direction, object geometry, highlights, direction of the illumination, intensity and color of the illumination. In conclusion, color systems $U^*V^*W^*$ and $L^*a^*b^*$ are particularly suited for category and associate search from broad-domains. Further, these color systems are also suited for object and target search for image coming from narrow-domains (i.e., under controlled imaging circumstances) possibly recorded by different imaging devices (i.e., device-independent).

The National Television Systems Committee (NTSC) developed the following three color attributes:

$$Y(R,G,B) = 0.299R + 0.587G + 0.114B\,, \tag{11}$$

$$I(R,G,B) = 0.596R - 0.274G - 0.312B\,, \tag{12}$$

$$Q(R,G,B) = 0.211R - 0.523G + 0.312B\,, \tag{13}$$

for transmission efficiency. The tristimulus value Y corresponds to the luminance of a color. I and Q correspond closely the hue and saturation of a color. By reducing the spatial bandwidth of I and Q without noticeable image degradation, efficient color transmission is obtained. For the PAL and SECAM standards used in Europe, the Y, U and V tristimulus values are used. The I and Q color attributes are related to U and V by a simple rotation of the color coordinates in color space.

The human color perception is conveniently represented by the following set of color features: I(ntensity), S(aturation) and H(ue):

$$I(R,G,B) = \frac{(R+G+B)}{3}\,, \tag{14}$$

$$H(R,G,B) = \arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right), \tag{15}$$

$$S(R,G,B) = 1 - \frac{\min(R,G,B)}{R+G+B}\,. \tag{16}$$

I is an attribute in terms of which a light or surface color may be ordered on a scale from dim to bright. S denotes the relative white content of a color and H is the color aspect of a visual impression. The problem of hue is that it becomes unstable when S and I are near zero due to the nonremovable singularities in the nonlinear transformation, where a small perturbation of the input RGB-values can cause a large jump in the transformed values.[6] Saturation becomes unstable when intensity is near zero. Intensity I depends on viewing direction, object geometry, direction of the illumination, intensity and color of the illumination. Saturation S depends on highlights and a change in the color of the illumination. Hue H depends only on the color of the illumination. In conclusion, the IHS system is well suited for proper color query specification such as the use of a color picker to specify colors ranges. Further, for narrow-domains, it could be used for category and associate search. Finally, hue is a good candidate for object search (color invariance but not color constant) of colorful objects (stability), when the recordings have been made by the same imaging device.

3.2. *Color constancy*

As stated before, the color (or rather, the apparent color) of an object varies with changes in illuminant color, illumination geometry (i.e., angle of incidence), viewing geometry (angle of reflectance) and miscellaneous sensor parameters. In outdoor images, the color of the illuminant (i.e., daylight) varies with the time-of-day, cloud cover and other atmospheric conditions. The illuminant and viewing geometry vary with changes in object and camera position and orientation. In addition, certain sensor response parameters, shadows and interreflection, may also affect the apparent color of objects. Consequently, at different times of the day, under different weather conditions and at various positions and orientations of the object and camera, the apparent color of an object can be different. *Color invariance* aims to discount the illumination geometry, viewing geometry and miscellaneous sensor parameters to obtain object reflectance color (i.e., surface albedo). Further, the goal of *color constancy* is to discount the illumination color to obtain the object reflectance color.

The problem of color constancy has been the topic of much research in psychology and computer vision. Existing color constancy methods require specific *a priori* information about the observed scene (e.g., the placement of calibration patches of known spectral reflectance in the scene), which will not be feasible in practical situations.[7–9] In contrast, without any *a priori* information, Healey and Slater[10] and Finlayson *et al.*[11] use illumination-invariant moments of color distributions for object recognition. However, these methods are sensitive to object occlusion and

cluttering as the moments are defined as an integral property on the object as one. In global methods in general, occluded parts will disturb recognition. Slater and Healey[12] circumvent this problem by computing the color features from small object regions instead of the entire object. Further, to avoid sensitivity on object occlusion and cluttering, simple and effective illumination-independent color ratio's have been proposed by Funt and Finlayson,[13] Nayar and Bolle,[14] and Gevers and Smeulders.[4] These color constant models are based on the ratio of surface albedos rather than the recovering of the actual surface albedo itself. However, these color models assume that the variation in spectral power distribution of the illumination can be modeled by the coefficient rule or von Kries model, where the change in the illumination color is approximated by a 3×3 diagonal matrix among the sensor bands and is equal to the multiplication of each RGB-color band by an independent scalar factor. The diagonal model of illumination change holds exactly in the case of narrow-band sensors. Although standard video cameras are not equipped with narrow-band filters, spectral sharpening could be applied[15] to achieve this to a large extend.

The color ratio's proposed by Nayar and Bolle are given by[14]:

$$N(C^{\mathbf{x}_1}, C^{\mathbf{x}_2}) = \frac{C^{\mathbf{x}_1} - C^{\mathbf{x}_2}}{C^{\mathbf{x}_2} + C^{\mathbf{x}_1}}, \tag{17}$$

and those proposed by Funt and Finlayson[13] are defined by:

$$F(C^{\mathbf{x}_1}, C^{\mathbf{x}_2}) = \frac{C^{\mathbf{x}_1}}{C^{\mathbf{x}_2}}, \tag{18}$$

expressing color ratio's between two neighboring image locations, for $C \in \{R, G, B\}$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the image locations of the two neighboring pixels.

The color ratio's of Gevers and Smeulders are given by[4]:

$$M(C_1^{\mathbf{x}_1}, C_1^{\mathbf{x}_2}, C_2^{\mathbf{x}_1}, C_2^{\mathbf{x}_2}) = \frac{C_1^{\mathbf{x}_1} C_2^{\mathbf{x}_2}}{C_1^{\mathbf{x}_2} C_2^{\mathbf{x}_1}}, \tag{19}$$

expressing the color ratio between two neighboring image locations, for C_1, $C_2 \in \{R, G, B\}$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the image locations of the two neighboring pixels. All these color ratio's are device-dependent, not perceptual uniform and they become unstable when intensity is near zero. Further, N and F are dependent on the object geometry. M has no viewing and lighting dependencies.

