Color and Texture Descriptors

B. S. Manjunath, *Member, IEEE*, Jens-Rainer Ohm, *Member, IEEE*, Vinod V. Vasudevan, *Member, IEEE*, and Akio Yamada

***Abstract*—This paper presents an overview of color and texture descriptors that have been approved for the Final Committee Draft of the MPEG-7 standard. The color and texture descriptors that are described in this paper have undergone extensive evaluation and development during the past two years. Evaluation criteria include effectiveness of the descriptors in similarity retrieval, as well as extraction, storage, and representation complexities. The color descriptors in the standard include a histogram descriptor that is coded using the Haar transform, a color structure histogram, a dominant color descriptor, and a color layout descriptor. The three texture descriptors include one that characterizes homogeneous texture regions and another that represents the local edge distribution. A compact descriptor that facilitates texture browsing is also defined. Each of the descriptors is explained in detail by their semantics, extraction and usage. Effectiveness is documented by experimental results.**

I. Introduction

COLOR and texture are among the more expressive of the visual features. Considerable work has been done in designing efficient descriptors for these features for applications such as similarity retrieval. For example, a color histogram is one of the most frequently used color descriptors that characterizes the color distribution in an image. This paper provides the reader with an overview of the technologies that are being considered by the MPEG-7 group for describing visual content based on its color and texture. More detailed information regarding the color and texture descriptors in MPEG-7 may be found in the references and other related MPEG documents.

The color and texture descriptors that are described in this paper have undergone rigorous testing and development during the past two years, and thus represent some of the more mature technologies for content representation. These tests and development were conducted under the various Core Experiments defined by the MPEG Video group and its *Ad-Hoc* Group on Color and Texture Core Experiments.

Section II describes the MPEG-7 Color and Texture Core Experiments, including a brief discussion on the color and texture datasets used in these experiments. This is followed by a description of color descriptors in Section III. Texture descriptors are discussed in Section IV. We conclude with a brief note on some of the unresolved issues at the time of writing this paper.

Manuscript received September 25, 2000; revised March 25, 2001

B. S. Manjunath is with the Electrical and Computer Engineering Department, University of California, Santa Barbara, CA 93106 USA (e-mail: manj@ece.ucsb.edu).

J.-R. Ohm is with the Institute for Communications Engineering, Aachen University of Technology, Aachen, Germany.

V. V. Vasudevan is with the NewsTakes Inc., Burlingame, CA 94010 USA.

A. Yamada is with the Computer and Communication Media Research, NEC Corporation, Kawasaki 216-85555, Japan (e-mail: a-yamada@da.jp.nec.com).

Publisher Item Identifier S 1051-8215(01)04987-4.

It must be emphasized that the main objective of this paper is to provide an overview of the MPEG-7 descriptors. Given the page restrictions on a transactions paper, the level of technical detail is not as thorough as we would have liked to provide. For a complete technical description, the interested reader is referred to [1], [2].

II. MPEG-7 Color/Texture Core Experiment Procedures

Core experiments are usually conducted during the MPEG standardization process to compare different competing technologies as well as to establish the merits of a proposed technology. Technologies in the video group under previous MPEG standards primarily dealt with efficient compression, and the signal-to-noise ratio (SNR) constituted an effective yardstick for comparison. Comparing and evaluating technologies for MPEG-7 visual descriptors presented a different set of challenges, as there existed no common ground rules for evaluating different methods. For visual descriptors, the retrieval application was found to be the best model. 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 Color and Texture Core Experiments, the so-called *query by example* paradigm has been employed as the primary method for evaluations. 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 needed. This requires specification of the datasets, the 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 the Color and Texture Core Experiments, the number of queries was about 1% of the number of images in the database. For example, in the color experiments, a common color dataset (CCD) consisting of around 5000 images, and a set of 50 common color queries (CCQ), each with specified ground truth images, have been defined.

The various sub-committees and *ad-hoc* groups within MPEG-7 worked in compiling this set of data over a period of over six months. For the Color and Texture Core Experiments, the dataset 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 were manually established through a process of visual inspection and cross verification by different groups of participants in MPEG. In the case of some descriptors, targeting stationary image features (such as the homogenous texture) a more objective strategy based on tiling a large image (e.g., image from the Brodatz album of textures) into smaller sub-images

was adopted [10]. All the color core experiments used the same datasets in computing the performance even though each descriptor addresses a different aspect of the visual content.

After databases and queries with ground truth have been defined, it is necessary to weigh the query results based on some numeric measure. A very popular measure is the retrieval rate (RR)

$$\mathrm{RR}(q) = \frac{\mathrm{NF}(a, q)}{\mathrm{NG}(q)} \tag{1}$$

where

- $\mathrm{NG}(q)$ size of the ground truth set for a query q;
- $\mathrm{NF}(\alpha, q)$ number of ground truth images found within the first $\alpha \cdot \mathrm{NG}(q)$ retrievals;
- $\mathrm{RR}(q)$ takes values between 0 and 1, where 0 stands for "no image found," and 1 for "all images found."

The factor α should be ≥ 1, where a larger α is more tolerant. If (1) is performed over the whole set of NQ queries, the average retrieval rate (ARR) is given by

$$\mathrm{ARR} = \frac{1}{\mathrm{NQ}} \sum_{q=1}^{\mathrm{NQ}} \mathrm{RR}(q). \tag{2}$$

While the RR and ARR are straightforward to compute, some issues remain. For an unconstrained dataset—typical of the image datasets used in retrieval experiments—it is not possible to have a fixed number of ground truth items for all the queries. Letting $\mathrm{NG}(q)$ vary with q introduces a bias for certain queries, particularly if there is a large variation in this number.

Further, RR as defined in (1) is a hard-limiting measure. Hence, setting $\alpha = 1$ may not be appropriate as retrieving an image from the ground truth with rank $\mathrm{NG} + 1$ would exclude it from contributing to (1), while in terms of subjective retrieval accuracy, this might not be too severe. On the other hand, selecting larger α values would be less discriminative between very good retrieval results and the not so good ones. For example, with $\alpha = 2$, RR would be equal for the cases where all images are found at ranks $1 \ldots \mathrm{NG}$, or where all images are found at ranks $\mathrm{NG} + 1 \ldots 2 \cdot \mathrm{NG}$, the latter one clearly being a worse result.

To address these problems, normalized measures that take into account different sizes of ground truth sets and the actual ranks obtained from the retrieval were defined. Retrievals that miss items are assigned a penalty. Consider a query q. Assume that as a result of the retrieval, the kth ground truth image for this query q is found at a specific $\mathrm{Rank}(k)$. Further, a number $K \geq \mathrm{NG}$ is defined which specifies the "relevant ranks," i.e., the ranks that would still count as feasible in terms of subjective evaluation of retrieval. For relatively large NG (20–25 items), subjects would judge the retrieval results as still useful if items are found with ranks around $2 \times \mathrm{NG}$, while for smaller ground truth sets, even more tolerance would be allowed. The penalty assigned should be $\geq K$, but it was argued that a penalty just equalling K would put retrievals with too many misses into advantage. A good compromise is to define a $\mathrm{Rank}^*(k)$ as

$$\mathrm{Rank}^*(k) = \begin{cases} \mathrm{Rank}(k), & \text{if } \mathrm{Rank}(k) \leq K(q) \\ 1.25K, & \text{if } \mathrm{Rank}(k) > K(q) \end{cases}. \tag{3}$$

From (3), we get the average rank (AVR) for query q

$$\mathrm{AVR}(q) = \frac{1}{\mathrm{NG}(q)} \sum_{k=1}^{\mathrm{NG}(q)} \mathrm{Rank}^*(k). \tag{4}$$

