The MPEG-7 Color Descriptors

Jens-Rainer Ohm (RWTH Aachen, Institute of Communications Engineering)
Leszek Cieplinski (Mitsubishi Electric ITE-VIL)
Heon Jun Kim (MI Group, Information Technology Lab., LGE)
Santhana Krishnamachari (Video Communications, Philips Research)
B. S. Manjunath (University California Santa Barbara)
Dean S. Messing (Information Systems Technologies Dept., Sharp Labs of America)
Akio Yamada (Multimedia Research Laboratories, NEC Corp.)

1 Introduction

Color is an important visual attribute for both human vision and computer processing. This chapter provides an overview of MPEG-7 color descriptors. As is the case with the other descriptors, extraction of these descriptors and their use in similarity matching are outside the scope of the normative components of the standard. Nevertheless, efficient extraction and matching techniques are indispensable for a practical system. For each of the descriptors, we provide details of its syntax (and associated semantics, as applicable), descriptor computation (extraction), and experimental results on retrieval effectiveness on a controlled dataset with known ground truth. Even though not explicitly stated, all the color descriptors discussed in the following can be computed from arbitrarily shaped image regions as well.

Various factors influenced the selection of these color descriptors. These include:
(a) their ability to characterize the perceptual color similarity, judged by performance of the descriptors in matching images and video segments based on color characteristics
(b) low complexity of the associated extraction and matching techniques, as MPEG-7 systems must be able to handle search and retrieval tasks over large multimedia databases, or may be small, portable devices with limited computational power
(c) the sizes of the coded descriptions, which play an important role in indexing, and in transmission of the descriptors over bandwidth limited networks
(d) the scalability and interoperability of the descriptors.

To evaluate the retrieval performance of various color descriptors, experiments based on the *query by example* paradigm were conducted. To perform these experiments, a Common Color Dataset (CCD) consisting of about 5000 images, and a set of 50 Common Color Queries (CCQ), each with specified ground truth images, was defined [4]. This means that the number of queries was about 1% of the number of images in the database, which guarantees sufficient statistical significance of the results. The CCD consists of a variety of still images, images from stock photo galleries, screen shots of television programs and animations. The query and corresponding ground truth images in CCQ are manually established through a process of visual inspection by different groups of participants. The effectiveness of the individual descriptors is evaluated using the averaged normalized retrieval accuracy measure (ANMRR) (see Appendix). The final selection of the descriptors was mainly based on their retrieval effectiveness on the CCD as well as their overall complexity.

We now briefly summarize the color descriptors that were defined as part of the standard. The following sections will describe the individual descriptors in more detail.

- The **Color Space Descriptor** allows a selection of a color space to be used in the description. The associated **Color Quantization Descriptor** specifies the partitioning of the given color space into discrete bins. These two descriptors are used in conjunction with other color descriptors.
- The **Dominant Color Descriptor** allows specification of a small number of dominant color values as well as their statistical properties like distribution and variance. Its purpose is to provide an effective, compact and intuitive representation of colors present in a region or image.
- The **Scalable Color Descriptor** is derived from a color histogram defined in the Hue-Saturation-Value (HSV) color space with fixed color space quantization. It uses a Haar transform coefficient encoding, allowing scalable representation of description, as well as complexity scalability of feature extraction and matching procedures.
- The **Group of Frames/Group of Pictures Descriptor** is an extension of the scalable color descriptor to a group of frames in a video or a collection of pictures. This descriptor is based on aggregating the color properties of the individual images or video frames.
- The **Color Structure Descriptor** is also based on color histograms, but aims at identifying localized color distributions using a small structuring window. To guarantee interoperability, the color structure descriptor is bound to the Hue-Min-Max-Difference (HMMD) color space (see Section 2.2).
- The **Color Layout Descriptor** captures the spatial layout of the dominant colors on a grid superimposed on a region or image. Representation is based on coefficients of the Discrete Cosine Transform (DCT). This is a very compact descriptor being highly efficient in fast browsing and search applications. It can be applied to still images as well as to video segments.

The main for defining a standardized description is interoperability. This aspect has been extensively investigated in core experiments [6], [7], after which it was concluded to constrain the possible variations in the descriptions. In the context of color descriptors, this has led to specifying a unique choice of color space for each of the color descriptors (with the exception of the dominant color descriptor), as leaving this specification up to the user would have deteriorated the retrieval efficiency and raised complexity of the matching process. As a consequence, the set of histogram-derived descriptors was strictly limited for two variants – Scalable Color and Color Structure – with fixed definition of Color Spaces and limited, interoperable set of bin-quantization choices.

2 Color Spaces

The Color Space Descriptor specifies a selection of a color space to be used in another color descriptor, specifically, the Dominant Color Descriptor. The color spaces specified in the MPEG-7 are– RGB, YCbCr, HSV, HMMD, Monochrome, and Linear transformation matrix with reference to RGB. In addition, a flag is provided to indicate reference to a color primary and mapping to a standard reference white value.

The Color Space Descriptor defines the color components as continuous-value entities. For discrete representation, a quantization is necessary. The Color Quantization Descriptor specifies the number of quantization levels for each color component in the color space. A uniform quantization in each of the color components in a given color space is assumed. The only exception is the HMMD color space, the quantization for which is described in detail in Section 2.2.1.

The RGB color space is one of the most popular color models. This space is defined as the unit cube in the Cartesian coordinate system. The YCbCr is a legacy color space of the precedent MPEG standards, MPEG-1/2/4. It is defined by a linear transformation of RGB color space as follows:

```
Y =   0.299*R + 0.587*G + 0.114*B
Cb = -0.169*R - 0.331*G + 0.500*B
Cr =  0.500*R - 0.419*G - 0.081*B
```

For the Monochrome color representation, Y component alone in the YCrCb is used.

2.1 HSV Color Space

The HSV color space defined as a cylinder (see Figure 2.1) consists of Hue, Saturation and Value. Hue (H) represented by the angle from 0 to 360 degrees specifies one color family from another, as red from yellow, green, blue or purple. Saturation (=[0,1]) specifies how pure a color is; pure red, yellow, green, blue and so on. Value (=[0,1]) specifies how bright or dark a color is. The three components are expressed by a non-linear transform of the three components of RGB color space as shown below:

```
Max = max(R, G, B);
Min = min(R, G, B);
Value = Max;
if( Max == 0 ) then
  Saturation = 0;
else
  Saturation = (Max-Min)/Max;
if( Max == Min ) Hue=0; /* It is achromatic color */
otherwise:
if( Max == R && G >= B )
  Hue = 60*(G-B)/(Max-Min)
else if( Max == R && G < B )
  Hue = 360 + 60*(G-B)/(Max-Min)
else if( G == Max )
  Hue = 60*(2.0 + (B-R)/(Max-Min))
else
  Hue = 60*(4.0 + (R-G)/(Max-Min))
```

When Max value is equal to Min value (Saturation = 0), it is an achromatic color (white, black or gray). In this case, Hue is set to 0 degree (means red).

The HSV color space is the color space associated with the scalable color histogram and the group of frames histogram descriptors (see Section 4 and Section 5). For these two descriptors, the HSV space is uniformly quantized into 256 bins--16 levels in H, 4 levels in S, and 4 levels in V. These two descriptors can also be computed using fewer than 256 histogram bins. Table 1 summarizes the partitioning of the HSV space into 128, 64, and 32 bins and the corresponding number of coefficients used in the Scalable Color and Group of Frames descriptors. See Section 4 and Section 5 for more details.

2.2 HMMD Color Space

The HMMD (Hue-Max-Min-Diff) color space is closer to a perceptually uniform color space. The double cone shape confines this color space as shown in Figure 2.2. The component names, "Max", "Min" and "Diff" are according to the following transform equations between RGB and HMMD:

```
Max = max(R, G, B);
Min = min(R, G, B);
Diff  =  Max – Min;
```

Even though the four components are identified in the name of the HMMD color space, one more component, Sum can be defined.

```
Sum = (Max+Min) /2;
```

Therefore, a total of five components are identified in this color space. However, a set of three components, {H, Max, Min} or {H, Diff, Sum}, is sufficient to form the HMMD color space and specify a color point. The semantics of each component is distinct and described as follows. Hue (H = [0°,360°]) has the same property as Hue in the HSV color space. Max (=[0,1]) specifies how much black color is present, giving the flavor of shade or blackness. Max has the same RGB related transform as Value in HSV but the valid sub-space is different in HMMD. Thus, the interpretation is different from Value. Min (=[0,1]) specifies how much white color is present, giving the flavor of tint or whiteness. Diff (=[0,1]) specifies how much a color is close to pure colors, giving the flavor of tone or colorfulness. It has a similar property as Saturation in HSV but the valid sub-space is again different. Finally, Sum (=[0,1]) specifies the brightness of the color.