3.3. *Discussion*

It has been shown that each color system has its own characteristics: a number of systems are linear combinations of the RGB values, such as the XYZ color system, or normalized with respect to intensity, such as the rgb and the xyz color system. The $U^*V^*W^*$ and the $L^*a^*b^*$ color systems have distances, which reflect the perceived similarity. As each image retrieval application demands a specific color system, in Fig. 2, a color taxonomy is given. In conclusion, xyz is well suited

Color system	Device indep.	Perc. Uniform	Linear	Intuitive	View point	Object shape	Highlights	Illum. Intensity	Illum. SPD
RGB	-	-	+	-	-	-	-	-	-
XYZ	+	-	+	-	-	-	-	-	-
Norm. rgb	-	-	-	-	+	+	-	+	-
Norm. xyz	+	-	-	-	+	+	-	+	-
L*a*b*	+	+	-	-	-	-	-	-	-
U*V*W*	+	+	-	-	-	-	-	-	-
I1I2I3	-	-	+	-	-	-	-	-	-
YIQ	-	-	+	-	-	-	-	-	-
YUV	-	-	+	-	-	-	-	-	-
Intensity	-	-	+	+	-	-	-	-	-
Hue	-	-	-	+	+	+	+	+	-
Saturation	-	-	-	+	+	+	-	+	-
F, N	-	-	-	-	+	-	-	+	+
M	-	-	-	-	+	+	-	+	+

Fig. 2. Overview of the dependencies differentiated for the various color systems. + denotes that the condition is satisfied, − denotes that the condition is not satisfied.

for object and target search for broad domains (device-independent) recorded under varying imaging conditions under the constraint of a specific light source (color invariant but not color constant). Color systems $U^*V^*W^*$ and $L^*a^*b^*$ are particularly suited for category and associate search from broad-domains. The *IHS* system is well suited for color query specification. Further, the hue color feature H is a good candidate for object search (color invariance but not color constant) of colorful objects. For broad scale image datasets such the Internet, where no constraints can be applied on the imaging conditions, a color constant space should be used such as the N, F and M.

The various color systems and their performance can be experienced within the Pic2Seek systems on-line at: http://www.science.uva.nl/research/isis/zomax/.

4. Color Indexing

Various color based image search schemes have been proposed based on various representation schemes such as color histograms, color moments, color edge orientation, color texture and color correlograms.[16,17]

In this section, an overview will be given on color histograms in Sec. 4.1. Then, in Sec. 4.2, standard similarity measures are discussed to compute similarity between color histograms. As color histograms do not incorporate spatial information, we will give an overview of spatial color distribution schemes in Sec. 4.3.

4.1. *Color histograms*

The goal of histogram construction is the reduction of the number of colors representing the content of an image. The (one-dimensional) histogram is defined as:

$$\hat{f}(x) = \frac{1}{nh}\,(\text{number of } X_i \text{ in the same bin as } x)\,, \tag{20}$$

where n is the number of pixels X_i in the image, h is the bin width and x the range of the data. Two choices have to be made when constructing a histogram. First, the bin-width parameter needs to be chosen. Secondly, the position of the bin edges needs to be established. Both choices affect the resulting estimation.

One of the earlier approaches to color-based image matching, using the color at pixels directly as indices, has been proposed by Swain and Ballard.[18] If the opponent color (a linear transformation of RGB) distributions of two images are globally similar, the matching rate is high. The work by Swain and Ballard had an enormous impact on color-based histogram matching resulting in many histogram variations.

For example, the QBIC system[16] allows for a user-defined computation of the histogram by the introduction of variable k denoting the number of bins of the histogram. Then, for each $3 \times k$ cells, the average modified Munsell color is computed together with the five most frequently occurring colors. Using a standard clustering algorithm, they obtain k super cells resulting in the partitioning of the color system.

In Gevers,[19] various color invariant features are selected to construct color pattern-cards. First, histograms are created in a standard way. Because the color distributions of histograms depend on the scale of the recorded object (e.g., distance object-camera), they define color pattern-cards as thresholded histograms. In this way, color pattern-cards are scale-independent by indicating whether a particular color model value is substantially present in an image or not. Matching measures are defined, expressing similarity between color pattern-cards, robust to a substantial amount of object occlusion and cluttering. Based on the color pattern-cards and matching functions, a hashing scheme is presented offering run-time image retrieval independent of the number of images in the image database.

In the ImageRover system, proposed by Sclaroff,[20] the $L^*u^*v^*$ color space is used, where each color axis has been split into four equally sized bins resulting in a total of 64 bins. Further, Dimai[21] uses the $L^*a^*b^*$ system to compute the average color and covariance matrix of each of the color channels. Smith and Chang[22] use the HSV color space to obtain a partition into 144 bins giving more emphasis on hue then value and saturation. Further, Androutsos[23] also focuses on the HSV color space to extract regions of dominant colors. To obtain colors, which are perceptually the same but still being distinctive, Syeda-Mahmood[24] proposes to partition the RGB color space into 220 subspaces. Di Sciascio[25] computes the average color describing a cell of 4×4 grid, which is superimposed on the image. The MARS[26] uses the $L^*a^*b^*$ color space because the color space consists of perceptually uniform colors, which better matches the human perception of color. Gong,[27] roughly

partitions the Munsell color space into eleven color zones. Similar partitioning have been proposed by Cox[28] and Ciocca.[29]

Another approach, proposed by Stricker and Orengo,[30] is the introduction of the cumulative color histogram, which generate more dense vectors. This enables us to cope with coarsely quantized color spaces. Zhang[31] proposes a variation of the cumulative histograms by applying cumulative histograms to each sub-space.

Other approaches are based on the computation of moments of each color channel. For example, Appas[32] represents color regions by the first three moments of the color models in the HSV-space. Instead of constructing histograms from color invariants, Healey and Slater[10] and Finlayson *et al.*[11] propose the computation of illumination-invariant moments from color histograms. In a similar way, Slater and Healey[12] computes the color features from small object regions instead of the entire object.

Jacobs[33] proposes to use integrated wavelet decomposition. In fact, the color features generate wavelet coefficients together with their energy distribution among channels and quantization layers. Similar approaches based on wavelets have been proposed by Vellaikel and Kuo[34] and Liang and Kuo.[35]

4.2. *Similarity measures for histograms*

As stated before, histograms are created by counting the number of times a discrete color feature occurs in the image. The histogram from the query image is created in a similar way. Then, image retrieval is reduced to the problem to what extent histogram $\mathbf{k}$ derived from the query image $\mathcal{Q}$ is similar to a histogram $\mathbf{l}$ constructed for each image in the image database. A similarity function $\mathcal{D}(\mathbf{k}, \mathbf{l})$ is required returning a numerical measure of similarity.

