INTERNATIONAL STANDARD

ISO/IEC TR 15938-8

First edition
2002-12-15

AMENDMENT 3
2007-12-15

Information technology — Multimedia content description interface —

Part 8: Extraction and use of MPEG-7 descriptions

AMENDMENT 3: Technologies for digital photo management using MPEG-7 visual tools

Technologies de l'information — Interface de description du contenu multimédia —

Partie 8: Extraction et utilisation des descriptions MPEG-7

AMENDEMENT 3: Technologies pour la gestion des photos numériques à l'aide des outils visuels MPEG-7

Reference number
ISO/IEC TR 15938-8:2002/Amd 3:2007(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 3 to ISO/IEC TR 15938-8:2002 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia Information*.

Information technology — Multimedia content description interface —

Part 8:
Extraction and use of MPEG-7 descriptions

AMENDMENT 3: Technologies for digital photo management using MPEG-7 visual tools

Add after subclause 4.2.3.3:

4.2.3.4 Dominant Color Temperature

4.2.3.4.1 General

This subclause provides an advanced use scenario of the Dominant Color descriptor. The Dominant Color Temperature is a variation of Dominant Color, but suitable to implement perceptual similarity based retrieval. Images usually have one of a few dominant color temperatures perceived by users when they look at them. Dominant Color Temperatures enable users to search for images in scenarios such as query by example or query by value, and for image browsing regarding their color temperature. It can be useful for users who want to find images which look similar according to color temperature rather than to find images which have similar color regions.

4.2.3.4.2 Use scenario

Dominant Color Temperatures can be used in query by example and query by value search scenarios. Examples of such queries are depicted in Figure AMD3.1. In a query by example a user inputs an example image or draws a colored sketch (query by sketch) and the search application returns the most similar images regarding their color temperature. In a query by value a user chooses a temperature value, and the system retrieves images in which the appearance of color temperature is closest to the user choice.

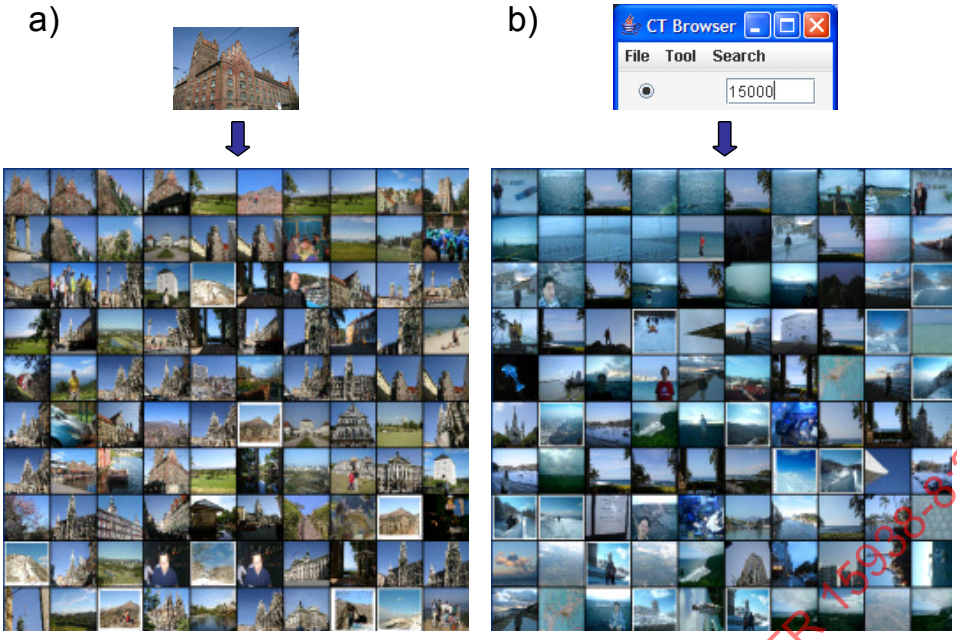


Figure AMD3.1 — Examples of image retrieval using Dominant Color Temperatures: a) query by example; b) query by color temperature value given in kelvins

4.2.3.4.3 Feature extraction

The Dominant Color Temperature, which consists of a maximum of eight pairs of color temperature and percentage, is obtained by the following steps.