2.2.1 HMMD Color Space Quantisation

This subsection describes the non-uniform quantisation of the HMMD color space used by the Color Structure Descriptor.

As already discussed, the HMMD space can be defined in three dimensions using, *sum*- and *diff*-axes as well as *hue angle* as in Figure **2**. A three-dimensional quantisation of such a space corresponds to a partition[1] of the space into 3-D cells. Four non-uniform quantisations of HMMD are defined in the MPEG-7 Standard. The four quantisations partition the space into 256, 128, 64, and 32 cells, respectively.

Comment [dsm1]: Hue angle shd. be added to Heon Jun’s HMMD color space figure.

Each 3-D quantisation is defined via five *subspaces* of HMMD as follows. The *diff*-axis, itself defined on the interval [0, 255], is cut into five sub-intervals: [0,6), [6, 20), [20, 60), [60, 110) and [110, 255]. This 1-D partition of the *diff*-axis implicitly defines five subspaces numbered 0, 1,…, 4, respectively. Each subspace is that subset of HMMD where *sum* and *hue* are allowed to take all values in their respective ranges, and where *diff* is restricted to one of the five intervals.

A partition of HMMD is obtained by partitioning the ranges of *hue* and *sum* into *uniform* intervals within each subspaces according to Table 4. The table actually consists of four tables (one for each quantisation of HMMD) the columns of which are alternately white or shaded. For each quantisation the table tells how to partition the subspaces to yield the overall partition.

For example, to partition HMMD space into 128 cells the table instructs us to partition Subspace 4 by cutting *hue* into 8 uniform intervals and *sum* into 4 uniform intervals, giving 32 cells in Subspace 4. The other four subspaces are partitioned in like manner to yield the overall *non-uniform* quantisation of HMMD into 128 cells.

[1] A *partition* of a space, S, is a collection, $\{p_1,\ldots,p_N\}$, of cells such that $p_i \subseteq S$, $p_i \cap p_j = \varnothing$ for $i \neq j$, and $\bigcup_{i=1}^{N} p_i = S$.

We note that for the 32-cell quantisation the dividing line between subspaces 1 and 2 is missing in the table. This indicates that the two subspaces have been united into a single subspace. Thus the 32-cell partition of HMMD is defined via four subspaces instead of five.

Figure 3 depicts a slice of HMMD color space in the *diff-sum* plane for zero hue angle and shows the cells for the 128-cell quantisation. Subspace boundaries are indicated in the figure by vertical lines in the plane. The *diff*-axis values that determine the subspace boundaries are shown in black at the top of the dashed cut-point markers along the upper edge of the plane.

Horizontal lines within each subspace depict the division of the *sum*-axis into uniform intervals. The grey rotation arrows around each cut-point marker indicate the partition of *hue angle*. The grey number to the right of a rotation arrow corresponds to the number of intervals into which the range of *hue* has been partitioned in the subspace to the right of the cut-point. For example, Figure 3 states that the range *hue* associated with the subspace between *diff* = 60 and *diff* = 110 (i.e. subspace 3) is divided into 8 equal intervals. This agrees with the entry in Table 4.

Finally, Figure 3 indicates the scheme for numbering the cells in a partition of HMMD space. This is important because of the 1-to-1 association between cells and Color Structure Descriptor bin indices discussed in Section 6.

Besides the five color spaces described up to now, the color space descriptor can describe any 3 by 3 color transform matrix which specifies the linear transformation between RGB and the respective color space. Thus, any linear transformation from the RGB color space can be specified in the MPEG-7 color space descriptor.

3 Dominant Color Descriptor

The Dominant Color Descriptor (DCD) provides a compact description of the representative colors in an image or image region. Its main target applications are similarity retrieval in image databases and browsing of image databases based on single or several color values. Unlike the traditional histogram based descriptors, the representative colors are computed from each image instead of being fixed in the color space, thus allowing the color representation to be accurate and compact.. The DCD allows for efficient indexing [2]of large databases as presented in [14].

The dominant color descriptor is defined to be

$$F = \{\{c_i, p_i, v_i\}, s\}, (i = 1, 2, ..., N)$$

most where N is the number of dominant colors. Each dominant color value c_i is a vector of corresponding color space component values (for example, a 3-D vector in the RGB color

[2] Indexing in the database context refers to an efficient pruning of the search space so as to minimize the number of distance computations and disk I/O accesses needed for computing the nearest neighbors of a given query feature vector. The feature vector dimensions of typical visual descriptors are quite large. For example, the number of bins in a histogtram descriptor may be of the order of few hundreds. It is a well known fact that nearest neighbor search for similarity retrieval in such high dimensional spaces is quite expensive and is often referred to as the *dimensionality curse* in the database literature.

space). The percentage p_i (normalized to a value between 0 and 1) is the fraction of pixels in the image or image region corresponding to color c_i, and $\sum_i p_i = 1$. The optional color variance v_i describes the variation of the color values of the pixels in a cluster around the corresponding representative color. The spatial coherency s is a single number that represents the overall spatial homogeneity of the dominant colors in the image. The number of dominant colors N can vary from image to image and a maximum of eight dominant colors was found to be sufficient to represent an image or an image region . The color space quantization depends on the color space specifications defined for the entire database and need not be specified with each descriptor.

The binary syntax of the dominant color descriptor specifies 3 bits to represent the number of dominant colors and 5 bits for each of the percentage values (uniform quantization of [0,1]). The Color Space and Color Quantization descriptors are referred to by this descriptor and RGB is the default color space. The optional color variances are encoded at 3 bits per color with non-uniform quantization. The table below summarizes the binary syntax of the DCD. See [1] for detailed specifications.

Field	**Number of Bits**	**Meaning**
NumberofColors	3	Specifies number of dominant colors.
SpatialCoherency	5	Spatial Coherency value.
Percentage[ ]	5	Noralized percentage associated with each dominant color.
ColorVariance[ ] [ ]	1	Color variance of each dominant color.
Index[ ][ ]	1-12	Dominant color values

3.1 EXTRACTION

The extraction procedure described in [12] for the dominant color uses the Generalized Lloyd Algorithm [13] to cluster the pixel color values. It is recommended that the clustering be performed in a perceptually uniform color space such as the CIE LUV. The distortion D_i in the i-th cluster is given

$$D_i = \sum_n h(n) \left\| \mathbf{x}(n) - c_i \right\|^2, \quad x(n) \in C_i,$$

where c_i the centroid of cluster Ci, x(n) is the color vector at pixel n, and h(n) is the perceptual weight for pixel n. The perceptual weights are calculated from local pixel statistics to account for the fact that human visual perception is more sensitive to changes in smooth regions than in textured regions [15] The update rule for the above distortion metric can be derived to be:

$$c_i = \frac{\sum h(n) x(n)}{\sum h(n)}, \quad x(n) \in C$$

The procedure is initialized with one cluster consisting of all pixels and one representative color computed as the centroid (center of mass) of the cluster. The algorithm then follows a sequence of centroid calculation and clustering steps until a stopping criterion (minimum distortion or maximum number of iterations) is met. The clusters with highest distortion are divided by adding perturbation vectors to the centroids until the maximum distortion falls below a predefined threshold or the maximum number of clusters is generated. The percentage or fraction of pixel in the image belonging to each of the quantized colors is then calculated and these resulting percentages are uniformly quantized to 5 bits. The color values are quantized according to the specifications of the color space and the associated color quantization descriptors.

A simple connected component analysis is performed to identify groups of pixels of the same dominant color that are spatially connected. Four connectivity (the four nearest neighbors of a pixels) is assumed. The normalized average number of connecting pixels of each dominant color is then computed. A 3x3 masking window is used for this purpose. This is used as a measure of spatial coherency for that dominant color. The overall spatial coherence is then a linear combination of the individual spatial coherence values with the corresponding percentages p_i being the weights. The spatial coherence value is then non-uniformly quantized to 5 bits, where 31 means highest confidence and 1 means no confidence. The value 0 is used for cases where it is not computed. Finally, the color variances are computed as variances of the pixel values within each cluster and non-uniformly quantized to 1 bit per color component.

3.2 Similarity Matching

Each object or region in the database is represented using the dominant color descriptor as defined above. Typically, 3-4 colors provide a good characterization of the region colors. Given a query image, similarity retrieval involves searching the database for similar color distributions as the input query. Since the number of representative colors is small, one can first search the database for each of the representative colors separately, and then combine the results. Searching for individual colors can be done very efficiently in a 3-D color space. Consider two dominant color descriptors,

$F_1 = \{\{c_{1i}, p_{1i}, v_{1i}\}, s_1\}, \quad (i = 1, 2, \cdots, N_1)$ and

$F_2 = \{\{c_{2i}, p_{2i}, v_{2i}\}, s_2\}, \quad (i = 1, 2, \cdots, N_2)$.