Before the query can be submitted, a choice has to be made for the desired classes of similarity robustness. For each image retrieval query, a proper definition of the desired robustness is in order. Does the applicant wish to search for the object in real-world cluttered environments containing occlusion? Therefore, depending on the domain (broad/narrow), the following criteria are defined on the similarity measure:

(1) robustness against object fragmentation
(2) robustness against (self) occlusion
(3) robustness against clutter by the presence of other objects in the scene.

Various distance functions have been proposed. Some of these are general functions such as Euclidean distance and cosine distance. Others are specially designed for image retrieval such as histogram intersection.[18]

The Minkowski-form distance for two vectors or histograms $\mathbf{k}$ and $\mathbf{l}$ with dimension n is given by:

$$\mathcal{D}_M^k(\mathbf{k}, \mathbf{l}) = \sqrt{\sum_{i=1}^{n} |k_i - l_i|^k}\,. \tag{21}$$

The Eulidean distance between two vectors $\mathbf{k}$ and $\mathbf{l}$ is defined as follows:

$$\mathcal{D}_E(\mathbf{k},\mathbf{l}) = \sqrt{\sum_{i=1}^{n}(k_i - l_i)^2}\,. \tag{22}$$

The Euclidean distance is an instance of the Minkowski distance with $k = 2$.

The cosine distance compares the feature vectors of two images and returns the cosine of the angle between the two vectors:

$$\mathcal{D}_C(\mathbf{k},\mathbf{l}) = 1 - \cos\phi\,, \tag{23}$$

where ϕ is the angle between the vectors $\mathbf{k}$ and $\mathbf{l}$. When the two vectors have equal directions, the cosine will add to one. The angle ϕ can also be described as a function of $\mathbf{k}$ and $\mathbf{l}$:

$$\cos\phi = \frac{\mathbf{k}\cdot\mathbf{l}}{\|\mathbf{k}\|\,\|\mathbf{l}\|}\,. \tag{24}$$

The cosine distance is well suited for features that are real vectors and not a collection of independent scalar features.

The histogram intersection distance compares two histograms $\mathbf{k}$ and $\mathbf{l}$ of n bins by taking the intersection of both histograms:

$$\mathcal{D}_H(\mathbf{k},\mathbf{l}) = 1 - \frac{\sum_{i=1}^{n}\min(k_i,l_i)}{\sum_{i=1}^{n}k_i}\,. \tag{25}$$

When considering images of different sizes, this distance function is not a metric due to $\mathcal{D}_H(\mathbf{k},\mathbf{l}) \neq \mathcal{D}_H(\mathbf{l},\mathbf{k})$. In order to become a valid distance metric, histograms need to be normalized first:

$$\mathbf{k}^n = \frac{\mathbf{k}}{\sum_i^n k_i}\,. \tag{26}$$

For normalized histograms (total sum of 1), the histogram intersection is given by:

$$\mathcal{D}_H^n(\mathbf{k}^n,\mathbf{l}^n) = 1 - \sum_{i}^{n}|k_i^n - l_i^n|\,. \tag{27}$$

This is again the Minkowski-form distance metric with $k = 1$. Histogram intersection has the property that it allows for occlusion, i.e., when an object in one image is partly occluded, the visible part still contributes to the similarity.[4,36]

Alternative, histogram matching is defined by normalized cross correlation:

$$\mathcal{D}_x(\mathbf{k},\mathbf{l}) = \frac{\sum_{i=1}^{n}k_i l_i}{\sum_{i=1}^{n}k_i^2}\,. \tag{28}$$

The normalized cross correlation has a maximum of unity that occurs if and only if $\mathbf{k}$ matches exactly $\mathbf{l}$.

In the QBIC system,[16] the weighted Euclidean distance has been used for the similarity of color histograms. In fact, the distance measure is based on the correlation between histograms $\mathbf{k}$ and $\mathbf{l}$:

$$\mathcal{D}_Q(\mathbf{k},\mathbf{l}) = (k_i - l_i)^t A(k_i - l_j)\,. \tag{29}$$

Further, A is a weight matrix with term a_{ij} expressing the perceptual distance between bin i and j.

The average color distance has been proposed by Ref. 37 to obtain a simpler low-dimensional distance measure:

$$\mathcal{D}_{\text{Haf}}(\mathbf{k}, \mathbf{l}) = (k_{\text{avg}} - l_{\text{avg}})^t (k_{\text{avg}} - l_{\text{avg}}) \,, \tag{30}$$

where k_{avg} and l_{avg} are 3×1 average color vectors of $\mathbf{k}$ and $\mathbf{l}$.

As stated before, for broad domains, a proper similarity measure should be robust to object fragmentation, occlusion and clutter by the presence of other objects in the view. In Gevers and Smeulders,[19] various similarity function were compared for color-based histogram matching. From these results, it is concluded that retrieval accuracy of similarity functions depend on the presence of object clutter in the scene. The histogram cross correlation provide best retrieval accuracy without any object clutter (narrow domain). This is due to the fact that this similarity function is symmetric and can be interpreted as the number of pixels with the same values in the query image, which can be found present in the retrieved image and vice versa. This is a desirable property, when one object per image is recorded without any object clutter. In the presence of object clutter (broad domain), highest image retrieval accuracy is provided by the quadratic similarity function (e.g., histogram intersection). This is because this similarity measure counts number of similar hits and hence is insensitive to false positives.

In conclusion, for a search in broad domains, the quadratic similarity function (e.g., histogram intersection) is most appropriate. For narrow domains, without any object cluttering and occlusion, the cross correlation or Euclidean distance is most useful.

4.3. *Spatial color distributions*

In the previous section, all representation schemes do not include spatial or shape information. The lack of spatial information may yield lower retrieval accuracy. As for general image databases, a manual segmentation is not feasible due to the large amount of images, a simple approach is to divide images into smaller sub-images and then index them. This is known as fixed partitioning, which is discussed in Sec. 4.3.1. Other systems use a segmentation scheme, prior to the actual image search, to partition each image into regions. These region-based partitioning schemes will be discussed in Sec. 4.3.2. Nearly all region-based partitioning schemes use some kind of weak segmentation decomposing the image into coherent regions rather than objects (strong segmentation).