1. Get RGB color values and percentages of dominant colors from a Dominant Color descriptor instance.

2. Convert each dominant color value from RGB to color temperature using the relevant method specified in the feature extraction method of Color Temperature descriptor [subclause 6.9.1.1]. The number of obtained color temperatures cannot, therefore, exceed the number of dominant colors in the Dominant Color descriptor instance. The colors that do not have significant color temperature (colors having luminance values below the luminance threshold specified in the extraction method of the Color Temperature descriptor) should be omitted.

3. Use the obtained color temperatures and their percentages given by the Dominant Color descriptor instance in queries: query by example, query by color temperature value, ranking search results, and others.

4.2.3.4.4 Similarity matching

The similarity is based on a distance function which is defined as an integral of absolute difference between two percentage distributions of dominant color temperature. The percentage distributions of dominant color temperature should be obtained first in the following steps:

1. Convert color temperature values T_i of Dominant Color Temperature description to Reciprocal Megakelvin scale RT_i [MK^{-1}] = 1000000/T_i [K].

2. Sort, in ascending order, the dominant color temperatures expressed in reciprocal scale.

3. Create the percentage distribution of dominant color temperature $D_i(RT_i)$ using the following equations:

$D(RT) = 0$ for $RT < RT_0$;

$D(RT) = p_0 + p_1 + ... + p_{i-1}$ for $RT_{i-1} \le RT < RT_i$, $1 \le i \le n\text{-}1$;

$D(RT) = p_0 + p_1 + ... + p_{n-1}$ for $RT \geq RT_{n-1}$;

Where:

n – number of dominant color temperatures;

RT_0 , RT_1 , ... , RT_{n-1} – sorted dominant color temperatures;

p_0 , p_1 , ... , p_{n-1} – percentages.

Figure AMD3.2 shows an example of a dominant color temperature distribution.

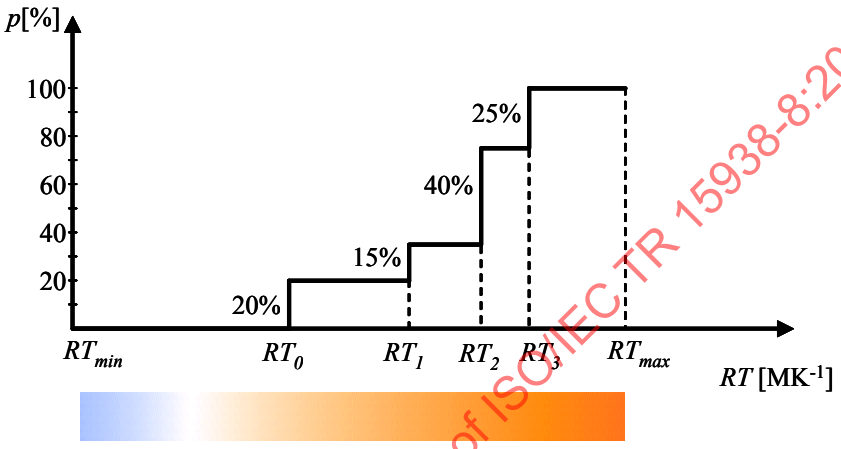


Figure AMD3.2 — Example of cumulative dominant color temperature distribution

The proposed distance function is given by the following equation, which is an integral of difference between two color temperature distributions.

$$dist = \int_{RT_{\min}}^{RT_{\max}} |D_1(RT) - D_2(RT)| dRT$$

This expression is equivalent to the geometrical area bounded by the two distributions. An example of distance calculation is depicted in Figure AMD3.3, where the distribution distances are shown graphically on distribution diagrams.

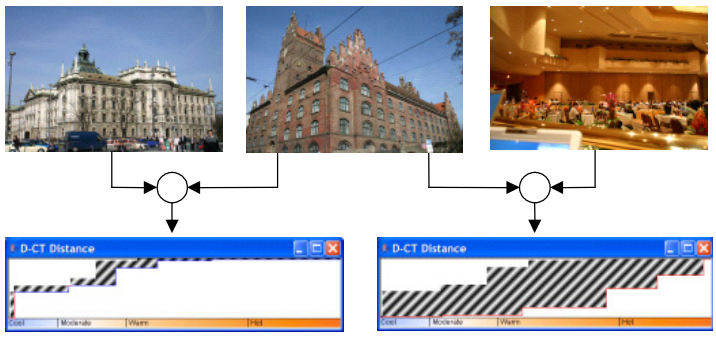


Figure AMD3.3 — Example of distance calculation

The distance function presented in the above equation can be efficiently implemented using the following steps:

1. Input: two percentage distributions of dominant color temperature:

RT1, *D1* – tables of temperature and percentage distribution for image 1,

RT2, *D2* – tables of temperature and percentage distribution for image 2;

2. Initialize: *dist*=0, $x_1 = RT_{min}$;

3. Take the next minimum temperature value t_{curr} from tables *RT1*, *RT2*, and let $x_2 = t_{curr}$;

4. Find in *D1*, *D2* the lower bound y_1 and the upper bound y_2 of the rectangle corresponding to the current x_1, x_2 coordinates;

5. $dist = dist + (x_2 - x_1)(y_2 - y_1)$;

6. $x_1 = x_2$;

7. If all values from tables *D1*, *D2* have been taken then return *dist* else go to step 3.

The tables used as an input to the algorithm above are obtained from the percentage distributions of dominant color temperature in the following way: $RTX[i] = RT_i$, $DX[i]=D(RT_i)$, for $0 \le i \le n$, where *X* stands for image 1 or 2.

In the case of query by color temperature value, the same distance function can be used, by assuming that the query value given by the user is a single dominant color temperature with a percentage of 100%. Although in this case, the distance function can be simplified to the following:

$$\Delta RT = \sum_{i=0}^{n-1} |RT_i - RT_{REF}| p_i$$

where RT_{REF} is the value of the query color temperature, RT_i are dominant color temperatures, p_i are percentages, and *n* is the number of dominant color temperatures in image.

4.2.3.4.5 Condition of usage

The same restrictions are applied as for the Dominant Color descriptor. Additionally, Dominant Color Temperatures cannot be used for very dark images of which all dominant colors have luminance values below the luminance threshold specified in the extraction method of Color Temperature descriptor.

Add after subclause 4.7:

4.8 High-level use scenarios

4.8.1 Content based Image retrieval

4.8.1.1 General

Content-based image retrieval gives an efficient and easy way of managing and retrieving digital images from enormous digital contents. In content-based image retrieval, there are two representative methods. One is a query by example, where a user selects a similar image to those expected for a query. The other is query by

sketch, in which a user must draw a sketch and use it as a query. Since a seed picture is needed, some mechanism to assist users finding query image itself is required in the former scenario. One possible solution is to combine text-based image retrieval or query by sketch as a pre-processing step of query by example.

4.8.1.2 Query within region of interest (ROI)

4.8.1.2.1 General

This section provides a usage scenario to enable users to dynamically retrieve photographs with similar Region of Interest (such as background) in image space. Region-based image retrieval can be implemented by portioning an image into several small regions and assigning a StillRegionFeatureDS for each of them. However, in practice, such an approach is difficult as it requires prior segmentations that are often subjective and may depend on a particular query. The ROI-based photo retrieval gives users the benefit of defining ROI when making a query. Although query by example is very useful for image retrieval, one may want to retrieve photos with similar backgrounds. In other words, if the scenery is well known or quite beautiful, people tend to take pictures with the same background but different persons. For those photos, it will be more efficient to retrieve the photos by matching the background regions only. In this scenario, the user can select the region that he wants to retrieve in particular and send it to the system as a query.

4.8.1.2.2 Use Scenario

Figure AMD3.4 shows the flow of the proposed query method. The user first selects a query image. In the query image, the user selects a ROI by selecting local regions (shown in blue). The ROI is used as a query image for retrieval. Figure AMD3.5 shows the example of image retrieval within a ROI.