Ignoring the optional variance parameter and the spatial coherence, the dissimilarity *D(F₁, F₂)* between the two descriptors can be computed as:

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2a_{1i,2j} p_{1i} p_{2j}$$

where the subscripts 1 and 2 in all variables stand for descriptions F_1 and F_2 respectively, and $a_{k,l}$ is the similarity coefficient between two colors c_k and c_l,

$$a_{k,l} = \begin{cases} 1 - d_{k,l} / d_{\max} & d_{k,l} \le T_d \\ 0 & d_{k,l} > T_d \end{cases}$$

where $d_{k,l} = \|c_k - c_l\|$ is the Euclidean distance between two colors c_k and c_l, T_d is the maximum distance for two colors to be considered similar, and $d_{max} = \alpha T_d$. In particular, this means that any two dominant colors from one single description are at least T_d distance apart. A recommended value for T_d is between 10-20 in the CIE-LUV color space and for α is between 1.0-1.5. The above dissimilarity measure can be shown to be equivalent to the quadratic distance measure that is commonly used in comparing two color histogram descriptors.

One variation of the above distance is to use the spatial coherence field. For example, in the MPEG-7 experiments the following distance was used:

$$D_S = w_1 abs(s_1 - s_2) D + w_2 D$$

where s_1 and s_2 are the spatial coherencies of the query and target descriptors, and w_1 and w_2 are fixed weights, with recommended settings to 0.3 and 0.7, respectively.

This distance can be modified to take into account the optional variance. If the color variance field is present, the matching function is based on modeling of the color distribution as a mixture of Gaussian distributions with parameters defined as color values and color variance [14]. Calculation of the squared difference between the query and target distributions then leads to the following formula for the matching function:

$$D_V = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} p_{1i} p_{1j} f_{1i\,1j} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} p_{2i} p_{2j} f_{2i\,2j} - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2 p_{1i} p_{2j} f_{1i\,2j},$$

where

$$f_{xi\,yj} = \frac{1}{2\pi\sqrt{v_{xi\,yj}^{(l)} v_{xi\,yj}^{(u)} v_{xi\,yj}^{(v)}}} \exp\left[-\left(\frac{c_{xi\,yj}^{(l)}}{v_{xi\,yj}^{(l)}} + \frac{c_{xi\,yj}^{(u)}}{v_{xi\,yj}^{(u)}} + \frac{c_{xi\,yj}^{(v)}}{v_{xi\,yj}^{(v)}} \right) / 2 \right]$$

and $$c_{xi\,yj}^{(l)} = (c_{xi}^{(l)} - c_{yj}^{(l)})^2, \; \mathrm{v}_{xi\,yj}^{(l)} = (\mathrm{v}_{xi}^{(l)} + \mathrm{v}_{yj}^{(l)}).$$

In the equations above, $c_{xi}^{(l)}$ and $v_{xi}^{(l)}$ are dominant color values and color variances, *x,y* index the query and target descriptors, *i,j* index the descriptor components and *l, u* and *v* the components of the color space.

3.3 Experimental Results:

Table 2 shows a comparison of ANMRR results for two average descriptor sizes using the CCD/CCQ. It can be seen that reasonable results are obtained even for the basic version of the descriptor and a significant improvement can be achieved by using the optional fields. Table 3 gives results using the spatial variance parameter and comparing with the DC descriptor (without variance). These experiments suggest that using 5 bits for the spatial coherence field is a reasonable trade-off between complexity and effectiveness of the descriptor. It should be noted that one of the main objectives of the dominant color descriptor is to provide a compact and intuitive representation of salient colors in a given region of interest.

4 Scalable Color Descriptor

The Scalable Color Descriptor (SCD) can be interpreted as a Haar transform based encoding scheme applied across values of a color histogram in the HSV color space (see Section 2.1). The histogram values are extracted, normalized and non-linearly mapped into a 4-bit integer representation, giving higher significance to small values. The Haar transform is applied to the 4-bit integer values across the histogram bins. The basic unit of the transform consists of a sum operation and a difference operation (see Figure 4 (a)), which relate to primitive low pass and high pass filters. Summing pairs of adjacent bins is equivalent to the calculation of a histogram with half number of bins. From the sums of every two adjacent Hue bin values out of the 256-bin histogram , we get a representation of a 128-bin histogram with 8 levels in H, 4 levels in S and 4 levels in V. If this process is repeated, the resulting 64, 32 or 16 sum coefficients from the Haar representation are equivalent to histograms with 64, 32 or 16 bins. Table 1 shows the equivalent partitioning of the HSV color space for different number of coefficients of the Haar transform. If an application does not require the full resolution, limited number of Haar coefficients may simply be extracted from a 128, 64 or 32 bin histogram; this would still guarantee interoperability with another representation where all coefficients were extracted, but only to the precision of the coefficients that are available in both of the representations. Note that since all partitions in the original color space quantization are powers of 2, the combination with the Haar transform appears to be very natural.

The high pass (difference) coefficients of the Haar transform express the information contained in finer-resolution levels (with higher number of bins). Histograms of natural image signals usually exhibit high redundancy between adjacent histogram bins. This can be explained by the "impurity" (slight variation) of colors caused by variable illumination and shadowing effects. Hence, it can be expected that the high pass coefficients expressing differences between adjacent histogram bins usually have only small values. Exploiting this property, it is possible to truncate the high pass coefficients to an integer representation with only a small number of bits.

4.1 Extraction and Matching

Figure 4b shows the block diagram of the of the SCD extraction process. The output representation is scalable in terms of numbers of bins, by varying the number of coefficients used. Interoperability between different resolution levels is retained due to the scaling property of the Haar transform. Thus, matching based on the information from subsets of coefficients guarantees an approximation of the similarity in full resolution. Furthermore, as mentioned above, also the feature extraction operation can be scaled to lower levels (less bins in the source histogram).

Besides the scalability in the number of histogram bins, another form of scalability is achieved by scaling the quantized (integer) representation of the coefficients to different numbers of bits. The "difference" coefficients in the Haar transform can take either positive or negative values. The sign part is always retained whereas the magnitude part can be scaled by skipping the least significant bits. Using the sign-bit only (1 bit / coefficient) leads to an extremely compact representation, while good retrieval efficiency is retained. At the highest-accuracy level, 1-8 bits are defined for integer representations of the magnitude part, depending on the relevance of the respective coefficients. Between these extremes, it is possible to scale to different resolution levels. For example, consider a set of five coefficients whose magnitudes are encoded using 8,4,7,3, and 7 bits, respectively, as shown in Figure 5. If the lowest 3 bits are discarded in the scalable bit representation, only 5,1,4,0, and 4 bits remain to encode the absolute value.

ℓ_1-norm based matching (sum of absolute differences) can be applied in the Haar transform domain; however, results are not identical with ℓ_1-norm based matching in the histogram domain. In the case where only the sign bit is used (all bit planes representing the absolute value discarded), the ℓ_1-norm degenerates to a Hamming distance, allowing very low complexity in the distance calculation.

4.2 Representation

The scalability in the number of histogram bins and the number of bit planes are represented by the fields *NumberofCoefficients* and *NumberofBitplanesDiscarded.* The *NumberofCoefficients* is used to indicate whether 16, 32, 64, 128 or 256 bins (coefficients) are used. The *NumberofBitplanesDiscarded* specifies the number of bitplanes of the coefficients that are discarded, ranging from 0 to 8. In the case this value is 8, the magnitude of the coefficients are not present, only the sign of each coefficient is retained which is represented by the *CoefficientSign*. The magnitudes of the coefficients are represented in a bit-plane fashion, which means that the most significant bits of all coefficients are taken first, followed by the next most significant, etc. The bit plane representation allows the transmission of only a certain amount of most significant bits for bandwidth constrained applications. The representation is as follows:

Field	**Number of Bits**	**Meaning**
NumberofCoefficients	3	Specifies the number of histogram bins = 16,32,64,128,256
NumberofBitplanesDiscarded	3	Specifies discarding 0 to 8 bitplanes
CoefficientSign[ ]	NumberofCoefficients	The sign of each coefficient
BitPlane[ ][ ]	See text	Coefficient magnitudes represented in a bitplane fashion

4.3 Experimental Results

Retrieval results achieved by the SCD are shown in Figure 6. In addition, the ANMRR quality measure was calculated from matching in the histogram domain, after performing an inverse Haar transform. The results show that a reasonable performance can be achieved even with small numbers of bits, while the performance saturates between 256 and 512 bits.