However, with ground truth sets of different size (actually, NG varies between 3 and 32 in the CCQ), the AVR counted from ground truth sets with small and large $\mathrm{NG}(q)$ values would differ significantly. To minimize the influence of variations in $\mathrm{NG}(q)$, a *modified retrieval rank* (MRR) is defined as follows:

$$\mathrm{MRR}(q) = \mathrm{AVR}(q) - 0.5 \cdot [1 + \mathrm{NG}(q)]. \tag{5}$$

Note that $\mathrm{MRR}(q)$ is 0 in the case of a perfect retrieval (ground truth items found at first $\mathrm{NG}(q)$ positions). However, the upper bound is still dependent on NG. A final normalization with respect to $\mathrm{NG}(q)$ leads to the *normalized modified retrieval rank* (NMRR)

$$\mathrm{NMRR}(q) = \frac{\mathrm{MRR}(q)}{1.25K - 0.5 \cdot [1 + \mathrm{NG}(q)]}. \tag{6}$$

$\mathrm{NMRR}(q)$ can take on values between 0 (indicating whole ground truth found) and 1 (indicating nothing found) only, irrespective of $\mathrm{NG}(q)$. From (6), it is straightforward to define the *average normalized modified retrieval rank* (ANMRR), giving just one number indicating the retrieval quality over all queries. The ANMRR is defined as

$$\mathrm{ANMRR} = \frac{1}{\mathrm{NQ}} \sum_{q=1}^{\mathrm{NQ}} \mathrm{NMRR}(q). \tag{7}$$

ANMRR is the evaluation criterion used in all of the MPEG-7 color core experiments. Evidence was shown that the ANMRR measure approximately coincides linearly with the results of subjective evaluation about retrieval accuracy of search engines [12]. Of course, evaluation of visual descriptors cannot be based only on retrieval accuracy. Further criteria are compactness, complexity of feature extraction and matching, scalability. Interestingly enough, it was found in the core experiments that there is a strong interrelationship between the compactness of a descriptor (as counted in 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). To make evaluation of the descriptors mostly independent from the design of the matching procedure, common matching methods have been used within core experiments wherever possible. Most experiments relied on the L1 norm, while some adopted the L2 norm and certain others employed statistical distance measures.

III. Color

Color is perhaps the most expressive of all the visual features and has been extensively studied in the image retrieval research during the last decade. A schematic of the color descriptors in the current version of the MPEG-7 Final Committee Draft (FCD) [1] is shown in Fig. 1. The color descriptors consist of a number of histogram descriptors, a dominant color descriptor,

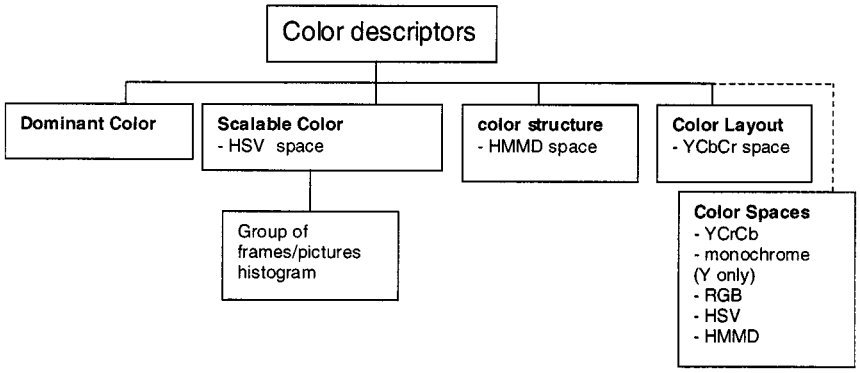

Fig. 1. MPEG-7 color descriptors.

and a color layout descriptor (CLD). Definition of this set of descriptors was done to serve different application domains while keeping the number of possible variants to a minimum, to guarantee interoperability between differently generated MPEG-7 color descriptions (see below). It is beyond the scope of this paper to summarize the whole selection process that occurred in the core experiments process; in general, descriptors were accepted and defined based on detailed studies of their efficiency (in terms of descriptor size and retrieval accuracy), complexity, as well as other criteria like applicability to a broad range of applications.

Color descriptors originating from histogram analysis have played a central role in the development of visual descriptors in MPEG-7. First, a generic color histogram descriptor was defined that would be able to capture the color distribution with reasonable accuracy for image search and retrieval applications. However, there are too many independent dimensions in a generic color histogram. These include choice of color space, choice of quantization in color space, and quantization of the histogram values. It was soon realized (after extensive experiments) that leaving this choice to the user would defeat the very purpose of the standard, i.e., the interoperability between descriptors generated by different MPEG-7 systems. There was a clear need to limit the set of histogram derived descriptors. The *scalable color descriptor* (SCD) is defined in the hue-saturation-value (HSV) color space with fixed color space quantization, and uses a novel Haar transform encoding. The Haar transform based encoding facilitates a scalable representation of the description, as well as complexity scalability for feature extraction and matching procedures. This descriptor can be extended to a collection of pictures or a group of frames from a video, and the *group of frames/group of pictures* (GoP) descriptor specifies different ways of constructing such a histogram. The *color structure histogram* aims at identifying localized color distributions using a small structuring window. To ensure interoperability, the color structure histogram is constructed in the *hue-min-max-difference* (HMMD) color space. A description of the HMMD color space is given in Section III-A.

The *dominant color* descriptor gives the distribution of the salient colors in the image. Unlike the bin quantization in the histograms, the specification of colors in a dominant color descriptor is limited only by the color space quantization. Its purpose is to provide an effective, compact, and intuitive representation of colors present in a region of interest.

The CLD captures the spatial layout of the dominant colors on a grid superimposed on the region of interest. This is a very compact descriptor that is very effective in fast browsing and search applications. It can be applied to still images, as well as to video segments.

The following sections provide more technical details on each of these color descriptors beginning with a brief description of the color spaces used in MPEG-7.

A. Color Space

The different color spaces used in MPEG-7 include the familiar monochrome, RGB, HSV, YCrCb, and the new HMMD. The monochrome (intensity only) space is also supported. This corresponds to the Y component in the YCrCb space. It is possible to define RGB with reference chromaticity primaries, if available from the capture process. The conversion from normalized RGB (where the values of each of the spectral components range from 0 to 1) to the other color spaces are shown in Fig. 2.

The HSV color space is a popular choice for manipulating color. The HSV color space is developed to provide an intuitive representation of color and to approximate the way in which humans perceive and manipulate color. RGB to HSV is a nonlinear, but reversible, transformation. The hue (H) represents the dominant spectral component—color in its pure form, as in green, red, or yellow. Adding white to the pure color changes the color: the less white, the more saturated the color is. This corresponds to the saturation (S). The value (V) corresponds to the brightness of color. The coordinate system is cylindrical, and is often represented by a subspace defined by a six-sided inverted pyramid. The top of the pyramid corresponds to $V = 1$, with the "white" at the center. The hue is measured by the angle around the vertical axis, with red corresponding to 0°. The saturation S ranges from 0 at the center to 1 on the surface of the pyramid. An inverted cone is also used to denote the subspace instead of the pyramid.

A new color space, the HMMD color space, is also supported in MPEG-7. The hue has the same meaning as in the HSV space, and max and min are the maximum and minimum among the R, G, and B values, respectively. The diff component is defined

```
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
```

(a)

```
Max = max(R, G, B); Min = min( R, G, B);
Value = max(R, G, B);
if( Max == 0 ) then
  Saturation = 0; else
  Saturation = (Max-Min)/Max;
if( Max == Min ) Hue is undefined (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))
```

(b)

```
Diff=Max-Min
Sum=(max+min)/2
Hue as defined for the HSV.
```

(c)

Fig. 2. Color spaces used in MPEG-7. (a) RGB to YCbCr color space. (b) RGB to HSV color space. (c) RGB to HMMD color space.

as the difference between max and min. Only three of the four components are sufficient to describe the HMMD space. This color space can be depicted using the double cone structure as shown in Fig. 3. In the MPEG-7 core experiments for image retrieval, it was observed that the HMMD color space is very effective and compared favorably with the HSV color space. Note that the HMMD color space is a slight twist on the HSI color space [6], where the *diff* component is scaled by the intensity value. The HMMD color space is used in the color structure descriptor (CSD).