4.3.1. *Fixed partitioning*

The simplest way is to use fixed decomposition in which an image is partitioned into equally sized segments. The disadvantage of a fixed partitioning is that blocks usually do not correspond with the visual content of an image. For example, Gong,[27]

splits an image into nine equally sized sub-images, where each sub-region is represented by a color histogram. Guibas[38] segments the image by a quadtree, and Leung[39] uses a multi-resolution representation of each image. Sciascio[25] also uses a 4×4 grid to segment the image. Sebe[40] partitions images into three layers, where the first layer is the whole image, the second layer is a 3×3 grid and the third layer a 5×5 grid. A similar approach is proposed by Malki,[41] where three levels of a quadtree is used to decompose the images. Dimai[21] proposes the use of inter-hierarchical distances measuring the difference between color vectors of a region and its sub-segments. Chen and Wong[42] use an augmented color histogram capturing the spatial information between pixels together with the color distribution.

In Gevers,[36] the aim is to combine color and shape invariants for indexing and retrieving images. Color invariant edges are derived from which shape invariant features are computed. Then, computational methods are described to combine the color and shape invariants into a unified high-dimensional histogram for discriminatory object retrieval.

Huang *et al.*[43] propose the use of color correlograms for image retrieval. Color correlograms integrate the spatial information of colors by expressing the probability that a pixel of color c_i lies at a certain distance from a pixel of color c_j. It is shown that color correlograms are robust to a change in background, occlusion and scale (camera zoom). Cinque[44] introduces the spatial chromatic histograms, where for every pixel, the percentage of pixels having the same color is computed. Further, the spatial information is encoded by baricenter of the spatial distribution and the corresponding deviation.

4.3.2. *Region-based partitioning*

Segmentation is a computational method to assess the set of points in an image, which represent one object in the scene. Many different computational techniques exist, none of which is capable of handling any reasonable set of real world images. Segmentation is complicated as objects may be partially occluded from sight by the presence of other objects, or hard to distinguish in a surrounding of other objects. Segmentation is also complicated by the scene depending illumination conditions. However, in all cases, *weak segmentation* may be sufficient to recognize objects in a scene. Weak segmentation is to assess that a point set or a patch all correspond to one object but not reverse: not all points of the object are in the segmented set. Weak segmentation starts from the assumption that any point in the picture of an object may be invisible due to occlusion. In general, weak segmentation is based on finding connected regions with similar color feature distributions as the query object image. After segmentation, these regions are then used for the purpose of image retrieval.

In Ref. 45, a new image representation is proposed providing a transformation from the raw pixel data to a small set of image regions, which are coherent in color and texture space. This so-called Blobworld representation is based on

segmentation using the Expectation-Maximization algorithm on combined color and texture features.

In the Picasso system,[46] a competitive learning clustering algorithm is used to obtain a multiresolution representation of color regions. In this way, colors are represented in the $l^*u^*v^*$ space through a set of 128 reference colors as obtained by the clustering algorithm.

Gevers[47] proposes a method based on matching feature distributions derived from color ratio gradients. To cope with object cluttering, region-based texture segmentation is applied on the target images prior to the actual image retrieval process.

Colombo *et al.*[48] segment the image first into homogeneous regions by split and merge using a color distribution homogeneity condition. Then, histogram intersection is used to express the degree of similarity between pairs of image regions.

5. Color Image Retrieval

Having defined the features and representations of color images, we now turn our attention to the color retrieval process itself. This process can be decomposed into three major steps. In the *initialization* phase, the query space is instantiated and properly initialized. This is followed by a *specification* phase in which the user poses a certain query to the system. Finally, there is an output phase, where the system, in the ideal case, presents the user the expected result. For *interactive color retrieval*, there are two additional steps. After query specification, the effect of the query on query space is presented to the user in the *display* phase. At this moment, the true interaction takes place, when the user gives *feedback* on what is displayed on the screen. The feedback leads to an update of the display and the display/feedback loop continues till the user is satisfied and the final output can be generated. An overview of the whole process of (interactive) color retrieval can be seen in Fig. 3.

Each of the steps will now be considered in further detail.

5.1. *Query space initialization*

When starting a query session, the system should initialize all elements of the query space $Q = \{I_Q, F_Q, S_Q, Z_Q\}$.

To initialize the initial set of active images, a selection from the full set of database images has to be made. This is based on auxiliary information like the name of the archive, creation data of images and owner.

At that point, the set of color features and corresponding similarity function should be selected. The taxonomy of color spaces defined in Sec. 3 forms the basis for all features in the database. From the taxonomy, we can derive that the user has to specify:

- whether a narrow or broad domain is considered (to know if the device used can be assumed the same for all images)
- what kind of search is performed (object, target, category or associative)

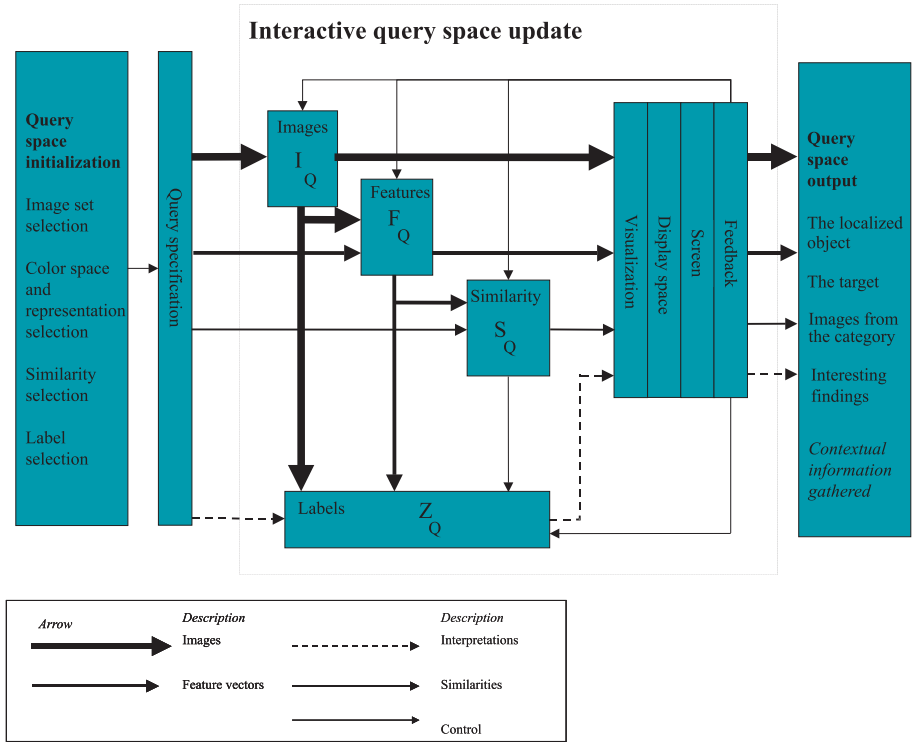

Fig. 3. Overview of the steps in interactive color retrieval and the role of interaction therein.