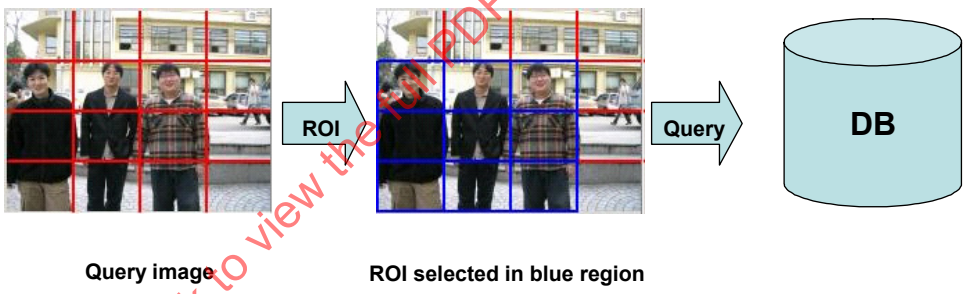


Figure AMD3.4 — Flow of query by ROI

Figure AMD3.5 — Retrieval of image within ROI

4.8.1.2.3 Tools to be used

StillRegionFeatureDS or VideoSegmentFeatureDS is used for this scenario. Among the several elements included in these DSs, the Edge Histogram descriptor and the Color Layout descriptor should be instantiated to implement the functionality of ROI-based retrieval. For video retrieval, shots are extracted from the video sequence and for each shot, localized features from the specific region are extracted. Then, the ROI is used as a query for video retrieval.

4.8.1.2.4 Feature Extraction

ROI-based retrieval can be implemented by extracting localized features from the specified region. The extraction process of the localized feature from the instances of two mandatory description tools, Color Layout and Edge Histogram, is described in this subclause. Figure AMD3.6 illustrates this process. From the Edge Histogram Descriptor one can obtain a localized edge distribution in each 4 x 4 local rectangular region. From the Color Layout descriptor, one can obtain an 8 x 8 region-based DCT: by performing inverse quantization and taking the 8 x 8 inverse DCT (as described in subclause 4.2.5.2.3), we can obtain average color values for 8 x 8 local rectangular regions. Feature extraction of each descriptor is defined in ISO/IEC 15938-3, MPEG-7 Visual. As in Figure AMD3.6, a combination of the Edge Histogram Descriptor and Color Layout Descriptors can be used for the rectangular region-based query-by-ROI.

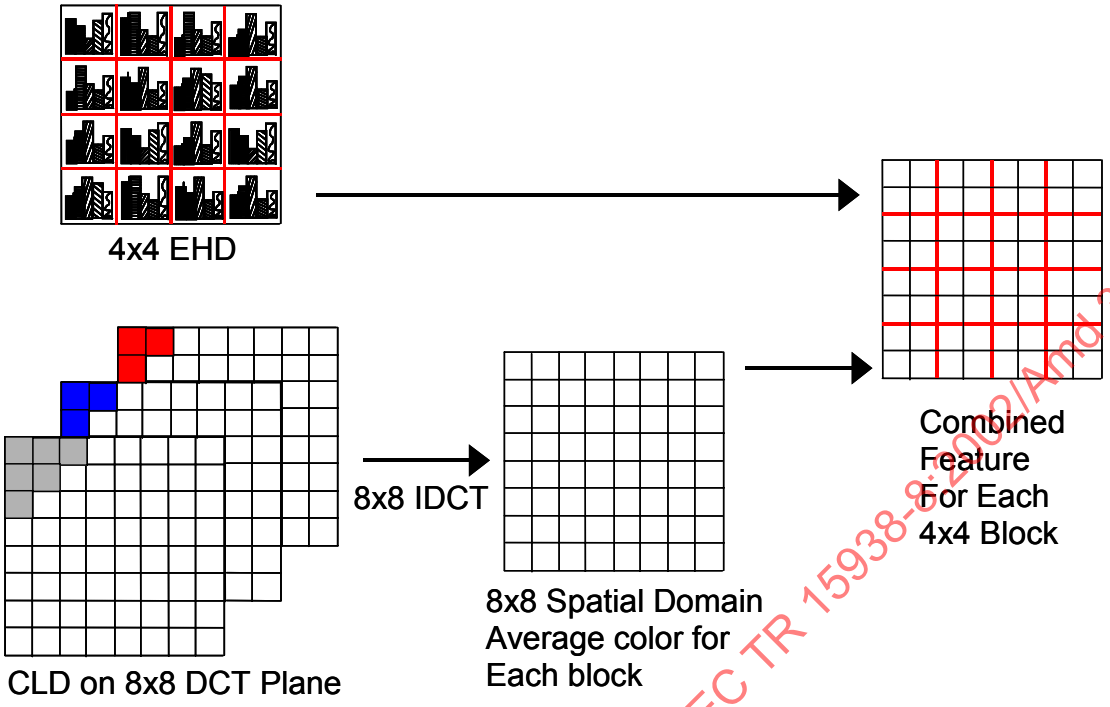


Figure AMD3.6 — 4x4 block-based "Query-by-ROI" with Edge Histogram Descriptor and Color Layout Descriptor