To ensure inter-operability, the color spaces allowed for the various color descriptors are constrained by the standard. The dominant color descriptor allows color specification in any of the color spaces supported by MPEG-7. The RGB space is not very efficient for search and retrieval tasks and is not explicitly used in any color descriptor. The SCD uses the HSV space and the color structure histogram uses the HMMD space. The CLD is defined for the YCrCb space. These color space descriptors are also used outside of the visual descriptors, for example, in specifying "media properties" in suitable description schemes.

B. SCD

The generic Color Histogram Descriptor defined in early MPEG-7 experiments is a compound descriptor consisting of color space, color quantization and histogram descriptors. This would allow specification of color histograms with varying numbers of bins and nonuniform quantization of different color spaces. However, it was not desirable to provide too much flexibility in such a specification, as it would limit interoperability between different descriptions based on MPEG-7. The SCD addresses the interoperability issue by fixing the color space to

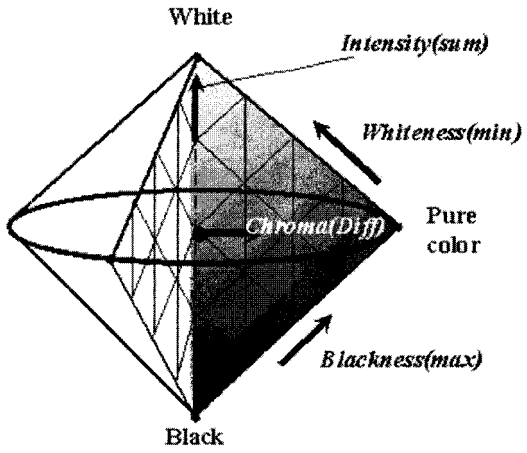

Fig. 3. HMMD color space.

HSV, with a uniform quantization of the HSV space to 256 bins. The bin values are nonuniformly quantized to a 11-bit value.

This method achieves full interoperability between different resolutions of the color representation, ranging from 16 bits/histogram at the low end to approximately 1000 bits/histogram at the high end. Of course, the accuracy of the feature description is highly dependent on the number of bits used. However, core experiments have shown that good retrieval results are still achievable using only 64 bits, while excellent results can be obtained using medium or full resolution of the descriptor.

The HSV space is uniformly quantized into a total of 256 bins. This includes 16 levels in H, four levels in S, and four levels in V. The histogram values are truncated into a 11-bit integer representation. To achieve a more efficient encoding, the 11-bit integer values are first mapped into a "nonlinear" 4-bit

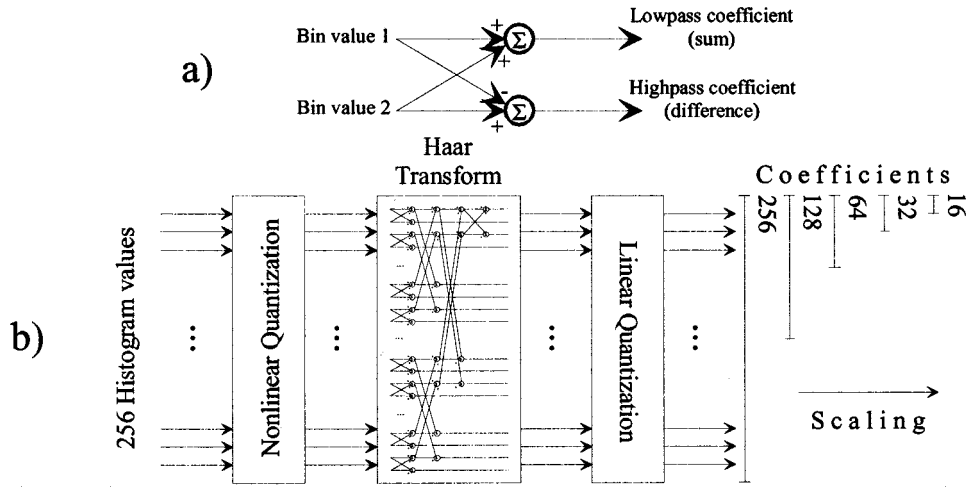

Fig. 4. (a) Basic unit of Haar transform. (b) A schematic diagram of SCD generation.

representation, giving higher significance to the small values with higher probability.

This 4-bit representation of the 256-bin HSV histogram would require 1024 bits/histogram, which is too large a number in the context of many MPEG-7 applications. To lower this number and make the application scalable, the histograms are encoded using a Haar transform.

The basic unit of the Haar transform consists of a sum operation and a difference operation [see Fig. 4(a)], which relate to primitive low- and high-pass filters. Summing pairs of adjacent histogram lines is equivalent to the calculation of a histogram with half number of bins. If this process is performed iteratively, usage of subsets of the coefficients in the Haar representation is equivalent to histograms of 128, 64, 32... bins, which are all calculated from the source histogram.

The high-pass (difference) coefficients of the Haar transform express the information contained in finer-resolution levels (with higher number of bins) of the histogram. Natural image signals usually exhibit high redundancy between adjacent histogram lines. 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 integer representation with only a low number of bits.

Fig. 4(b) shows the block diagram of the complete system. 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. Table I shows the relationship between numbers of Haar coefficients as specified in the SCD and partitions in the components of a corresponding HSV histogram that could be reconstructed from the coefficients.

A different type 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 retrained. 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. In 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. 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.

TABLE I
EQUIVALENT PARTITIONING OF THE HSV COLOR SPACE FOR DIFFERENT CONFIGURATIONS OF THE SCD

16	4	2	2
32	8	2	2
64	8	2	4
128	8	4	4
256	16	4	4

In similarity matching of histograms, the L1 norm (sum of absolute differences) usually results in good retrieval accuracy. L1-norm-based matching can likewise be applied in the Haar transform domain; however, results are not identical (except for the case where the "high-pass" coefficients have identical signs in the two descriptions compared), as matching directly in the histogram domain. In the case where only the sign bit is used (all bit planes representing the absolute value discarded), the L1 norm degenerates to a Hamming distance, allowing even less complexity in the search.

For computing the retrieval accuracy of the SCD, the ANMRR measure as described in Section II is used. The number of Haar coefficients used for matching was between 16 and 256 (see Table I), by which bin-number scalability is achieved. For bit plane scalability, a 1 bit (sign only) representation to a full range representation was explored. The results are shown in Fig. 5. In addition, the ANMRR was calculated in the histogram domain after performing an inverse Haar transform. This is given by the plot labeled H-Rec in Fig. 5.

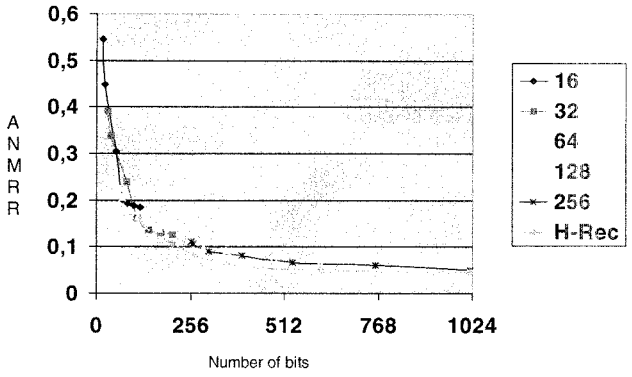

Fig. 5. Results with different numbers of Haar coefficients (16–256) quantized at different numbers of bits. H-Rec signifies retrieval results after reconstruction of histogram from Haar coefficients at full bit resolution.

Matching in the histogram domain appears to provide the best overall performance, as shown by the H-Rec curve. The results show that a reasonable performance can be achieved even at 16 and 32 bits/histogram representations and the performance appears to saturate at around 128 and 256 bits/histogram.

Matching coefficients directly in the Haar space is equal in complexity compared to matching in the histogram space, assuming the number of coefficients equal the number of histogram bins and the distance measures are the same in both cases. The complexity of generating the Haar coefficients is marginal compared to generating the histograms, and as such does not add to the feature extraction complexity.