- which invariances are required (viewing direction, object geometry, illumination direction, illumination intensity)
- whether occlusion is expected to occur (to choose the appropriate color representation)
- whether there is an interest in the content of dark or white color regions (to see whether a linear color space is needed).

The initial query space Q^0 should not be biased towards specific images or make some image pairs *a priori* more similar than others. Therefore, most methods normalize the features of F_Q based on the distribution of the feature values over I_Q, e.g., Refs. 2 and 49. To make S_Q unbiased over F_Q, the parameters should be tuned, arriving at a *natural distance measure*. Such a measure can be obtained by normalization of the similarity between individual features to a fixed range.[49,50]

Recall that Z_Q is a set of labels, which can be assigned to the images. Due to uncertainties in interpretation, none of these labels can be assigned with full certainty. Therefore, each image to which a label is assigned should also store the probability associated with that particular label. Which labels to select depend on the user's goal. For target search, it is sufficient to have $Z_Q = \{\text{target}\}$. Indicating for each image how likely this image is indeed the target the user is looking for.

For category search, we have $Z_Q = \{\text{category} - \text{member}\}$ indicating whether this image is part of the category the user is looking for. Finally, for associative search, the proper label is $Z_Q = \{\text{interesting}\}$. However, as the user can be interested in any category, the set of labels can also be a set of semantic category names like car, house, romantic or beachscene, which in some context are of relevance to the user. If the user is allowed to annotate images with semantic labels while browsing, the probability of that label for the associated image can be set to 1.

5.2. *Query specification*

For specifying a query in Q, many different interaction methodologies have been proposed. They fall in one of two major categories:

- *exact queries*: a request for all images in I_Q satisfying a set of criteria
- *approximate queries*: a request for a ranking of the images in I_Q with respect to the query, based on S_Q.

Within each of the two categories, three subclasses can be defined depending on whether the queries relate to the

- *spatial content of the image*
- *the global image information*
- *groups of images.*

The queries based on spatial content require segmentation of the image. As described in Sec. 4, this requires strong segmentation, weak segmentation or fixed partioning.

For exact queries, the three subclasses are based on different predicates the result should satisfy:

- *exact query by spatial predicate* is based on the absolute or relative location of color regions.

Query on color region location is suited for the goal of finding an object. It assumes that the user knows what color the object has and where they are located. Implicit spatial relations between regions sketched by the user in Ref. 51 yield a pictorial predicate. This can be used to do target search. Other systems let the user explicitly define the predicate on relations between homogeneous regions.[52] In both cases, to be added to the query result, the homogeneous regions as extracted from the image must comply with the predicate.

- *exact query by image predicate* is a specification of predicates on global image descriptions, often in the form of range predicates.

Due to the semantic gap, range predicates on features are seldomly used in a direct way. In Ref. 53, ranges on color values are pre-defined in predicates like "MostlyBlue" and "SomeYellow". Learning from user annotations of a partioning

of the image allows for feature range queries like: "amount of skycolor > 50% and amount of sandcolor > 30%".[54] An alternative is to use a color picker and select in the color space visualized the color range of interest.[2,3] This requires an intuitive color space as otherwise specification is cumbersome and will always lead to unexpected results.

- *exact query by group predicate* is a query using an element $z \in Z_Q$, where Z_Q is a set of categories that partitions I_Q.

In both Refs. 55 and 56, the user queries on a hierarchical taxonomy of categories. The difference is that the categories are based on contextual information in Ref. 55 while they are interpretations of the content in Ref. 56.

In the following types of query specifications, the user specifies a single feature vector or one particular spatial configuration in F_Q. As a consequence, they are all approximate queries as no image will satisfy the query exactly.

- *approximate query by spatial example* results in an image or spatial structure corresponding to literal image values and their spatial relationships.

Pictorial specification of a spatial example requires a feature space such that feature values can be selected or sketched by the user. Color pickers[2,3] have been used with limited success, as users find it difficult to specify their needs in low-level features. A color display explicitly based on invariance might improve on this as only relevant color differences are shown. When weak segmentation of the query image and all images in I_Q is performed, the user can specify the query by indicating example regions.[5,52] Kato was the first to give the user the opportunity to make a sketch of the global composition of the image.[57] His method could easily be combined with a floodfill method to color the regions in the sketch. As a spatial example allows to specify the composition of the image searched for, the specification method is suited best for target search.

- *approximate query by image example* feeds the system a complete array of pixels and queries for images most similar to the example.

Most of the current systems have relied upon this form of querying[2,3] for all classes of search goals. The use of a single query image is, however, basically suited for object search only. Even then, it can only be successful if the systems explicitly considers the relevant invariance classes.[58] For other search goals, the use of one image cannot provide sufficient context for the query to select one of its many interpretations.[59] Query by example queries are in Ref. 50 subclassified into *query by external image example*, if the query image is an image, which is not in the database, versus *query by internal image example*. The difference in external and internal example is minor for the user, but affects the computational support as for internal examples, all relations between images can be pre-computed. For search goals other than object search, but based on image examples, one should use:

- *approximate query by group example* specification through a selection of images, which ensemble defines the goal.

As a set of images is used in query specification, this technique is particularly suited for category search. If the specification can be made specific enough, the method can also be used in target search. The rationale is to put the image in its proper semantic context to make one of the possible interpretations $z \in Z_Q$ preponderant. One option is that the user selects $m > 1$ images from a palette of images presented to find images best matching the common characteristics of the m images.[28] Such a query set is capable of more precisely defining the target and the admissible feature value variations therein. At the same time, a large query set nullifies the irrelevant variance in the query. This can be specified further by negative examples.[60,61] If for each group in the database, a small set of representative images can be found, it can be stored in a visual dictionary from which the user can create its query.[59]

Of course, the above queries can always be combined into more complex queries. For example, both Refs. 52 and 51 compare the similarity of regions using features and in addition they encode spatial relations between the regions in predicates.