4.8.1.2.5 Similarity Matching

For the Color Layout descriptor, we can take an 8 x 8 inverse DCT for the quantized DCT coefficients of Y, Cr, and Cb. Then, we have representative color values for 8 x 8 blocks of the image. These block-wise color values are combined with the edge histogram bins for each 4 x 4 image region (see Figure AMD3.7). Thus, each rectangular image region of the (4 x 4) Edge Histogram descriptor blocks includes 4 (2 x 2) color blocks obtained by the inverse 8 x 8 DCT of the Color Layout descriptor. Now, a combination of the color and edge information in each of the (4 x 4) rectangular image regions will form a feature vector for the rectangular region-based similarity matching.

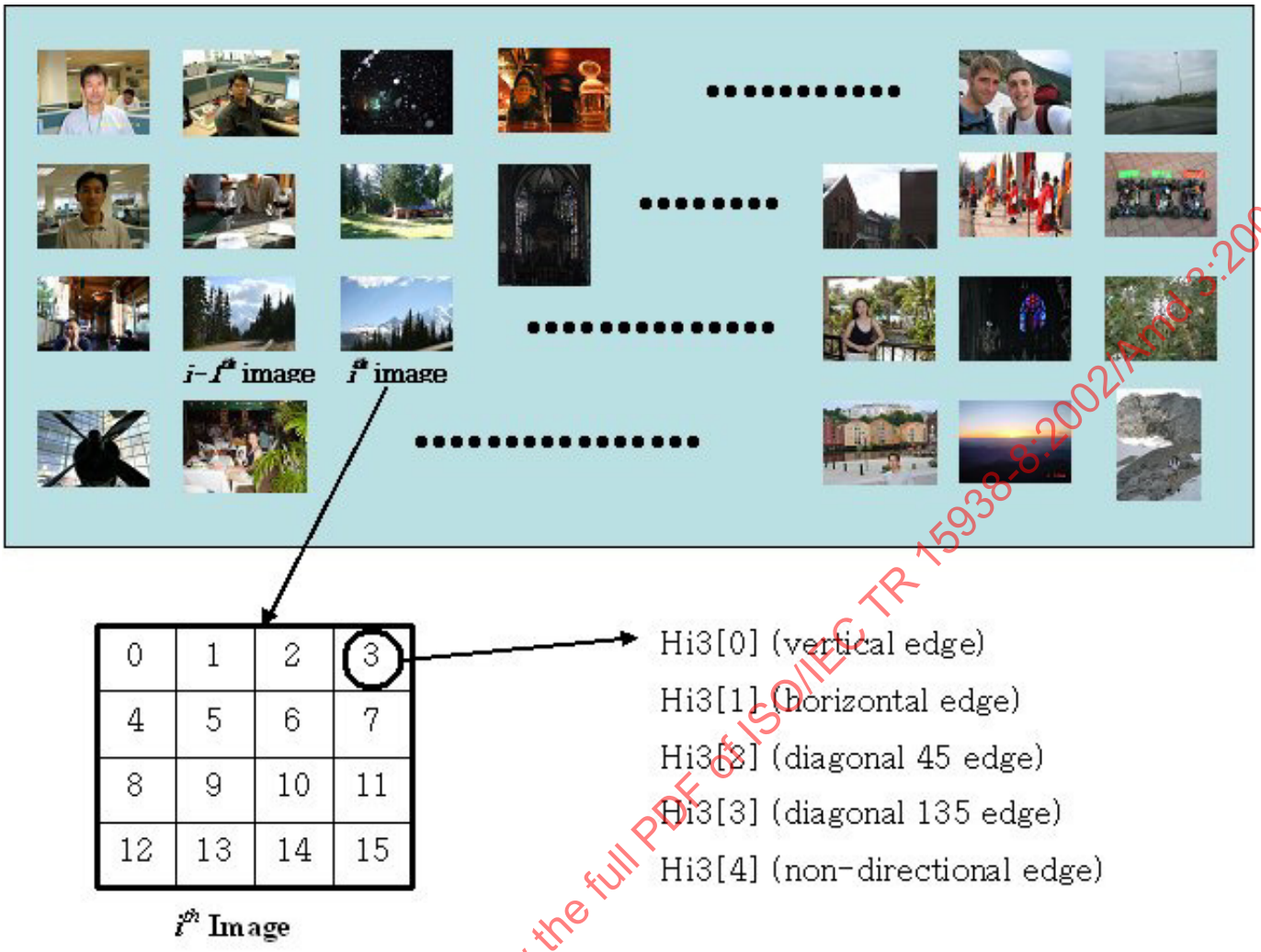


Figure AMD3.7 — Parameter value example of EHD

Figure AMD3.7 shows an example of parameter values when using the EHD for matching blocks. When the total number of images is N, the *j*th(*j*=0,1,2 ... 15) block of the *i*th(i=0,1,2,...N) image has five types(0°, 45°, 90°, 135°, non-directional) of edge value. If we represent these edge value as *k* (*k*=0,1,2,3,4), the parameter value $H_{ij}[k]$ is the *k*th edge value of the *j*th block of the *i*th image. For a query image Q, the edge value of the selected sub-image is $H^{Q}[k]$.The local distance of Edge Histogram LD^{EHD}_{ij} is as follows.

$$\mathrm{LD}^{\mathrm{EHD}}{}_{\mathrm{ij}} = \sum_{k=0}^{4} | H^{Q}[k] - H_{ij}[k] | \qquad \text{(AMD1)}$$