Comparison of different-size representations in the SCD is quite simple. In SCD, it is straightforward to perform matching on subsets of Haar coefficients, which correspond to a coarser approximation of the source histogram. This also allows application of coarse-to-fine matching. For a given query, a coarse version of SCD is matched first to select a subset of image candidates in a database, and a refined matching based on more coefficients is performed only for this subset. Such a procedure can achieve significant speed up in similarity search in large databases.

The GoP extends the SCD application to a collection of images, video segments, or moving regions. In the GoP descriptor, three different ways of computing the joint color histogram values for the whole series using the individual histograms from items within the collection are identified: averaging, median filtering, and histogram intersection. This joint color histogram is then processed as in the SCD using the Haar transform and encoded.

C. CSD

This descriptor expresses local color structure in an image using an 8×8-structuring element. It counts the number of times a particular color is contained within the structuring element as the structuring element scans the image. Suppose $c_0, c_1, c_2, \ldots, c_{M-1}$ denote the M quantized colors. A color structure histogram can then be denoted by $h(m)$, $m = 0, 1, \ldots, M-1$, where the value in each bin represents the number of structuring elements in the image containing one or more pixels with color c_m. The HMMD color space is used in this descriptor.

TABLE II
HMMD COLOR SPACE QUANTIZATION FOR CSD

Component	Subspace	Number of quantisation levels for different numbers of histogram bins			
		184	120	64	32
Hue	0	1	1	1	1
	1	8	4	4	4
	2	12	12	6	3
	3	12	12	4	2
	4	24			
Sum	0	8	8	8	8
	1	4	4	4	2
	2	4	4	4	4
	3	4	4	4	2
	4	2			

The CSD is defined using four color space quantization operating points: 184, 120, 64, and 32 bins. To construct a 184-level quantized color, HMMD color space is quantized nonuniformly as follows. The whole HMMD color space is divided into five subspaces. This sub-space division is performed on the diff parameter (see Section III-A). For the respective subspaces, uniform color quantization on the Hue and Sum values results in a 184-level color quantization. The number of quantization levels for each subspace for different number of histogram bins is given in Table II.

In order to compute the CSD, an 8×8-structuring element is used. Even though the total number of samples is kept fixed at 64, the spatial extent of the structuring element scales with the image size. The following simple rule determines the spatial extent of the structuring element (equivalently, the sub sampling factor) given the image size:

$$p = \max\{0, \text{round}(0.5 \log_2 WH - 8)\}$$
$$K = 2^p, \quad E = 8K \tag{8}$$

where

W, H image width and height, respectively;
$E \times E$ spatial extent of the structuring element;
K sub-sampling factor.

For images smaller than 256×256 pixels, an 8×8 element with no sub-sampling is used. As another example, if the image size is 640×480, then $p = 1$, $K = 2$, and $E = 16$. So, every alternate sample along the rows and columns of a 16×16-structuring element is then used to compute the histogram.

Fig. 6 (only a part of the image is shown) shows the structuring element in the initial location at the upper left corner of the image. The structuring element slides over the image and is shifted by one pixel in Fig. 6(a) and by two pixels in case Fig. 6(b). Case (b) corresponds to sub-sampling of the image by two in both directions and subsequently applying the same 8×8-structuring element. Each bin of the CSD $h(m)$ represents the number of locations of the structuring element at which a pixel with color c_m falls inside the element. The origin of the structure element is defined by its top-left sample. The locations of the structure element over which the descriptor is accumulated are defined by the grid of pixels of the possibly sub-sampled input image.

The bin values $h(m)$ of the CSD are normalized by the number of locations of the structuring element and lie in the

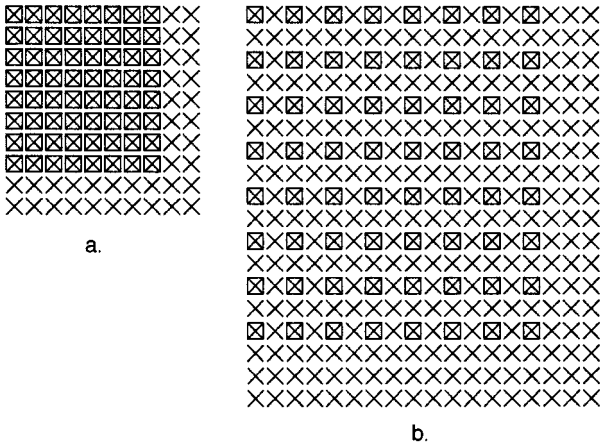

Fig. 6. Structuring elements for images with different resolutions: (a) 320×240 and (b) 640×480.

TABLE III
ANMRR RESULTS FOR THE CSD USING THE HMMD COLOR SPACE

# bins	8 bits	6 bits	4 bits	2 bits
184 bins	0.046	0.046	0.066	0.226
120 bins	0.049	0.051	0.067	0.230
64 bins	0.068	0.073	0.087	0.273
32 bins	0.105	0.107	0.130	0.342

range $[0.0, 1.0]$. The bin values are then nonlinearly quantized to 8 bits/bin.

CSDs containing 120, 64, or 32 bins are computed based on approximations computed using the 184-bin descriptor. The mapping of the 184-bin descriptor to a descriptor with a lower number of bins is defined by re-quantizing the color represented by each bin of the 184-bin descriptor into the more coarsely quantized color space as specified in Table II.

Similar to the other histogram descriptors, an L distance measure is used to compute the dissimilarity between two CSDs. Table III shows the performance of this descriptor for varying number of bins and bit quantization. The common color dataset was slightly modified by the addition of few more query images so as to illustrate the qualitative difference in the retrieval performance between the color structure and scalable color histograms.

D. Dominant Color

A set of dominant colors in a region of interest or in an image provide a compact description that is easy to index. The target application is similarity retrieval in large image databases using color. Colors in a given region are clustered into a small number of representative colors. The feature descriptor consists of the representative colors, their percentages in the region, spatial coherency of the dominant colors, and color variances for each dominant color. A similarity measure similar to the quadratic color histogram distance measure is defined for this descriptor. The representative colors can be indexed in the 3-D color space thus avoiding the high-dimensional indexing problems associated with the traditional color histogram. For similarity retrieval, each representative color in the query image or region is used independently to find regions containing that color. The matches from all of the query colors are then combined to obtain the final retrievals. An efficient indexing scheme for the dominant color descriptor is presented in [4].

The difference between the dominant color descriptor and the color histogram descriptor is that the representative colors are computed from each image instead of being fixed in the color space, thus allowing the feature representation to be accurate as well as compact.

In order to compute this descriptor, the colors present in a given image or region are first clustered (see [4] and [7] for more details). This results in a small number of colors and the percentages of these colors are calculated. As an option, the variances of the colors assigned to a given dominant color are also computed. The percentages of the colors present in the region should add up to 1. A spatial coherency value is also computed that differentiates between large color blobs versus colors that are spread all over the image. The descriptor is thus defined by

$$F = \{\{c_i, p_i, v_i\}, s\}, \quad (i = 1, 2 \ldots, N) \tag{9}$$

where

c_i ith dominant color;
p_i its percentage value;
v_i its color variance.

The color variance is an optional field. 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 can be used to represent the region. The percentage values p_i are quantized to 5 bits each. The color quantization depends on the color space specifications defined for the entire database and need not be specified with each descriptor.

The method described in [1] for dominant color extraction is based on using the generalized Lloyd algorithm for color clustering. This problem is formulated as one of minimizing the distortion D_i in each cluster i

$$D_i = \sum_n v(n) \|x(n) - c_i\|^2 \quad x(n) \in C_i \tag{10}$$

where

c_i centroid of cluster C_i;
$x(n)$ color vector at pixel;
$v(n)$ perceptual weight for pixel n.

The perceptual weights are calculated from the local pixel statistics to account for the fact that human vision perception is more sensitive to changes in smooth regions than in textured regions. These perceptual weights are given in [1]. The variance of the colors associated with a cluster C_i, (and hence the dominant color c_i) is then computed and quantized to 3 bits per color variance.