5 Group of Frame/Group of Picture Descriptor

The Group-of-frame/Group-of-picture (GoF/GoP) color descriptor is used for the joint representation of color-based features for multiple images or multiple frames in a video segment. This descriptor can be used to represent a collection of contiguous or non-contiguous video frames or a group of images. Traditionally for a group of frames or pictures, a key-frame or a key-image is selected from such a group, and the color-related features of the entire collection are represented by those of the chosen sample. Such methods are highly dependent on the quality of the representative sample selection, and may lead to unreliable results. The GoF/GoP color descriptors are histogram-based descriptors that reliably capture the color content of multiple images or video frames.

5.1 Extraction and Matching

GoF/GoP color descriptors are obtained by aggregating the histograms of multiple images or video frames and representing the aggregated histograms using the SCD. The individual image or video frame histogram is computed based on the uniform quantization of the HSV color space as detailed in Table 1. Three different ways are defined to compute the aggregate color histogram values for the whole series of images or video frames: average, median or intersection aggregation. The aggregated histogram is then input to the Haar transform to build the SCD representation as presented in the previous section.

The average histogram is computed by accumulating the frame/picture histograms in the group and subsequently normalizing each accumulated bin value by N, where N is the number of frames in the GoF or the number of images in the GoP. The average histogram is simple to compute. The descriptor can be updated easily if additional images or video frames are added to the group. A potential problem with using sample averages to compute the GoF/GoP histogram is the sensitivity of the mean operator to outliers. The median histogram is obtained by constructing, for each bin, the ascending list of N frame/picture histogram values over the length of the GoF/GoP, and assigning the median of this list to the corresponding bin in the GoF/GoP histogram. The median histogram eliminates aberrant effects such as lighting changes, occlusion, text overlays, etc., which the average histogram is vulnerable to. One concern regarding the use of the median histogram is the increased computational complexity. The intersection histogram (Int_Histogram) is obtained by computing for each bin the minimum value over all the N frame/picture histograms in the group. Each bin value in the intersection histogram thus represents the number of pixels of a particular color that appear in all of the GoF frames. The intersection histogram is characteristically different from the average and median histograms, in that it provides the “least common” color traits of the given GoF/GoP, rather than an estimate of the color distribution.

The matching for the GoF/GoP descriptor is performed exactly similar to the SCD descriptor. It should be ensured that the aggregation method used for the descriptors that are being matched are the same.

5.2 Descriptor Representation

As alluded to before, the GoF/GoP color descriptor is an extension of the SCD. The representation for the GoF/GoP descriptor is identical to the SCD with an additional attribute *aggregation.* The three different possible methods of aggregation is represented by this attribute using two bits. This is followed by the associated SCD descriptor.

Field	**Number of Bits**	**Meaning**
aggregation	2	Specifies the three different types of aggregation
ScalableColorDescriptor	See Section 4.2	Specifies the SCD

5.3 Experimental Results

The joint representation of a collection of video frames or images obtained using the GoF/GoP color descriptor can be used in different applications in video content management, namely, query-by-example based retrieval applications, shot grouping, fast search and browsing of a image or video databases. A description of the experimental results for video segment matching is presented here. The experiments were conducted on about 3 hours of video in the MPEG-7 data set. The data set contained various types of video, including sports programs, news clips, videos of natural scenes etc. Shot segmentation was performed on this data using cut, dissolve, fade and wipe detection resulting in 1544 shots (groups-of-frames). For each GoF, histograms were computed for the individual frames and the three different types of aggregation were performed to obtain the GoF descriptor. In addition, for each GoF, a key frame was identified to compare the performance GoF descriptor based search against the key-frame based search. The key frame was selected by searching through all the frames in a GoF to find the optimal frame that had the minimum mean absolute error with all the other frames within that GoF. From the 1544 GoFs, 31 queries were selected and for each query a set of ground truth GoFs were manually identified [10]. The queries ranged from almost static GoFs to dynamic scenes with edit effects and transitions. Table 5 shows the ANMRR retrieval results using the average and median aggregation based GoF matching against the key-frame based matching. From the table, it can be seen that the GoF descriptor performs better than the optimum key frame based matching. Results of another application to find if a given frame belongs to a GoF using the intersection aggregation can be found in [9]. Experimental results on the use of GoP descriptor for fast search of image databases and content-based browsing can be found in [8], [11].

6 Color Structure Descriptor (CSD)

The *Color Structure Descriptor* (CSD) represents an image by both the color distribution of the image (similar to a color histogram) and the local spatial structure of the color. The additional information about color structure makes the descriptor sensitive to certain image features to which the color histogram is blind. Figure 7 illustrates this with a pair of images each of which consists of two *iso-color planes*[3], one grey and one black. The grey iso-color plane on the left is highly structured whereas the one on the right is less so. The *structure* of an iso-color plane is the degree to which its pixels are clumped together relative to the scale of an associated structuring element.

Each image contains exactly 50 pixels in its grey plane and 250 pixels in its black plane. Hence they are indistinguishable based solely on the information in their two-bin color histograms. But their two-bin CS Descriptors are very different and thus the images can be easily distinguished in an indexing or retrieval application based on the CSD.

The CSD is identical in form to a color histogram but is semantically different. Specifically, the CSD is a one-dimensional array of 8-bit quantised values,

$$\mathrm{CSD} = \bar{h}_s(m), \quad m \in \{1, \ldots, M\} ,$$

where M is chosen from the set {256, 128, 64, 32} and where s is the scale of the associated square structuring element. In the example of

, $s = 3^2$. The M bins (array elements) of $\bar{h}_s$ are associated in an injective[4] manner to the M cells of the non-uniformly quantised HMMD color space (see Section 2).

[3] An image quantised to N colors is composed of N iso-color planes. The n-th plane is the set of all pixels having the n-th quantised color, $n \in \{1,\ldots, N\}$.

[4] A map, $f : A \rightarrow B$, is said to be *injective* if f maps set A onto set B in a one-to-one manner.

6.1 CSD Interoperability

Descriptor interoperability was discussed generally in Section 1. There is, however, an aspect of interoperability that is peculiar to the CSD. In retrieval applications it may be the case that a query descriptor presented (e.g., via the Web) to a remote search engine has a length that differs from the descriptors in the database. In order to compute the similarity between query and database descriptors the lengths must be equalised.

Now in the case of color histograms a length N histogram can be obtained either

- by extracting it directly from an image quantised to N colors or
- by extracting from the same image, quantised to $M > N$ colors, a length M histogram and then unifying (summing) appropriate subsets of its bin values to form the N-bin histogram.

Either method results in the *same* histogram as long as a *scalability condition* is met by the quantised color space as discussed in [16].

The CSD does not enjoy this property because the color quantisation of an image affects its color structure. The reader is directed to [16] for a discussion of the somewhat subtle reason for this. The salient point is that a CS Descriptor obtained from image I by one scheme will, in general, lead to different retrieval results than a CSD from I by the other scheme. That is, the two extraction / re-sizing methods are not interoperable.

Consequently, and in contrast to most other MPEG-7 visual descriptors, extraction and re-sizing of the CSD is a *normative* process within the standard, by which we mean that the major steps are specified by the standard. Deviation from these steps risks breaking the interoperability of the descriptor.

6.2 Extraction

The CSD is best understood in terms of the *Color Structure Histogram*, h_s, upon which $\overline{h}_s$ is based. Extraction of a CSD is a three-step process:

i. A 256-bin CS Histogram is extracted (i.e., accumulated) from an image represented in the 256 cell quantised HMMD color space. If the image is in another color space then it must be converted to HMMD and re-quantised prior to extraction.

ii. If $N < 256$ is desired then bins are *unified* to obtain a N-bin CS Histogram.

iii. The values (amplitudes) of each of the N bins are non-linearly quantised in accordance with the statistics of color occurrence in typical consumer imagery.

We now discuss these steps in more detail, a full description of which are given in [1] and [2].

6.2.1 Accumulation of CS Histogram

In the context of the CSD, the length and color space of the CS Histogram are fixed. Outside this context, however, the CS Histogram can, in general, be of any length and can be accumulated from an image represented in any quantised color space. The procedure is depicted in Figure 8 where a simple five-color “image” is shown together with a 4×4 structuring element. Also shown in tabular form on the right is an 8-bin CS Histogram, $h_s(m)$, whose bins are associated with 8 quantised colors, c_m, $m \in \{1,\ldots,8\}$, in which the image is represented.

In nominal operation, the structuring element scans the image such that

- the element visits every position in the pixel grid, and
- the element always lies entirely within the image.