Even with such complex queries, a single query is rarely sufficient to reach the goal except for object search. For most image queries, the user must engage in an active interaction with the system on the basis of the query results as displayed.

5.3. *Query space display*

The result of a query is a set of images, but in fact the query yields a new query space with a possibly new set of active images, new feature and similarity values, and new probabilities for the interpretations. Therefore, when considering the display, we do not restrict ourselves to the display of the set of images, but the display of all elements of query space.

Definition 2. The display space D is a space with perceived dimension d for visualization of query results.

Note that d is the intrinsic dimensionality of the query result or d is induced by the projection function if the query result is of too high a dimension to visualize directly. In both cases, d is not necessarily equal to the two dimensions of the screen.

When the query is exact, the result of the query is a set of images fulfilling the predicate. As an image either fulfills the predicate or not, there is no intrinsic order in the query result and $d = 0$ is sufficient. All images should be presented to the user as the system cannot decide by itself, which ones are appropriate for display.

For approximate queries, the images in I_Q are given a similarity ranking based on S_Q with respect to the query. As the size of I_Q is usually large, only the top most relevant images are selected for display. In spite of the 2D rectangular grid for presenting images that many systems[2,55] use, we should have $d = 1$. Although

developers are usually not aware of this, we do implicitly assume that reading order is part of the order of the images displayed. If the user refines its query, the images displayed do not have to be the images closest to the query. In Ref. 50, images are selected that together provide a representative overview of the whole active database. An alternative display model displays the image set minimizing the expected number of total iterations.[28]

As noted earlier, images are described by feature vectors. Backprojection[18,62] provides the user with an understanding of the associated color space and how the feature is located in the image. Every image has an associated position in feature space F_Q. The space spanned by the features is high dimensional. To give a view on query space, this high dimensional space should be mapped to a space suited for display. In both Refs. 59 and 63, F_Q is projected onto a display space with $d = 3$. Images are placed in such a way that distances between images in D reflect S_Q. To improve the user's comprehension of the query space,[63] provides the user with a dynamic view on F_Q through continuous variation of the 3D viewpoint. As an alternative to visualizing similarity through projection of the high dimensional feature space, the display can also be done directly on the similarity function. For this purpose the similarity of two images is viewed as the weight of a weighted graph on all image pairs with sufficient similarity. The problem is then reduced to visualizing the resulting graph.[64,65] For complexity reasons, only image pairs with sufficient similarity are taken into account.

The display in Ref. 66 allows for visualization of Z_Q. First, the images in I_Q are organized in 2D layers according to labels in Z_Q. Then, in each layer, images are positioned based on S_Q. Similar to the techniques described above.

5.4. *Interacting with query space*

In early systems, the above process of query specification and display of query result would be iterated, where in each step, the user would revise its query. For the user, it is far more convenient to give feedback on the results visualized, than going back to the query specification phase. Thus, in the course of the session, the system updates the query space, attempting to learn from the feedback the user gives on the relevance of the query result presented. The query specification is used only for initializing the display.

Definition 3. An interactive query session is a sequence of query spaces $\{Q^0, Q^1, \ldots, Q^{n-1}, Q^n\}$ such that Q^n bounds as close as possible what the user was searching for.

For each of the different search classes identified earlier, various ways of user feedback have been used. All are balancing between obtaining as much information from the user as possible and keeping the burden on the user minimal. The simplest form is to indicate, which images are relevant,[28] assuming "don't care" values for the others. In Refs. 60 and 61, the user in addition indicates nonrelevant

images. The system in Ref. 49 considers five levels of significance, which gives more information to the system, but makes the process more difficult for the user. When $d \geq 2$, the user can manipulate the projected distances between images, putting away nonrelevant images and bringing relevant images closer to each other.[59] The user can also explicitly bring in semantic information by annotating individual images, groups of images,[59] or regions inside images[54] with a semantic label.

The interaction of the user with the display thus yields a relevance feedback RF_i in every iteration i of the session. Combining this with Definitions 1 and 3, we have:

$$\{I_Q^i, F_Q^i, S_Q^i, Z_Q^i\} \xrightarrow{RF_i} \{I_Q^{i+1}, F_Q^{i+1}, S_Q^{i+1}, Z_Q^{i+1}\}\,. \tag{31}$$

Different ways of updating Q are possible as described now.

In Ref. 50, the displayed images correspond to a partitioning of I_Q. By selecting an image, one of the sets in the partition is selected and the set I_Q is reduced. Thus, the user zooms in on *a target or category*. The method follows the pattern:

$$I_Q^i \xrightarrow{RF_i} I_Q^{i+1}\,. \tag{32}$$

In many systems, the feature vectors in F_Q, corresponding to images in I_Q are fixed. When features are parameterized, and system feedback in the form of backprojection is given, feedback from the user could lead to optimization of the parameters. It corresponds to the pattern:

$$F_Q^i \xrightarrow{RF_i} F_Q^{i+1}\,. \tag{33}$$

For *associative search*, users typically need to interact to learn the system the right associations. Hence, the system should update the similarity function:

$$S_Q^i \xrightarrow{RF_i} S_Q^{i+1}\,. \tag{34}$$

In Refs. 49 and 60, S_Q is parameterized by a weight vector on the distances between individual features. The weights in Ref. 60 are updated by comparing the variance of a feature in the set of positive examples, to the variance in the union of positive and negative examples. If the variance for the positive examples is significantly smaller, it is likely that the feature is important to the user. The system in Ref. 49 first updates the weight of different feature classes. The ranking of images according to the overall similarity function is compared to the rankings corresponding to each individual feature class. Both positive and negative examples are used to compute the final weight. The weights for the different features in the feature class are taken as the inverse of the variance of the feature over positive examples.

The feedback RF_i in Ref. 59 leads to new user desired distances between some of the pairs of images in I_Q. The parameters of the continuous similarity function should be updated to match in optimal way the new distances. The optimization problem is ill-posed usually. A regularization term is introduced, which limits the departure from the natural distance function.

All of the methods below follow the pattern:

$$Z_Q^i \xrightarrow{RF_i} Z_Q^{i+1}. \tag{35}$$

For category and target search, a system can also refine the likelihood of particular interpretations. Either updating the label based on the features of images or on the similarity between images. The method in Ref. 61 falls in this class and considers *category search*. Images indicated by the user as relevant or nonrelevant in the current or previous iterations are collected. A Parzen estimator is incrementally constructed to find an optimal separation of the two classes.