The normalized average number of connecting pixels of the corresponding dominant color using a 3×3 masking window measures the spatial coherence of a given dominant color. The overall spatial variance is then a linear combination of the individual spatial variances with the corresponding percentages p_i being the weights. The spatial variance is quantized to 5 bits, where 31 means highest confidence and 1 means no confidence. 0 is used for cases where it is not computed.

TABLE IV
ANMRR RESULTS FOR THE DOMINANT CLD

Color Space	#dominant colors (average)	DC			DC+Variance		
		Size(bits)	ARR	ANMRR	Size (bits)	ARR	ANMRR
RGB	3	69	0.6368	0.3897	78	0.7163	0.3222
	6	130	0.7114	0.3214	148	0.7933	0.2295
CIE-LAB	3	67	0.7568	0.2784	76	0.8160	0.2350
	5	112	0.8083	0.2312	127	0.8951	0.1563

TABLE V
ANMRR RESULTS FOR THE DOMINANT COLOR WITH SPATIAL COHERENCE

for the spatial coherence	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)	

An average of 5.3 colors per image are used for the MPEG-7 common color dataset. Increasing the number of bits beyond 5 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.

Each object or region in the database is represented using the dominant color descriptor as defined in (9). 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, $F_1 = \{\{c_{1i}, p_{1i}, v_{1i}\}, s_1\}, \ (i = 1, 2, \ldots, N_1)$ and $F_2 = \{\{c_{2i}, p_{2i}, v_{2i}\}, s_2\}, \ (i = 1, 2, \ldots, N_2)$. Ignoring the optional variance parameter and the spatial coherence, the dissimilarity $D(F_1, F_2)$ between the two descriptors can be computed as

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2a_{1i,2j} p_{1i} p_{2j} \tag{11}$$

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

$$a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{\max}, & d_{k,l} \le T_d \\ 0, & d_{k,l} > T_d \end{cases} \tag{12}$$

where

$d_{k,l}$ $= \|c_k - c_l\|$ Euclidean distance between two colors c_k and c_l;

T_d maximum distance for two colors to be considered similar;

$d_{\max}$ $= \alpha T_d$.

In particular, this means that any two dominant colors from one single description are at least T_d distance apart. A normal 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. This distance can be modified to take into account the optional variance [2]. One can then take a linear combination of the spatial coherency and the above distance to give a combined distance as suggested in [2].

The binary semantics 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 quantization is not part of the descriptor. The optional color variances are encoded at 3 bits per color with nonuniform quantization. This is equivalent to 1 bit per component space in the 3-D color spaces. The ANMRR results on the CLD are given in Table IV. Here, the respective color spaces are uniformly quantized to 6 bits per color value. The results are shown for different number of average number of dominant colors used. Table V gives results using the spatial variance parameter and comparing with the dc descriptor (without variance). These results differ somewhat from those in Table IV due to the different color spaces and quantization used in the experiments. 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. The datasets and evaluation does not truly reflect this objective, and the results provided are to be interpreted accordingly. On the other hand, they did serve the useful purpose of identifying different extensions and as a baseline for comparisons among competing dominant color descriptors.

E. CLD

The CLD is designed to capture the spatial distribution of color in an image or an arbitrary-shaped region. The spatial distribution of color constitutes an effective descriptor for sketch-based image retrieval, content filtering using image indexing,

and visualization. The functionality of this descriptor can also be achieved using a combination of grid structure descriptor and grid-wise dominant colors. However, such a combination would require a relatively large number of bits, and matching will be more complex and expensive. For several applications, a compact yet effective descriptor is needed, and the CLD satisfies these needs.

The CLD is a compact descriptor that uses representative colors on an 8×8 grid followed by a DCT and encoding of the resulting coefficients. The feature extraction process consists of two parts; grid based representative color selection and DCT transform with quantization. An input picture is divided into 64 (8×8) blocks and their average colors are derived. Note that it is implicitly recommended that the average color be used as the representative color for each block. This partitioning process is important to guarantee the resolution or scale invariance. The derived average colors are transformed into a series of coefficients by performing 8×8 DCT. A few low-frequency coefficients are selected using zigzag scanning and quantized to form a CLD. The color space adopted for CLD is YCrCb.

For matching two CLDs, $\{\mathrm{DY}, \mathrm{DCr}, \mathrm{DCb}\}$ and $\{\mathrm{DY}', \mathrm{DCr}', \mathrm{DCb}'\}$, the following distance measure is used:

$$D = \sqrt{\sum_i w_{yi}(\mathrm{DY}_i - \mathrm{DY}'_i)^2} + \sqrt{\sum_i w_{bi}(\mathrm{DCb}_i - \mathrm{DCb}'_i)^2} + \sqrt{\sum_i w_{ri}(\mathrm{DCr}_i - \mathrm{DCr}'_i)^2}. \quad (13)$$

Here, $(\mathrm{DY}_i, \mathrm{DCr}_i, \mathrm{DCb}_i)$ represent the ith DCT coefficients of the respective color components. The distances are weighted appropriately, with larger weights given to the lower frequency components.

Fig. 7 shows the performance of this descriptor on the common color dataset and illustrates the bit-size scalability. The default recommended number of bits is 63. This includes six Y coefficients, and three each of Cr and Cb coefficients. The dc values are quantized to 6 bits and the remaining to 5 bits each. These results demonstrate that the CLD is quite effective in image retrieval. The results also compare favorably with a grid based dominant color approach wherein the image is partitioned and dominant colors for these partitions are used to represent the layout. This descriptor can also be used for fast video browsing and retrieval.

IV. TEXTURE

Texture, like color, is a powerful low-level descriptor for image search and retrieval applications. MPEG-7 is considering three texture descriptors at this time. The first one is referred to as the "texture browsing descriptor" and characterizes perceptual attributes such as directionality, regularity, and coarseness of a texture. The second one, the "homogeneous texture descriptor " (HTD) provides a quantitative characterization of homogeneous texture regions for similarity retrieval. It is based on computing the local spatial-frequency statistics of the texture. The last one, the "local edge histogram descriptor," is useful when the underlying region is not homogeneous in texture properties.

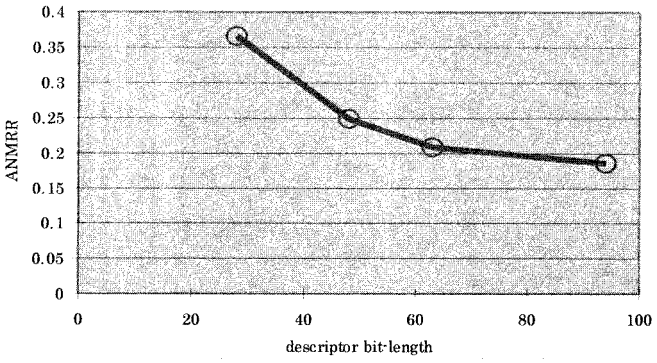

Fig. 7. Experimental results for the CLD.

A. Texture Browsing Descriptor

This is a compact descriptor that requires only 12 bits (maximum) to characterize a texture's regularity (2 bits), directionality $(3 \text{ bits} \times 2)$, and coarseness $(2 \text{ bits} \times 2)$. A texture may have more than one dominant direction and associated scale. For this reason, the specification allows a maximum of two different directions and coarseness values.

The regularity of a texture is graded on a scale of 0 to 3, with 0 indicating an irregular or random texture. A value of 3 indicates a periodic pattern with well-defined directionality and coarseness values. There is some flexibility (or implied ambiguity) in the two values in between. Having a well-defined directionality even in the absence of a perceivable micro-pattern is considered more regular than a pattern that lacks directionality and periodicity [Fig. 8(b)], even if the individual micro-patterns are clearly identified as in Fig. 8(c).