At each position the CS Histogram is updated based on the colors present within the element. The operation is illustrated in Figure 8 where, in its current position, 4 colors are present within the structuring element. Therefore, each of the four corresponding bins of the CS Histogram is incremented by one. Observe that in any given position, the increase in $h(m)$ is determined by whether color c_m is present or absent within the element rather than by how much of c_m is enclosed. Hence the final value of $h(m)$ is determined (up to normalisation) by the number of positions at which the structuring element contains c_m.

It is interesting to note that the CS Histogram may be viewed as a *generalised* color histogram since it reduces precisely to an ordinary color histogram when a 1×1 structuring element is used.

Although a 4×4 element is shown in Figure 8, the MPEG-7 Standard defines the scale to be 8×8. This was determined by experiment to be the optimal scale. In conjunction with this, the Standard calls for images that deviate from a nominal size to be uniformly subsampled, both horizontally and vertically, in order to reduce the computational load. The subsampling factor is given by $K = 2^p$ where

$$p = \max\{0, \lfloor \log_2 \sqrt{W \cdot H} - 7.5 \rfloor\},$$

where W and H are the picture width and height respectively, and where $\lfloor \cdot \rfloor$ is the *floor* operator. The reader is directed to [1] and [2] for an equivalent formulation where the accumulation requires no explicit sub-sampling of the image.

The CS Histogram (and hence the CSD) can be extracted from arbitrarily shaped, possibly disconnected, regions of an image. This is done in practice by means of a binary mask that defines the regions. Movement of the structuring element is as above (i.e., over the entire extent of the image) but the histogram is accumulated only with pixels that lie in the transparent portions of the mask.

6.2.2 Bin Unification

When a CSD of length $N \in \{128, 64, 32\}$ is required, the 256-bin CS Histogram is reduced in length by bin unification. This process adds the values in each of N disjoint subsets of bins from the full-length histogram to form the N bins of the shorter histogram.

We now describe the procedure for a general size reduction from M to $N < M$ bins. For the case at hand one merely lets $M = 256$ and $N \in \{128, 64, 32\}$. Let $P = \{p_1, \ldots, p_M\}$ and $Q = \{q_1, \ldots, q_N\}$ be two scalable quantisations of a color space, S, where the p_m and q_n are the individual cells of the two quantisations and where $M > N$. Quantisation *scalability* is equivalent to the conditions:

$$\bigcup_{m=1}^{M} p_m = S = \bigcup_{n=1}^{N} q_m,$$

$$\text{for each } n,\ q_n = \bigcup_{m \in J_n} p_m,\ \text{ where } J_n = \{n_1, \ldots, n_{k_n}\},$$

$$J_i \cap J_j = \varnothing \ \text{ for } \ i \neq j.$$

The first condition insures that both *P* and *Q* cover the space, *S*. The second condition implicitly defines the index subsets, J_n. The third condition is a consequence of the fact that quantisation cells are, themselves, disjoint. Hence it is redundant, following from the second condition. We include it for clarity.

In light of the bijection between bins and color space cells, the bin unification is defined by

$$h_s^N(n) = \sum_{m \in J_n} h_s^M(m), \quad n \in \{1, \ldots, N\}, \tag{1}$$

where the superscripts denote the respective histogram lengths. The index subsets, J_n, for reducing a length 256 CS Histogram to a shorter length can be derived from the four scalable quantisations of the HMMD color space defined in Section 2.2.

6.2.3 Bin Value Quantisation

The final step in extracting a *N*-bin CSD is to normalise to the range [0, 1] the bin values (amplitudes) of the *N*-bin CS Histogram from the preceding step, and then to non-linearly quantise the normalised values to 8-bits according to the quantisation table in [1]. The non-linear quantisation was derived using several heuristics as well as experiments conducted with the Comon Color Dataset and dramatically increases the retrieval accuracy of the CSD. The chief effect of the non-linearity is to give the small values greater weight in the Similarity Measure than they would otherwise have.

6.3 CSD Re-sizing

The extraction procedure of Section 6.2 insures that lengths, say *N* and *M >N*, of two different length CS Descriptors can always be equalised. The re-sizing procedure adjusts the longer descriptor to match that of the shorter. First the bin values must be de-quantised so that *linear* values participate in the bin unification. Next the *M* bins are unified just as discussed for the general case in Section 6.2.2. Finally bin values of the new *N*-bin histogram are non-linearly re-quantised to obtain the desired *N*-bin CS Descriptor. It can be shown that this re-sizing process gives the same result as having extracted an *N*-bin CS D in the first place.

6.4 Retrieval Results

As with other histogram descriptors, the CSD uses the ℓ_1-norm for matching in its Similarity Measure. The Common Color Dataset was modified by the addition of a few more query images to further differentiate the retrieval performance between the CSD and Scalable Color Descriptors. Table 6(a) shows CS Descriptor retrieval accuracy for the four lengths defined by the standard. The longest descriptors yield the best results.

To motive the choice of the non-uniformly quantised HMMD color space, Table 6(b) lists the retrieval results in the case where the CS histograms were extracted in the HSV color space followed by non-linear bin value quantisation. The (uniform) color space quantisation of the HSV space for each descriptor length is shown in the 2nd column of Table 2. A comparison of the results in the two tables clearly shows the performance gained by using the non-uniformly quantised HMMD color space.

7 Color Layout Descriptor

The Color Layout Descriptor (CLD) is a very compact and resolution-invariant representation of color for high-speed image retrieval. It is designed to efficiently represent spatial distribution of colors. This feature can be used for wide variety of similarity-based retrieval, content filtering, and visualization. It is especially useful for spatial-structure based retrieval applications, for example, sketch based retrieval and video segment identification. The sketch-based retrieval is considered to be a very important functionality since it can offer very user-friendly interfaces, especially when the search is fast enough.

The functionalities of this descriptor are image-to-image matching and video-clip-to-video-clip matching, and sketch to image/video-clip matching.. Description of the color layout can also be achieved using the Grid Layout data type of MPEG-7 and the Dominant Color Descriptor. However, this combination would require a relatively large number of bits, and matching will be more complex and expensive. CLD provides more precise and faster retrieval using more compact description.

7.1 Extraction

This descriptor is obtained by applying the DCT transformation on a two dimensional array of local representative colors in Y/Cb/Cr color space. Figure 9 illustrates the extraction process of the descriptor from an image. It consists of four stages, image partitioning, representative color detection, DCT transformation, and non-linear quantization of the zigzag-scanned coefficients. In the first stage, an input picture is divided into 64 blocks to guarantee the resolution or scale invariance. In the next stage, a single dominant color is selected from each block. Any method to select representative color can be applied, but it is recommended to use the average of pixel colors as the representative color since it is most simple and the description accuracy is enough in general. The selection results in a tiny image icon of size 8x8. In the third stage, each of the three color-components is transformed by 8x8 DCT, so three sets of 64 DCT-coefficients are obtained. They are zigzag scanned and the first few coefficients are non-linearly quantized (using 64 and 32 levels for DC and AC coefficients, respectively). The standard allows scalable representation of the feature by controlling the number of enclosed coefficients. It is recommended to use a total of 12 coefficients, 6 for luminance and 3 for each chrominance, for most of the images. However, another option to use a total 18 coefficients (6 for both luminance and chrominance) can also be considered to apply this descriptor for high-quality still pictures. The total bit-length of the recommended descriptor (12 coefficients) is just 64 bits including one signaling bit, which specifies the extension of the number of coefficients. It should be noted that this descriptor is one of the more compact descriptors in the MPEG-7/Visual and is quite suitable for applications having limitations on storage and/or bandwidth

7.1.1 Representation

The number of DCT coefficients used in the CLD is variable and is represented by the CoefficientPattern field. The CoefficientPattern field can take three possible values. The first value indicates the use of six DCT coefficients for luminance and three each for chrominance, the second values indicates the use of six coefficients for both luminance and chrominance. For the third value of the CoefficientPattern , the number of DCT coefficients are represented by the NumberofYCoeff and NumberofCCoeff fields. The possible number of coefficients is one of 3, 6, 10, 15, 21, 28, and 64. The actual values of the coefficients are represented by the arrays Ycoeff, CbCoeff and CrCoeff. The lengths of each of these is either five or six bits depending on the coefficient.