In Ref. 28, an elaborate Bayesian framework is derived to compute the likelihood of any image in the database to be the target, given the history of actions RF_i. In each iteration, the user selects an image from the set of images displayed. The crucial step then is the update of the probability for each image in I_Q of being the target, given that among the displayed images, the user decided to make this explicit selection. In the reference, a sigmoidal shaped update function is used, expressed in the similarity between the selected image and the remaining images on display.

The system in Ref. 54 pre-computes a hierarchical grouping of images[a] based on the similarity for each individual feature. The feedback from the user is employed to create compound groupings corresponding to a user given $z \in Z_Q$. The compound groupings are such that they include all of the positive and none of the negative examples. Images that were not yet annotated falling in the compound group receive the label z. The update of probabilities P is based on different partionings of I_Q.

5.5. *Query output*

The final stage of the retrieval process is the output of the final result. This stage is reached whenever the user indicates that he has reached the goal, or is bored of trying to find it. When the search has been successful and depending on the search goal, the following outputs should be generated:

- *object search*: the localized object in the images that contain the object.
- *target search*: the image in the active image set for which p(target) is highest.
- *category search*: the image in the active image set for which p(category-member) is highest and all its images, which have high similarity to this image.
- *associative search*: the set of images in the active image set for which p(interest) is highest and all images, which have high similarity to these images.

In addition, the system can display all contextual information gathered in the search process, e.g., statistics on the search process.

[a]In fact, Ref. 54 is based on a fixed partitioning rather than on images. It does, however, equally apply to whole images.

6. Conclusion

We have made a concise analysis of methodologies for interactive retrieval of color images. Two issues are of importance. First, the domain, which can be *broad* or *narrow*. Second, the search method, which can be *object search*, *target search*, *category search* or *associative search*.

To choose the proper color space based on the domain and search method categorization, we conclude that xyz is well suited for object and target search for broad domains (device-independent) recorded under varying imaging conditions under the constraint of a specific light source (color invariant but not color constant). Color systems $U^*V^*W^*$ and $L^*a^*b^*$ are particularly suited for category and associative search in broad-domains. The IHS system is well suited for color query specification. Further, the hue color feature H is a good candidate for object search (color invariance but not color constant) of colorful objects. For broad domain image datasets such the Internet, where no constraints can be applied on the imaging conditions, a color constant space should be used such as the N, F and M.

To choose the proper similarity function using histogram based methods, it is concluded that for a search in broad domains, the quadratic similarity function (e.g., histogram intersection) is most appropriate. For narrow domains, without any object cluttering and occlusion, the cross correlation or Euclidean distance is most useful.

The full search process is an interactive refinement of query space based on user interaction. To that end, the visualization of the query result based on the six categories of query specification needs great attention as otherwise it is difficult for the user to interactively proceed the search. As query space is composed of images, features, similarity and interpretations, the system must select which ones to update in the query process. We conclude that the appropriate query specification method has an immediate relation with the search goal of the user. The same holds for the different feedback and query update paradigms.

In conclusion, we have given complete guidelines for choosing and designing methods for interactive color image retrieval based on the domain and search goal characteristics.

References

1. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, *IEEE Trans. PAMI* **22**(12), 1349 (2000).
2. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Comput.* (1995).
3. A. Gupta and R. Jain, *Commun. ACM* **40**(5), 71 (1997).
4. Th. Gevers and A. W. M. Smeulders, *Pattern Recognition* **32**, 453 (1999).
5. J. R. Kender, "Saturation, hue, and normalized colors: Calculation, digitization effects, and use," Technical report, Department of Computer Science, Carnegie-Mellon University, 1976.

6. H. M. G. Stokman and Th. Gevers, *J. Electronic Imaging* **10**(1), 221 (2001).
7. D. Forsyth, *Int. J. Comput. Vision* **5**, 5 (1990).
8. B. V. Funt and M. S. Drew, in *Computer Vision and Pattern Recognition* (1988), pp. 544–549.
9. E. H. Land, *Scientific American* **218**(6), 108 (1977).
10. G. Healey and D. Slater, *J. Opt. Soc. Am.* **11**(11), 3003 (1995).
11. G. D. Finlayson, S. S. Chatterjee, and B. V. Funt, in *ECCV96* (1996), pp. 16–27.
12. D. Slater and G. Healey, "The illumination-invariant recognition of 3D objects using local color invariants," *IEEE Trans. PAMI* **18**(2) (1996).
13. B. V. Funt and G. D. Finlayson, *IEEE Trans. PAMI* **17**(5), 522 (1995).
14. S. K. Nayar and R. M. Bolle, *Int. J. Comput. Vision* **17**(3), 219 (1996).
15. G. D. Finlayson, M. S. Drew, and B. V. Funt, *JOSA* **11**, 1553 (1994).
16. M. Flicker *et al.*, "Query by image and video content: The QBIC system," *IEEE Comput.* **28**(9) (1995).
17. A. Pentland, R. W. Picard, and S. Sclaroff, in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases II*, Vol. 2 (1995), pp. 259–270.
18. M. J. Swain and D. H. Ballard, *Int. J. Comput. Vision* **7**(1), 11 (1991).
19. Th. Gevers and A. W. M. Smeulders, "Content-based image retrieval by viewpoint-invariant image indexing," *Image and Vision Comput.* **7**(17) (1999).
20. S. Sclaroff, L. Taycher, and M. La Cascia, "Imagerover: A content-based image browser for the world wide web," in *IEEE Workshop on Content-based Access and Video Libraries* (1997).
21. A. Dimai, in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases IV* (1997), pp. 352–360.
22. J. R. Smith and S.-F. Chang, "Visualseek: A fully automated content-based image query system," in *ACM Multimedia* (1996).
23. D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos, *Image Understanding* **75**(1–2), 46 (1999).
24. T. F. Syeda-Mahmood, *Int. J. Comput. Vision* **21**(1), 9 (1997).
25. E. Di Sciascio, G. Mingolla, and M. Mongielle, in *VISUAL99* (1999), pp. 123–130.
26. S. Servetto, Y. Rui, K. Ramchandran, and T. S. Huang, *J. VLSI Signal Processing Syst.* **20**(2), 137 (1998).
27. Y. Gong, C. H. Chuan, and G. Xiaoyi, *Multimedia Tools Appl.* **2**, 133 (1996).
28. I. J. Cox, M. L. Miller, T. P. Minka, and T. V. Papathomas, *IEEE Trans. Image Processing* **9**(1), 20 (2000).
29. G. Ciocca and R. Schettini, *Information Processing and Management* **35**, 605 (1999).
30. M. A. Stricker and M. Orengo, "Similarity of color images," in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases IV* (1996).
31. Y. J. Zhang, Z. W. Liu, and Y. He, in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases IV* (1996), pp. 371–382.
32. A. R. Appas, A. M. Darwish, A. I. El-Desouki, and S. I. Shaheen, in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases VII* (1999), pp. 492–500.
33. C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying," in *Comput. Graphics* (1995).
34. A. Vellaikal and C. C. J. Kuo, in *Digital Image Storage Archiving Systems* (1995), pp. 312–323.