The directionality of a texture is quantized to six values, ranging from 0° to 150° in steps of 30°. The texture in Fig. 8(a) has strong vertical and horizontal directionalities. Up to two directions can be specified. Three bits are used to represent the different directions. The value "0" is used to signal textures that do not have any dominant directionality, and the remaining directions are represented by values from 1 to 6. Associated with each dominant direction is a coarseness component. Coarseness is related to image scale or resolution. It is quantized to four levels, with 0 indicating a fine grain texture and a "3" indicating a coarse texture. These values are also related to the frequency space partitioning (see Fig. 9) used in computing the HTD.

The computation of the browsing descriptor is described in detail in [11]. The image is filtered using a bank of scale and orientation selective band-pass filters and the filtered outputs are then used to compute the texture browsing descriptor components. The image filtering part is similar to the one for the HTD (see below) and as such both these descriptors can be efficiently computed. Since the descriptor semantics can be related to human perception of the texture, manual specification of the descriptor is also possible.

This descriptor is useful for browsing applications, and in conjunction with the HTD can help in fast and accurate image retrieval. In browsing, any combination of the three main

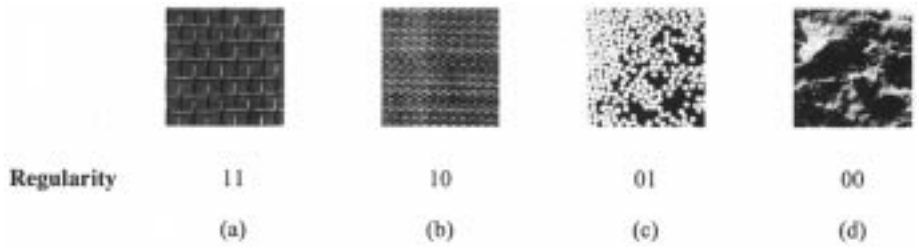

Fig. 8. Examples of regularity component.

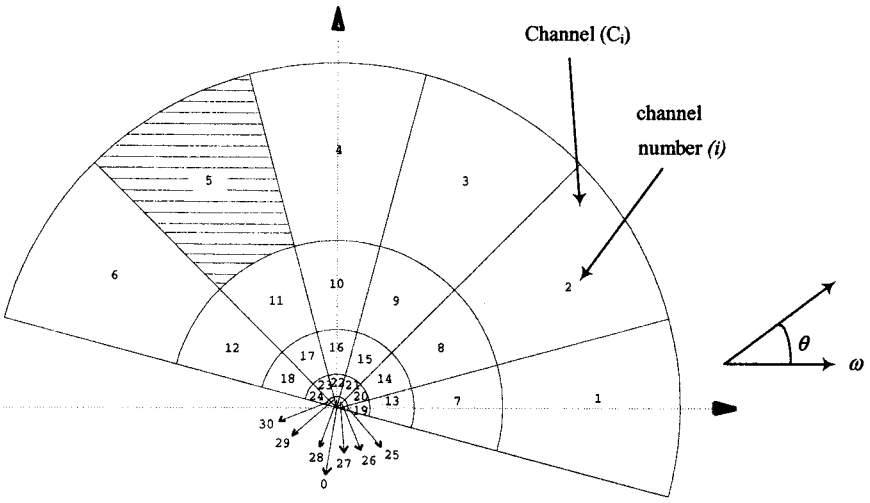

Fig. 9. Frequency layout for texture feature extraction.

components—regularity, directionality, and coarseness—can be used to browse the database. For example, one can look for textures that are very regular and oriented at 30°. In similarity retrieval, the texture browsing descriptor can be used to find a set of candidates with similar perceptual properties and then use the HTD to get a precise similarity match list among the candidate images.

B. HTD

The HTD provides a quantitative characterization of texture for similarity-based image-to-image matching. This descriptor is computed by first filtering the image with a bank of orientation and scale sensitive filters, and computing the mean and standard deviation of the filtered outputs in the frequency domain. Previous extensive work on this feature descriptor has shown that this descriptor is robust, effective, and easy to compute [1], [5], [8], [10], [17]. During the MPEG-7 Core Experiments, it was realized that the computational complexity of this descriptor can be reduced significantly by computing the values in the frequency domain rather than in the spatial domain, and an efficient implementation using Radon transform is described in [18].

The computation of this descriptor is as follows. The frequency space is partitioned into 30 channels with equal divisions in the angular direction (at 30° intervals) and octave division in the radial direction (five octaves), as shown in Fig. 9. In a normalized frequency space $0 \leq \mathsf{W} \leq 1$, the center frequencies of the feature channels are spaced equally in 30° in angular direction such that $\mathsf{q}_r = 30° \times r$, where r is angular index with $r \in \{0,1,2,3,4,5\}$. In the radial direction, the center frequencies of the neighboring feature channels are spaced one octave apart such that $\omega_s = \omega_0 \cdot 2^{-s}, s \in \{0,1,2,3,4\}$ where s is radial index and $\mathsf{W}_0 = 3/4$ is the highest center frequency. The various channels are numbered as shown in Fig. 9 and the channel index i can be expressed as $i = 6 \times s + r + 1$.

The individual feature channels are modeled using 2-D Gabor functions. Gabor functions are modulated Gaussians. The Fourier transform of a 2-D Gabor function in the polar coordinates can be written as

$$G_{P_{s,r}}(\mathsf{W},\mathsf{q}) = \exp\left[\frac{-(\mathsf{W}-\mathsf{W}_s)^2}{2\mathsf{S}_{r_s}^2}\right] \cdot \exp\left[\frac{-(\mathsf{q}-\mathsf{q}_r)^2}{2\mathsf{S}_{q_r}^2}\right] \quad (14)$$

For the bank of filters used, the filter parameters are selected such that the half-maximum contours of the 2-D Gaussians of adjacent filters in the radial and angular directions touch each other. In the angular direction, S_{q_r} has a constant value of $15°/\sqrt{2\ln 2}$. In the radial direction, S_{r_s} depends on the octave bandwidth and is written as

$$\mathsf{S}_{r_s} = \frac{B_s}{2\sqrt{2\ln 2}}. \quad (15)$$

The image texture energy in each of the filtered channels is then computed. Note that this is equivalent to weighting the Fourier transform coefficients of the image with a Gaussian centered at the frequency channels as defined above. The deviation of the energy is also computed. Both the energy and the energy

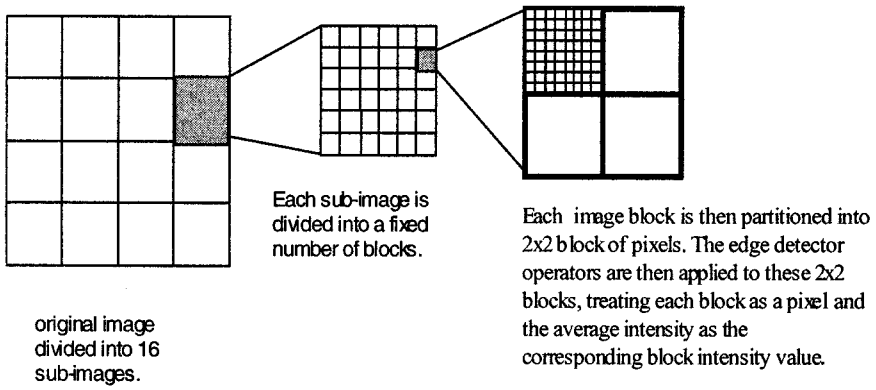

Fig. 10. Computing the edge histogram descriptor.

deviation are then logarithmically scaled to obtain two numbers, e_i and d_i, for the ith feature channel. The HTD is then given by

$$\mathrm{TD} = [f_{\mathrm{DC}}, f_{\mathrm{SD}}, e_1, e_2, \ldots, e_{30}, d_1, d_2, \ldots, d_{30}]. \quad (16)$$

The first two components of the feature vector are the mean intensity and the standard deviation of the image texture, respectively. The details of nonlinear scaling and quantization of these values can be found in the current MPEG-7 visual final committee draft [1].