Field	Number of Bits	Meaning

CoefficientPattern	1-2	Specifies the number DCT coefficients
NumberofYCoeff	3	Number of DCT coefficients for the luminance
NumberofCCoeff	3	Number of DCT coefficients for the chrominance
YCoeff[ ]	5-6	The DCT coefficients values for the luminance
CbCoeff[]	5-6	The DCT coefficients values for the chrominance
CrCoeff[ ]	5-6	The DCT coefficients values for the chrominance

7.2 Matching

This descriptor is applicable both to an image as a whole and any parts of an image with arbitrary shapes. On applying to an arbitrary shaped region, the representative color selection should be performed using only valid pixel values, and a padding process is required before the DCT transform. Representative colors of grid blocks containing no valid pixels are substituted with the average color of all valid pixels in the image.

For matching two CLDs, {*DY, DCr, DCb*} amd {*DY*', *DCr*', *DCb'*}, the following distance measure can be used :

$$D = \sqrt{\sum_i w_{yi}\left(DY_i - DY_i'\right)^2} + \sqrt{\sum_i w_{bi}\left(DCb_i - DCb_i'\right)^2} + \sqrt{\sum_i w_{ri}\left(DCr_i - DCr_i'\right)^2}$$

Here, the subscript *i* represents the zigzag-scanning order of the coefficients. The perceptual characteristic of human vision system could be included for similarity calculation since the feature description is in frequency domain. The distances should be weighted appropriately, with larger weights given to the lower frequency components, to match the characteristic. Since the complexity of the similarity matching process shown above is low (about 110 clocks with Intel SSE instruction set), super high-speed image matching can be achieved.

7.3 Experimental Results

Figure 10 shows the retrieval efficiency of this descriptor evaluated using the Common Color Dataset and Queries. These results demonstrate that the color layout descriptor is quite effective in image retrieval in spite of its compact size. The retrieval efficiency is compared with a traditional approach (GRC) wherein the image is partitioned and representative colors for each partition is used to represent the layout feature. The results in Figure 10 indicate that CLD achieves a much superior performance than GRC.

Video-clip retrieval is one of the more promising applications of the Color Layout Descriptor [17]. It requires repetitive use of the matching computation, so very fast matching is necessary to obtain the retrievals in a reasonable time. A temporal series of CLDs can be used to implement this functionality. The similarity between video-clips is obtained by averaging the distances between corresponding frames of the video clips to be matched.

Summary

Within the Visual part of MPEG-7, a set of color descriptors has been defined that are able to capture the important aspects of color feature, allowing color similarity computations. These descriptors are compact, hence allowing efficient description of color properties. Extensive effort, based on fruitful cooperation of a large group of people, has been spent to achieve optimized solutions. It can be expected that the MPEG-7 color descriptors will be extremely useful in those applications that are based on color similarity judgment.

References

[1] ISO/IEC/JTC1/SC29/WG11 : "Text of ISO/IEC 15938-3 Multimedia Content Description Interface – Part 3 : Visual. Final Committee Draft", document no. N4062, Singapore, March 2001. [LESZEK]

[2] ISO/IEC/JTC1/SC29/WG11 : "MPEG-7 Visual Experimentation Model (XM), Version 10.0", document no. N4063, Singapore, March 2001. [LESZEK]

[3] W.K. Pratt : "Digital Image Processing", second edition, Wiley 1991

[4] D. Zier, J. -R. Ohm : "Common Datasets and Queries in MPEG-7 Color Core Experiments ", ISO/IEC JTC1/SC29/WG11 (MPEG) document no. M5060, Melbourne, October 1999. [pdf file enclosed]

[5] P. Ndjiki-Nya, J. Restat, T. Meiers, J. -R. Ohm, A. Seyferth, R. Sniehotta : "Subjective Evaluation of the MPEG-7 Retrieval Accuracy Measure (ANMRR)", ISO/IEC JTC1/SC29/WG11 (MPEG) document no. M6029, Geneva, May 2000. [pdf file enclosed]

[6] H. J. Kim, J. E. Lee : "CE Result of CT1 : Interoperability between color histogram descriptors using different color spaces and quantization methods", ISO/IEC JTC1/SC29/WG11 (MPEG) document no. M5744, Noordwijkerhout, March 2000. [pdf file enclosed]

[7] J.-R. Ohm, B. Makai : "Results of CE CT1 on interoperability of different color histograms", ISO/IEC JTC1/SC29/WG11 (MPEG) document no. M5755, Noordwijkerhout, March 2000. [pdf file enclosed]

[8] S. Krishnamachari and M. Abdel-Mottaleb, "Hierarchical clustering for fast image retrieval", SPIE Proceedings, *Proc. Storage and Retrieval for Image and Video Databases VIII*, pp 427-435, Jan. 1999.

[9] A. Mufit Ferman, S. Krishnamachari, A. Murat Tekalp, M. Abdel-Mottaleb, and R. Mehrotra, "Group-of-Frame/Picture Color Histogram Descriptors for Multimedia Applications", Proc. of the IEEE Intl. Conf. On Image Processing (ICIP'2000), vol. 1, pp. 65-68, Vancouver, Canada, September 2000.

[10] A. Mufit Ferman, S. Krishnamachari, A. Murat Tekalp, M. Abdel-Mottaleb, and R. Mehrotra, "Core Experiment on Group-of-Frames/Pictures Histogram Descriptors (CT7)", ISO/IEC JTC1/SC29/WG11 (MPEG) document no. M5124, Melbourne, October 1999. [pdf file enclosed]

[11] S. Krishnamachari and M. Abdel-Mottaleb, "Image Browsing using Hierarchical Clustering", Proceedings of the Fourth IEEE Symposium on Computers and Communications, ISCC'1999. Red Sea, Egypt, July 1999, pp. 301-307.

[12] Y. Deng, B.S. Manjunath, C. Kenney, M.S. Moore and H. Shin, "An Efficient Color Representation for Image Retrieval", *IEEE Transactions on Image Processing*, vol. 10 (1) January 2001, pp. 140-147.

[13] A. Gersho and R.M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, 1993.

[14] L. Cieplinski, "Results of Core Experiment CT4 on Dominant Color Extension", ISO/IEC JTC1/SC29/WG11 (MPEG) document no. M5775, Nordwijkerhout, March 2000. [LESZEK]

[15] C. Kenney, Y. Deng, B. S. Manjunath, and G. Hewer, "Peer group image enhancement," *IEEE Transactions on Image Processing*, vol. 10 (2), February 2001, pp. 326-334.

[16] D. S. Messing, P. van Beek, and J. Errico, "The MPEG-7 Colour Structure Descriptor: Image Description Using Colour and Spatial Information," in *IEEE Proc. Int'l Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001. (PDF FILE ENCLOSED)

Comment [BSM2]: Page: 1
Eventually this will be a reference to the IS ,right?

Comment [BSM3]: Page: 1
Ref to CD content as this will be included on the cd/dvd.

Formatted: Bullets and Numbering

Comment [BSM4]: Page: 1
Same comment for the MPEG doc

[17] E.Kasutani and A.Yamada, "THE MPEG-7 COLOR LAYOUT DESCRIPTOR: A COMPACT IMAGE FEATURE DESCRIPTION FOR HIGH-SPEED IMAGE/VIDEO SEGMENT RETRIEVAL", Proc of Int.Conf. on Image Processing 2001, Oct.2001.

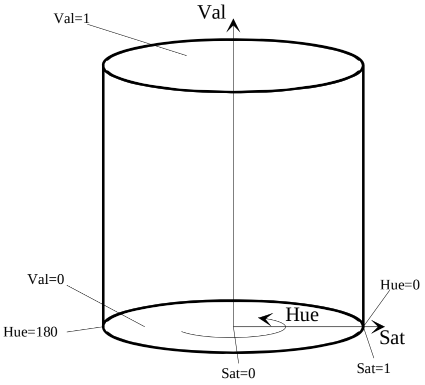


Figure 1. HSV color space.