35. K. C. Liang and C. C. J. Kuo, in *IEEE International Conference on Image Processing*, Vol. 1 (1997), pp. 572–575.
36. Th. Gevers and A. W. M. Smeulders, *IEEE Trans. Image Processing* **9**(1), 102 (2000).
37. J. Hafner, H. S. Sawhney, W. Equit, M. Flickner, and W. Niblack, *IEEE Trans. PAMI* **17**(7), 729 (1995).
38. L. J. Guibas, B. Rogoff, and C. Tomasi, in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases III* (1995), pp. 352–362.
39. K.-S. Leung and R. Ng, in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases VI* (1998), pp. 259–270.
40. N. Sebe, M. S. Lew, and D. P. Huijsmands, "Multi-scale sub-image search," in *ACM International Conference on Multimedia* (1999).
41. J. Malki, N. Boujemaa, C. Nastar, and A. Winter, in *International Conference on Visual Information Systems, VISUAL99* (1999), pp. 115–122.
42. Y. Chen and E. K. Wong, in *IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases VII* (1999), pp. 423–429.
43. J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Ramin, in *Computer Vision and Pattern Recognition* (1997), pp. 762–768.
44. L. Cinque, S. Levialdi, and A. Pellicano, in *IEEE Multimedia Systems*, Vol. 2 (1999), pp. 969–973.
45. S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color- and texture-based image segmentation using em and its application to content-based image retrieval," in *Sixth International Conference on Computer Vision* (1998).
46. A. Del Bimbo, M. Mugnaini, P. Pala, and F. Turco, *Pattern Recognition* **31**(9), 1241 (1998).
47. Th. Gevers, P. Vreman, and J. van der Weijer, "Color constant texture segmentation," in *IS&T/SPIE Symposium on Electronic Imaging: Internet Imaging I* (2000).
48. C. Colombo, A. Rizzi, and I. Genovesi, "Histogram families for color-based retrieval in image databases," in *Proc. ICIAP'97* (1997).
49. Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Video Tech.* (1998).
50. J. Vendrig, M. Worring, and A. W. M. Smeulders, "Filter image browsing: Exploiting interaction in retrieval," in *Proceedings of Visual Information and Information Systems*, eds. D. P. Huijsmans and A. W. M. Smeulders, *Lecture Notes in Computer Science*, Vol. 1614 (1999).
51. J. R. Smith and S.-F. Chang, *Multimedia Systems* **7**(2), 129 (1999).
52. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-based image querying," in *Proceedings of the IEEE International Workshop on Content-Based Access of Image and Video Databases* (1997).
53. V. E. Ogle, *IEEE Comput.* **28**(9), 40 (1995).
54. T. P. Minka and R. W. Picard, *Pattern Recognition* **30**(4), 565 (1997).
55. J. Smith and S.-F. Chang, *IEEE Multimedia* **4**(3), 12 (1997).
56. A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang, "Content-based hierarchical classification of vacation images," in *IEEE International Conference on Multimedia Computing and Systems* (1999).
57. T. Kato, T. Kurita, N. Otsu, and K. Hirata, in *Proceedings of the ICPR, Computer Vision and Applications, The Hague* (1992), pp. 530–533.

58. Th. Gevers and A. W. M. Smeulders, *IEEE Trans. Image Processing* **9**(1), 102 (2000).
59. S. Santini, A. Gupta, and R. Jain, "Emergent semantics through interaction in image databases," *IEEE Trans. Knowledge Data Eng.*, in press.
60. G. Ciocca and R Schettini, in *Proceedings of Visual Information and Information Systems* (1999), pp. 107–114.
61. C. Meilhac and C. Nastar, in *IEEE International Conference on Multimedia Computing and Systems* (1999), pp. 512–517.
62. J. Vendrig and M. Worring, in *Advances in Visual Information Systems*, ed. R. Laurini, *Lecture Notes in Computer Science*, No. 1929 (2000), pp. 338–348.
63. A. Hiroike, Y. Musha, A. Sugimoto, and Y. Mori, in *Proceedings of Visual 99, International Conference on Visual Information Systems*, eds. D. P. Huijsmans and A. W. M. Smeulders, *Lecture Notes in Computer Science*, Vol. 1614 (1999), pp. 155–162.
64. I. Herman, G. Melancon, and S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Trans. Visualization Comput. Graphics* **6**(1) (2000).
65. R. van Liere, W. de Leeuw, and F. Waes, "Interactive visualization of multidimensional feature spaces," in *Proceedings Workshop on New Paradigms for Information Visualization* (2000).
66. T. Kakimoto and Y. Kambayashi, *Int. J. Digital Libraries* **2**, 68 (1999).

Marcel Worring received his Masters (honors) and Doctoral, both in Computer Science, respectively from the Free University Amsterdam ('88), and the University of Amsterdam ('93), The Netherlands. He currently is an Assistant Professor at the University of Amsterdam. His interests are in multimedia informedia analysis. He is Project Leader in a large project covering knowledge engineering, language processing, image and video analysis, and information space interaction, conducted in close relation with industry. In 1998, he was a Visiting Research Fellow at the University of California, San Diego. He has published over 50 scientific publications and serves on the program commitee of several conferences.

Theo Gevers is Assistant Professor of Computer Science at the University of Amsterdam, The Netherlands. His main research interests are in the fundamentals of image and video databases, viusal retrieval by content, and color image processing. He has led several (inter)national projects and acts as a Reviewer. He is co-organizer of the First International Workshop on Image Databases and Multi Media Search and the Third International Conference on Visual Information Systems. Further, he is member of the program committee of CVPR, SPIE, ICPR, ICIP and others. He has published over 50 papers on color image processing, content-based image and video retrieval and image database design.