Similarity Matching: The distance between two HTDs is computed as follows:

$$\begin{aligned} &d(\mathrm{TD}_{\mathrm{query}}, \mathrm{TD}_{\mathrm{Database}}) \\ &\quad = \mathrm{distance}(\mathrm{TD}_{\mathrm{query}}, \mathrm{TD}_{\mathrm{Database}}) \\ &\quad = \sum_k \left| \frac{\mathrm{TD}_{\mathrm{query}}(k) - \mathrm{TD}_{\mathrm{Database}}(k)}{a(k)} \right|. \end{aligned} \quad (17)$$

The recommended normalization value $a(k)$ is the standard deviation of $\mathrm{TD}_{\mathrm{Database}}(k)$ for a given database.

Note that shifting the feature vector components corresponding to a given scale value is equivalent to a rotation in space. *Rotation invariant matching* [16] can be achieved by shifting the query vector components appropriately before matching with the database items. In patching two patterns i and j, the minimum of the distances between the shifted ith pattern vector and the jth feature vector is then used as the distance between the two patterns i and j. This can be written as

$$d(i, j, m\mathsf{f}) = \mathrm{distance}(\mathrm{TD}_i|_{m\mathsf{f}}, \mathrm{TD}_j)$$

where $\mathsf{f} = 30°$. Then, for rotation invariant matching, distance is calculated as

$$d(i, j) = \text{minimum of}\{d(i, j, m\mathsf{f}) \,|\, m = 0 \text{ to } 5\}.$$

The performance of this texture descriptor is evaluated on a large texture image dataset consisting of images from the Brodatz album [1], aerial images [9], and stock photo images and textures from Corel. For rotation and scale invariant matching, additional images are created by digitally scaling and rotating the texture images from the above datasets. The total number of images used in the Core Experiments exceeds 10 000. On the Brodatz data set, with experimental conditions as described in [10], the retrieval accuracy is about 77%.

C. Edge Histogram Descriptor

The edge histogram descriptor captures the spatial distribution of edges, somewhat in the same spirit as the CLD. The distribution of edges is a good texture signature that is useful for image to image matching even when the underlying texture is not homogeneous. The computation of this descriptor is fairly straightforward (see Fig. 10) . A given image is first sub-divided into 4×4 sub-images, and local edge histograms for each of these sub-images is computed . Edges are broadly grouped into five categories: vertical, horizontal, 45° diagonal, 135° diagonal, and isotropic (nonorientation specific). Thus, each local histogram has five bins corresponding to the above five categories. The image partitioned into 16 sub-images results in 80 bins. These bins are nonuniformly quantized using 3 bits/bin, resulting in a descriptor of size 240 bits [1].

To compute the edge histograms, each of the 16 sub-images is further subdivided into image blocks. The size of these image blocks scale with the image size and is assumed to be a power of 2. The number of image blocks per sub-image is kept constant, independent of the original image dimensions, by scaling their size appropriately. A simple edge detector is then applied to each of the macro-block, treating the macro-block as a 2×2 pixel image. The pixel intensities for the 2×2 partitions of the image block are computed by averaging the intensity values of the corresponding pixels. The edge-detector operators include four directional selective detectors and one isotropic operator (Fig. 11). Those image blocks whose edge strengths exceed a certain minimum threshold are used in computing the histogram.

Thus, for an image block, we can compute five edge strengths, one for each of the five filters from Fig. 11. If the maximum of these edge strengths exceed a certain preset threshold, then the corresponding image block is considered to be an edge block. An edge block contributes to the edge histogram bins. The edge computation method is quite simple and can be applied directly to MPEG-2 compressed bit streams.

Each of the image blocks labeled as edge blocks contribute to the appropriate bin of the histogram descriptor. These values are normalized to $[0, 1]$. A nonlinear quantization of the bin values results in a 3 bits/bin representation.

Similarity Matching: Note that there are a total of 80 bins, 3 bits/bin, in the edge histogram. One can use the 3-bit number as an integer value directly and compute the L1 distance between

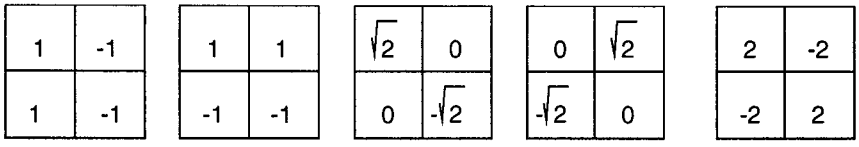

Fig. 11. Filters for edge detection.

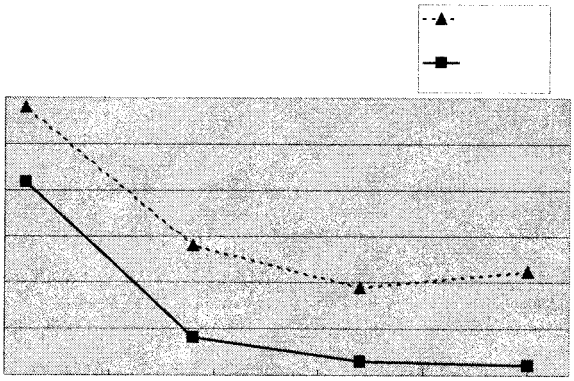

Fig. 12. ANMRR results for the edge histogram descriptor.

two edge histograms. A slightly better performance is obtained if the 3-bit values are decoded using look-up tables.

An interesting variation is to compute an extended histogram from these 80 bins [15]. The extended histogram is obtained by grouping the image blocks (and the corresponding bins). The extended bins are referred to as the global and semi-global histograms. The global histogram is obtained by combining all the 16 image blocks. The semi-global histograms are computed by pooling the image blocks/bins by rows (four rows), columns (four columns) and in groups of 2×2 (five groups). This results in five bins for the global histogram and 13×5 for the semi-global histograms from the 80 local histogram bins. The total number of bins is thus 150. A weighted L1 measure, with the distances corresponding to the global bins given more weight than the others, is used to compute the distance between two edge histograms. In the evaluation, a set of about 11 000 images from the MPEG-7 collection is used. On this data set, the ANMRR is about 0.34 using the 80-bin edge histograms and improved to about 0.30 when the extended histograms are used. In both cases, the bins are represented at 3 bits/bin.

The edge histogram descriptor is found to be quite effective for representing natural images with the primary application being image-to-image matching. The performance can be further enhanced by using this descriptor in conjunction with other image features, such as color [13]. Similar to color, this descriptor can be used in scene change detection and key frame clustering in video. One observed limitation of this descriptor, unlike the HTD, is that it cannot be used for object based image retrieval.

V. Conclusion

In this paper, we have presented the technical details of color and texture descriptors currently in the MPEG-7 standard. The color descriptors include two histogram-based descriptors, the SCD and the CSD, the dominant color descriptor, and the CLD. The histogram descriptors capture the global distribution of color where as the dominant color descriptor represents the dominant colors present. The CLD captures the spatial distribution or layout of the colors in a compact representation. While MPEG-7 standards accommodate different color spaces, most of the color descriptors are constrained to one or a limited number of color spaces for ensuring inter-operability.

The texture descriptors include a HTD and an edge histogram texture descriptor. Both these descriptors support search and retrieval based on content descriptions. In addition, a compact texture based browsing descriptor is also supported.

All these descriptors have been rigorously tested and evaluated following the MPEG-7 Core Experiment procedures to ensure their effectiveness and efficiency in a wide variety of applications based on multimedia content description. While MPEG-7 standardizes only the representation of these descriptors, a detailed description of the recommended methods for extracting and matching the descriptors are presented in the current visual XM document [2] that is intended to become a non-normative part of the MPEG-7 standard as a Technical Report.

The MPEG-7 Final Committee Draft was just released at the time of writing this paper [1]. While most of the technical work on color and texture descriptors have been completed, there are a few interesting technologies which are still in various stages of evaluations. Notably, the color descriptors discussed in this article are mainly suited for natural images and video and will cover the needs of the bulk of applications based on content descriptions. However, for synthetic images or for very specialized domains such as bio-medical imagery, refinements of existing descriptors and/or additional descriptors may be needed.