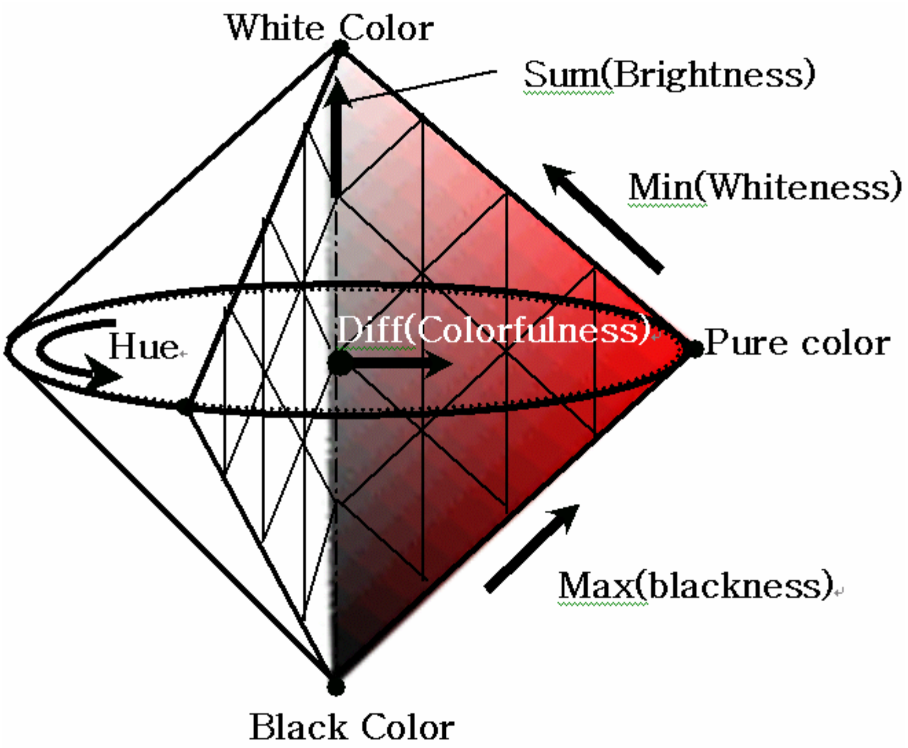


Figure 2. **Double cone representation of the HMMD color space.**

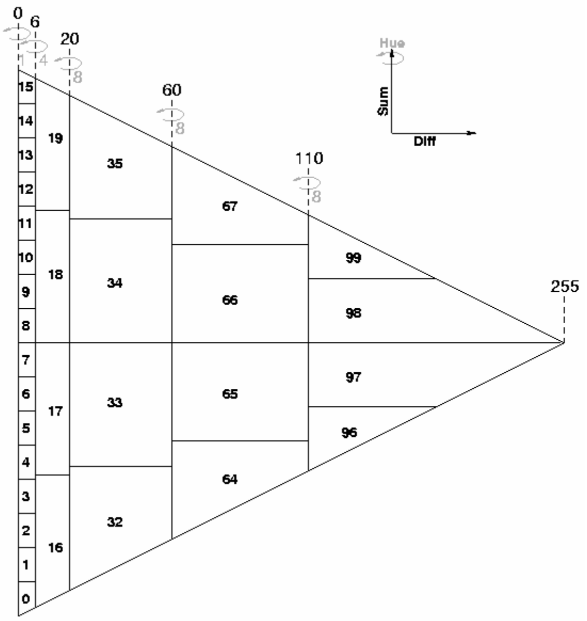


Figure 3: A slice of 128-cell quantised HMMD color space at *hue* = 0 the indexing scheme used to number the cells.

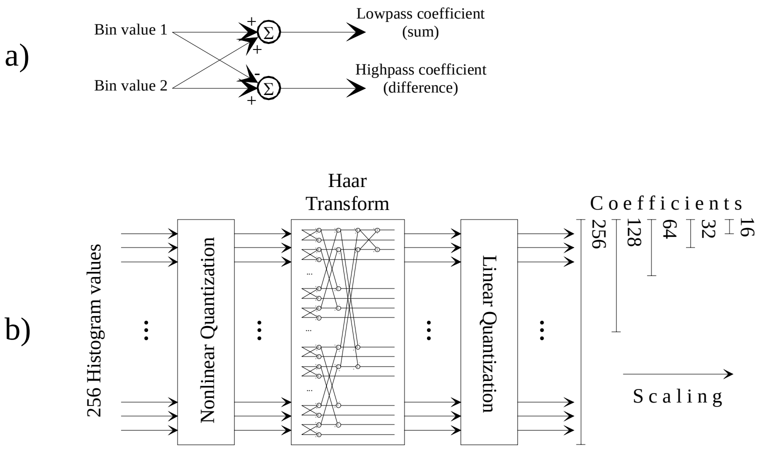


Figure 4. a Basic unit of Haar transform b A schematic diagram of Scalable Color Descriptor generation.

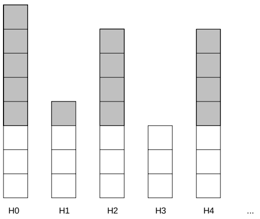


Figure 5. Illustration of bit plane scalability.

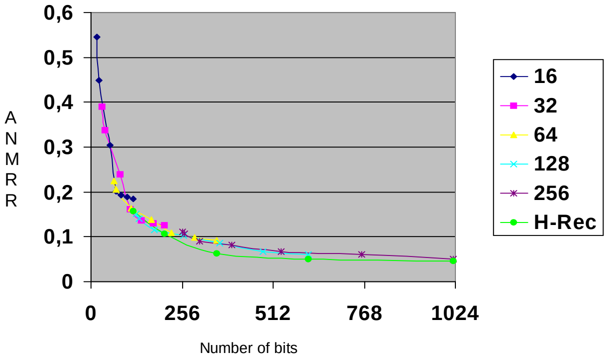


Figure 6. Retrieval results with different numbers of Haar coefficients (16-256) quantized at different numbers of bits. H-Rec signifies retrieval results after reconstruction of histogram from Haar coefficients at full bit resolution, which constitutes a lower limit of the efficiency curve.

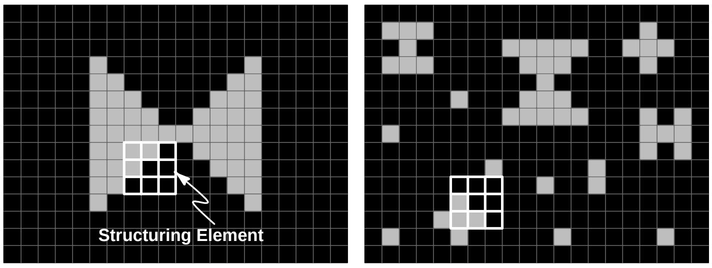
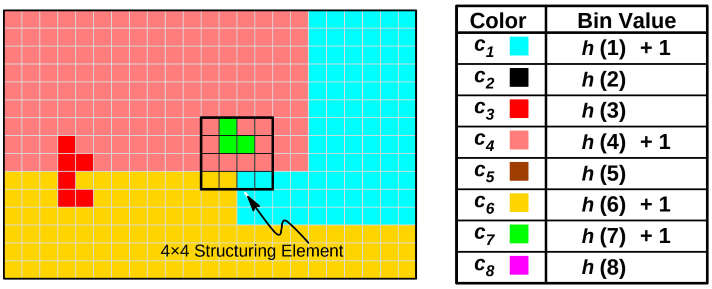


Figure 7. Two iso-color planes with differing amounts of structure.



Color	Bin Value
c_1 🟦	h (1) + 1
c_2 ⬛	h (2)
c_3 🟥	h (3)
c_4 ▨	h (4) + 1
c_5 🟫	h (5)
c_6 🟨	h (6) + 1
c_7 🟩	h (7) + 1
c_8 🟪	h (8)

Figure 8. Accumulation of Color Structure Histogram

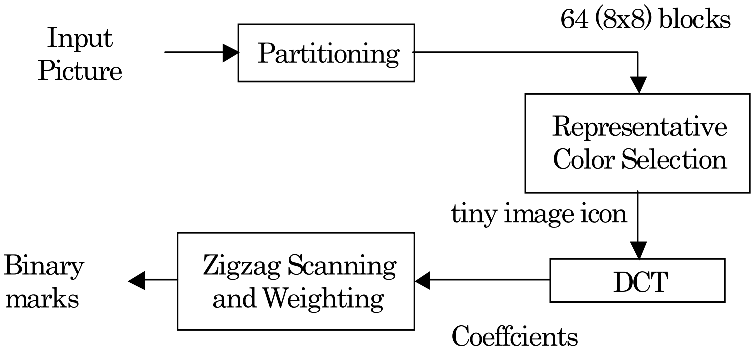


Figure 9. The extraction process of the Color Layout descriptor.

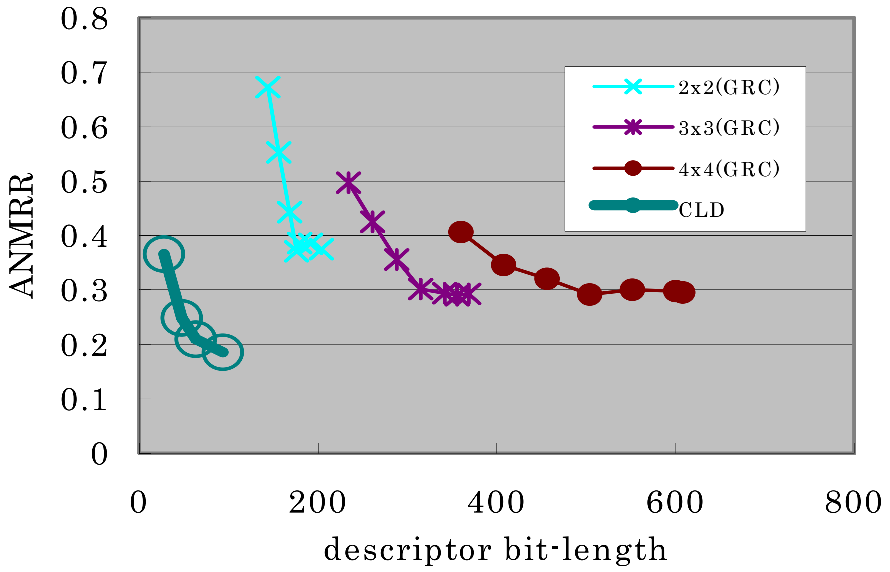


Figure 10. The retrieval efficiency of Color Layout compared with grid-based representative Color.

# of coeff's	# of bins H	# of bins S	# of bins V
16	4	2	2
32	8	2	2
64	8	2	4
128	8	4	4
256	16	4	4

Table 1. Equivalent partitioning of the HSV colour space for different numbers of coefficients in the Scalable Color Descriptor.

average number of colors	ANMRR(D)	ANMRR (D_S)	ANMRR(D_V)
3	0.31	0.30	0.25
5	0.25	0.21	0.16

Table 2: ANMRR results for Dominant Color.

# bits for the spatial coherence	ANMRR	
	Spatial coherence field with dominant colors	Spatial coherence for each dominant color
5	**0.221**	
4	0.227	
3	0.246	
2	0.250	**0.197**
1	0.252	0.202
0	**0.252** (without spatial coherence value)	

Table 3. ANMRR results for the dominant color with spatial coherence. An average of 5.3 colors per image are used for the MPEG-7 common color dataset. Increasing the number of bits beyond 5 bits did not give significant improvements. While assigning the bits to individual dominant colors gave better performance, the increased complexity of the descriptor was the main factor in choosing a single spatial coherence value.

No. of cells	**256**		**128**		**64**		**32**	
Subspace	**Hue**	**Sum**	**Hue**	**Sum**	**Hue**	**Sum**	**Hue**	**Sum**
0	1	32	1	16	1	8	1	8
1	4	8	4	4	4	4	4	4
2	16	4	8	4	4	4		
3	16	4	8	4	8	2	4	1
4	16	4	8	4	8	1	4	1

Table 4. HMMD subspace quantisation for each of the four partitions of the space.

	L_1	L_2
GoF - Average	0.041367	0.089982
GoF - Median	0.042614	0.090640
Optimal Keyframe	0.053500	0.101852

Table 5. Comparison of ANMRR video segment retrieval results using average, median GoF descriptor and the key-frame based histogram.

Descriptor Size	ANMRR
256 bins	0.06799
128 bins	0.07613
64 bins	0.09374
32 bins	0.14438

(a)

Descriptor Size	H×S×V quant.	ANMRR
256 bins	16×4×4	0.08707
128 bins	8×4×4	0.09204
64 bins	8×2×4	0.10700
32 bins	8×2×2	0.14832

(b)

Table 6. Color Structure Descriptor retrieval results using (a) HMMD color space and (b) HSV color space.

Component	Subspace	Number of quantization levels for different numbers of histogram bins			
		256	128	64	32
Hue	0	1	1	1	1
	1	4	4	4	4
	2	16	8		3
	3			8	2
	4				
Sum	0	32	16	8	8
	1	8	4	4	4
	2	4			
	3			2	1
	4			1	

Table 7. HMMD color space quantization for Color Structure Descriptor.

APPENDIX: Quantitative Evaluation

.

Experiments were conducted during the MPEG-7 standardization process to compare different competing technologies as well as to optimize adopted methods. Comparing and evaluating technologies for MPEG-7 visual descriptors presented a different set of challenges compared to previous MPEG standardization efforts, since there was no common ground rules for evaluating different methods. For visual descriptors, the retrieval application was found to be the best model to perform experiments. A good retrieval result in response to a visual-feature based query would be a good indicator for the expressiveness of the descriptor. In the experiments, the so-called *query by example* paradigm has been employed as the primary method for evaluation. In query-by-example, the respective descriptor values are extracted from the query image, and then matched to the corresponding descriptors of images contained in a database. In order to be objective in the comparisons, a quantitative measure was developed based on the specification of a dataset, a query set and the corresponding ground-truth data. The ground-truth data is a set of visually similar images for a given query image.

In defining an objective measure of retrieval effectiveness given a set of queries and the corresponding ground truth, the following factors are considered:

- The measure should be normalized to account for the variation in the size of the ground truth among different queries
- The measure should favor algorithms that retrieve the ground truth items as the top matches
- The measure should assign a penalty for each of the missed ground truth items. If a ground truth item is not retrieved within a certain number of top matches, then it is considered as missed.
- the order in which the ground truth items are retrieved the measure should favour algorithms that retrieve ground truth items in highest ranks
- the number of missed ground truth items by assigning a *penalty*. The penalty should be selected such that beyond a certain limit on the rank, it should not matter whether a ground truth item is found or not e.g. at the 200th or at the 2000th rank

The following solution was adopted. Consider a query *q* with a ground truth size of *NG(q)*; the rank **Rank**(*k)* of the k^{th} ground truth image is defined as the position at which this ground truth image is retrieved (a rank value of one corresponds to the top match). Further, a number $K(q) \geq NG(q)$ is defined that specifies the "relevant ranks", i.e. retrieval with rank larger than *K(q)* should be considered as a *miss*. For relatively large *NG*(*q*) (20-25 items), subjects would judge the retrieval results as useful if items are found within ranks around 2x*NG*(*q*), while for smaller ground truth sets, even more tolerance would be allowed. For ground truth items that are not retrieved in the top *K(q)* ranks, the penalty assigned should be greater than equal to *K*(*q*). But a penalty just equalling *K*(*q*) would place retrievals with too many misses at an advantage. A good compromise derived from this reasoning was found by defining a **Rank**(*k*) as:

$$\mathbf{Rank}(k) = \begin{cases} \mathbf{Rank}(k) & \text{if} \quad \mathbf{Rank}(k) \leq K(q) \\ 1.25 \cdot K(q) & \text{if} \quad \mathbf{Rank}(k) > K(q) \end{cases} \qquad \textbf{(A1)}$$

$$K(q) = \min\{4 \cdot NG(q), 2 \cdot \max[NG(q) \,\forall\, q]\}$$

From (A1) we get the Average Rank for query *q*

$$\mathbf{AVR}(q) = \frac{1}{NG(q)} \sum_{k=1}^{NG(q)} \mathbf{Rank}*(k) \qquad \textbf{(A2)}$$

However, with ground truth sets of different sizes (actually, *NG* varies between 3 and 32 in the CCQ), the **AVR** counted from ground truth sets with small and large *NG*(*q*) values would largely differ. To eliminate influences of different *NG*(*q*), the *Modified Retrieval Rank*

$$\mathbf{MRR}(q) = \mathbf{AVR}(q) - 0.5 \cdot \left[1 + NG(q)\right] \qquad \textbf{(A3)}$$

is defined, which is always larger than or equal to 0, but with upper margin still dependent on *NG*. This finally leads to the *Normalized Modified Retrieval Rank*

$$\mathbf{NMRR}(q) = \frac{\mathbf{MRR}(q)}{1.25 \cdot K(q) - 0.5 \cdot \left[1 + NG(q)\right]} \qquad \textbf{(A4)}$$

Note that NMRR(q) can take values between 0 (indicating whole ground truth found) and 1 (indicating nothing found), irrespective of the size of the ground truth items for query q, *NG(q)*. From (A4), it is straightforward to define the *Average Normalized Modified Retrieval Rank* (**ANMRR**), giving just one number indicating the retrieval quality over all queries. This has been used as the evaluation criterion in all MPEG-7 color experiments (as well as for the texture and shape descriptors discussed in the following chapters):

$$\mathbf{ANMRR} = \frac{1}{NQ} \sum_{q=1}^{NQ} \mathbf{NMRR}(q)\,\mathbf{,} \qquad \textbf{(A5)}$$

where *NQ* is the number of queries. There is evidence that the **ANMRR** measure approximately coincides linearly with the results of subjective evaluation about retrieval accuracy of search engines [5]. It was found in the experiments that there is a strong interrelationship between the compactness of a descriptor (as measured by the numbers of bits needed for the representation), and the retrieval accuracy. This allows the setup of "rate-accuracy curves" (similar to SNR based rate-distortion curves widely used in image and video coding).