Acknowledgment

The authors acknowledge the help of the following individuals in preparing this manuscript. Dr. L. Cieplensky, Mitsubishi Electric; S. Jeannin, Philips Research Labs; Dr. H. J. Kim, LG Electronics; S.-J. Park, ETRI; Dr. Y. Choi, Samsung Electronics; Prof. Y. M. Ro, Information Communications University; Dr. P. Van Beek, Sharp Labs of America, Prof. C. S. Won, Dongguk University.

References

[1] *Text of ISO/IEC 15 938-3 Multimedia Content Description Interface—Part 3: Visual. Final Committee Draft*, ISO/IEC/JTC1/SC29/WG11, Doc. N4062, Mar. 2001.

[2] *MPEG-7 Visual Experimentation Model (XM), Version 10.0*, ISO/IEC/JTC1/SC29/WG11, Doc. N4063, Mar. 2001.

[3] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1966.

[4] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, and H. Shin, "An efficient color representation for image retrieval," *IEEE Trans. Image Processing*, vol. 10, pp. 140–147, Jan. 2001.

[5] G. M. Haley and B. S. Manjunath, "Rotation invariant texture classification using a complete space-frequency model," *IEEE Trans. Image Processing*, vol. 8, pp. 255–269, Feb. 1999.
[6] K. Jack, *Video Demystified*. Eagle Rock, VA: LLH Technology Publishing, 1996, ch. 3.
[7] C. Kenney, Y. Deng, B. S. Manjunath, and G. Hewer, "Peer group image enhancement," *IEEE Trans. Image Processing*, vol. 10, pp. 326–334, Feb. 2001.
[8] C. S. Li, J. R. Smith, V. Castelli, and L. Bergman, "Comparing texture feature sets for retrieving core images in petroleum aopplications," in *Proc. SPIE*, vol. 3656, San Jose, CA, 1999, pp. 2–11.
[9] W. Y. Ma and B. S. Manjunath, "A texture thesaurus for browsing large aerial photographs," *J. Amer. Soc. Inform. Sci.*, vol. 49, no. 7, pp. 633–648, May 1998.
[10] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.
[11] B. S. Manjunath, P. Wu, S. Newsam, and H. Shin, "A texture descriptor for browsing and image retrieval," *Int. Commun. J.*, vol. 16, pp. 33–43, Sept. 2000.
[12] P. Ndjiki-Nya, J. Restat, T. Meiers, J.-R. Ohm, A. Seyferth, and R. Sniehotta, "Subjective Evaluation of the MPEG-7 Retrieval Accuracy Measure (ANMRR)," ISO/WG11 MPEG Meeting, Geneva, Switzerland, Doc. M6029, May 2000.
[13] D. K. Park, Y. S. Jeon, C. S. Won, S.-J. Park, and S.-J. Yoo, "A composite histogram for image retrieval," in *Proc. ICME 2000*, vol. 1, July 2000, pp. 355–358.
[14] S.-J. Park, C. S. Won, and D. K. Park, "Core Experiments on MPEG-7 Edge Histogram Descriptor," ISO/WG11 MPEG Meeting, Beijing, China, MPEG Document M6174, July 2000.
[15] D. K. Park, Y. S. Jeon, C. S. Won, and S.-J. Park, "Efficient use of local edge histogram descriptor," in *Proc. ACM Workshop Standards, Interoperability, and Practice*, Los Angeles, CA, Nov. 2000.
[16] Y. M. Ro and H. K. Kang, "Hierarchical rotational invariant similarity measurement for MPEG-7 homogeneous texture descriptor," *Electron. Lett.*, vol. 36, no. 15, pp. 1268–1269, 2000.
[17] Y. M. Ro, "Matching pursuit: Contents featuring for image indexing," in *Proc. SPIE*, vol. 3527, 1998, pp. 89–100.
[18] Y. M. Ro and K.-W. Yoo, "Texture featuring and indexing using matching pursuit in radon space," *Proc. IEEE Int. Conf. Image Processing (ICIP'99)*, pp. 580–584.
[19] A. Yamada *et al.*, "Visual program navigation system based on spatial distribution of color," in *Proc. ICCE 2000*, June 2000.

B. S. Manjunath (M'91) received the B.E. degree in electronics (with distinction) from the Bangalore University, Bangalore, India, in 1985, the M.E. degree (with distinction) in systems science and automation from the Indian Institute of Science in 1987, and the Ph.D. degree in Electrical Engineering from the University of Southern California, Los Angeles, in 1991.

He joined the Electrical and Computer Engineering Department, University of California at Santa Barbara, in 1991. His current research interests include multimedia databases, digital libraries, image/video data hiding, and data mining. He has been an active participant in the development of the MPEG-7 standard.

Dr. Manjunath is currently an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and was a Guest Editor for the January 2000 Special Issue on Image and Video Processing for Digital Libraries. He was a recipient of the National Merit Scholarship (1978–1985) and was awarded the Bangalore University Gold Medal for the best graduating student in electronics engineering in 1985.

Jens-Rainer Ohm (M'92) received the Dipl.-Ing. degree in 1985, the Dr.-Ing. degree in 1990, and the Habil. degree in 1997, all from the Technical University of Berlin (TUB), Berlin, Germany.

From 1985 to 1990, he was a Research and Teaching Assistant with the Institute for Telecommunications, TUB, and from 1990 to 1995, he performed work within government-funded research projects on image and video coding at TUB. From 1992 to 2000, he has also served as a Lecturer on topics of digital image processing, coding, and transmission at TUB. From 1996 to 2000, he was Project Manager/Coordinator at the Image Processing Department, Heinrich–Hertz Institute (HHI), Berlin, Germany. He was involved in research projects on motion-compensated, stereoscopic, and 3-D image processing, image/video coding, and content description for image/video database retrieval. Since 1998, he has been a participant of the MPEG Group, where he has been active in the development of MPEG-4 and MPEG-7 standards. Since mid-2000, he has been chairing the Institute for Communications Engineering, Aachen University of Technology, Aachen, Germany. His current research activities are in the areas of multimedia communications, multimedia signal processing/coding, and services for mobile networks. He has authored a German-language textbook on image/video coding and numerous papers in his fields of research.

Vinod V. Vasudevan (M'97) received the B.Tech degree in 1988, the M.Tech degree in 1990, and the Ph.D degree 1994, all in computer science and engineering, from the Indian Institute of Technology, Kharagpur, India.

He is the Co-Founder and Chief Technology Officer of NewsTakes Inc., Burlingame, CA, a company providing MPEG-7 based products and services for delivering media-rich content to multiple devices and networks. Previously, he was a Research Scientist with NTT Basic Research Laboratories, Japan, and a Member of Research Staff at Kent Ridge Digital Labs, Singapore. His current research interests include video analysis and description, multimedia delivery, and content repurposing. He has been an active participant in the development of the MPEG-7 Standard, has co-chaired various MPEG *ad-hoc* groups during 1998–2000, and has authored several research papers.

Dr. Vasudevan was awarded Outstanding Achievement in a Paper Series Award by IEICE of Japan for his work on Active Search.

Akio Yamada received the Ph.D. degree in information electronics science from Nagoya University, Japan, in 1993.

He joined C&C Research Laboratories, NEC Corporation, Kawasaki, Japan, in 1993, where he is an Assistant Manager of the Multimedia Research Laboratories. He was also with NEC Research Institute Inc., Princeton NJ, in 1997 and 1998. His research interests include image coding, image analysis, image retrieval, 3-D depth estimation, and several image processing technologies. He has been involved in the MPEG project, especially for Phase 4 and Phase 7, and has been a co-editor of the MPEG Phase 7 Visual standard. His current works are developing a multimedia asset management system, automatic object extraction and tracking system, video sequence analysis, including object recognition, and panoramic picture processing.

Dr. Yamada received the ITE Niwa–Takayanagi Best Paper Award in 1994 and the IEICE Yong Engineer Award in 1996.