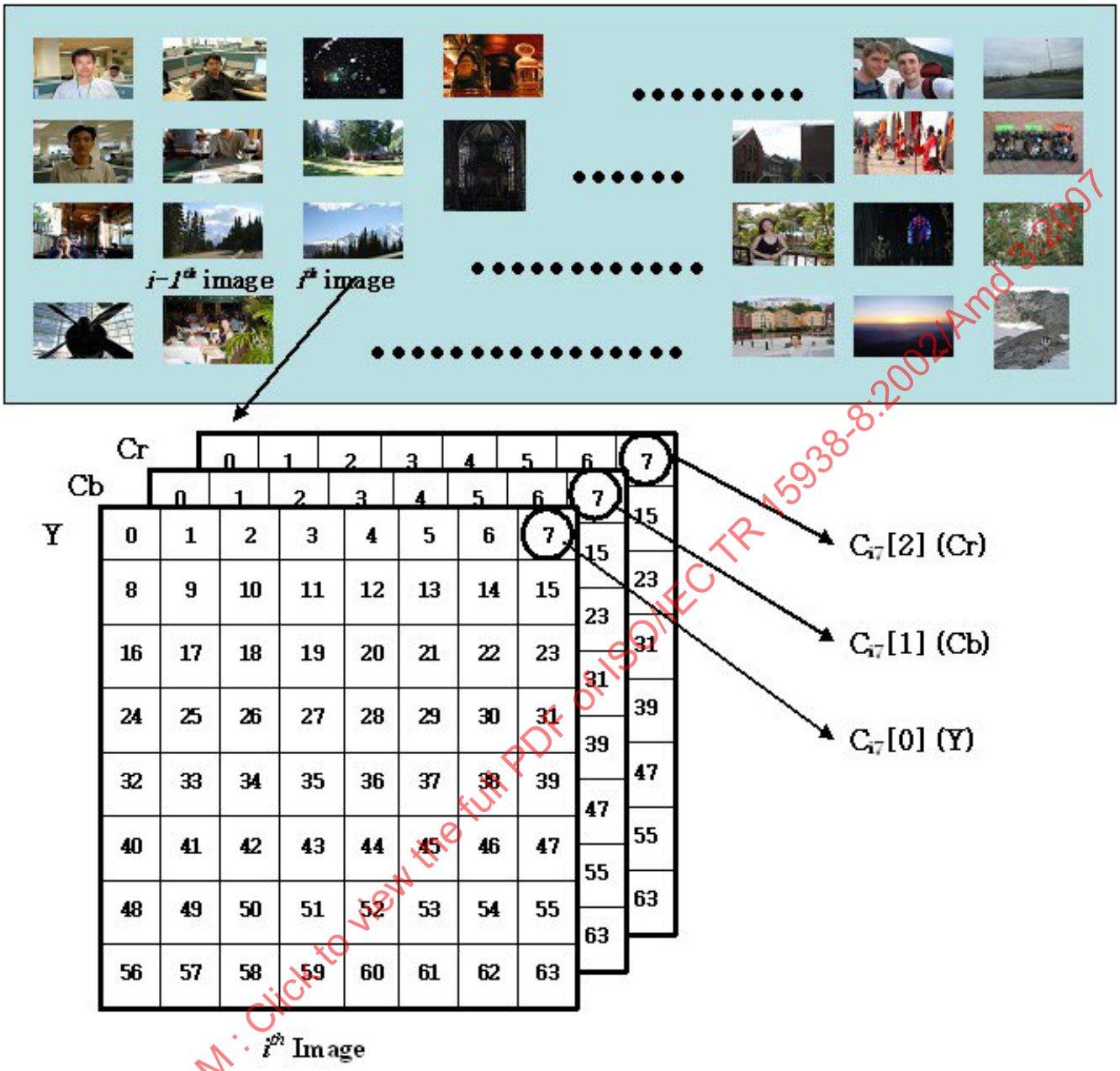


Figure AMD3.8 — Parameter of Inverse DCT Color Layout descriptor

Figure AMD3.8 shows the parameters of the inverse DCT Color Layout descriptor. The total number of images is N. We group 8 X 8 image blocks into 4 X 4 blocks. Newly grouped blocks can be labeled as β where each block consists of 4 blocks(β=0,1,2,3). Each Y, Cb, Cr is labeled as α(α=0,1,2) (α=0 for Y, α=1 for Cb, α=2 for Cr). Parameter value is $C_{ij\beta}[\alpha]$ for jth block(sub image) of ith image. $C_{ij\beta}[\alpha]$ represents color value α of β sub-block of jth block of ith image. $C^{Q}_{\beta}[\alpha]$ are parameter values of the query image Q. The local distance of Color Layout descriptor of jth sub image of ith image can be obtained as follows.

$$\mathrm{LD}^{\mathrm{CLD}}{}_{\mathrm{ij}} = \sum_{\alpha=0}^{2}\sum_{\beta=0}^{3}\frac{\left|C_{\beta}{}^{Q}[\alpha] - C_{ij\,\beta}[\alpha]\right|}{3} \qquad \text{(AMD2)}$$

The distance values are then scaled, so that the maximum values of LD^{EHD}_{ij} (AMD1) and LD^{CLD}_{ij} (AMD2) for all images in the database, are normalised to 1. The combined distance CD_{ij} can be obtained from these normalized distances, ND^{EHD}_{ij} and ND^{CLD}_{ij}, as follows:

$$CD_{ij} = \frac{ND^{EHD}_{ij} + ND^{CLD}_{ij}}{2} \qquad \text{(AMD3)}$$

4.8.1.2.6 Condition of Use

In order to use ROI there shall be rectangular regions and there is a restriction on the setting of rectangular regions as 4 X 4.

4.8.1.2.7 DDL instance examples

```
<Mpeg7 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mpeg:mpeg7:schema:2004">
    <DescriptionUnit xsi:type="StillRegionType">
        <VisualDescriptionScheme
              xsi:type="StillRegionFeatureType">
           <ColorLayout>
              <YDCCoeff>40</YDCCoeff>
              <CbDCCoeff>34</CbDCCoeff>
              <CrDCCoeff>30</CrDCCoeff>
              <YACCoeff5>16 12 15 12 17</YACCoeff5>
              <CbACCoeff2>12 17</CbACCoeff2>
              <CrACCoeff2>12 14</CrACCoeff2>
           </ColorLayout>
           <EdgeHistogram>
           <BinCounts>
              2  6  4  4  2  1  7  5  3  2  1  6  4  2  2  2  5  4
              5  3  1  5  5  6  5  2  6  5  4  4  1  6  4  4  4  0  6  3  5
              2  1  5  5  6  6  4  2  3  6  7  3  2  5  5  7  3  2  4  4  7
              1  5  6  4  6  1  5  7  4  5  1  6  4  6  5  1  3  4  7  6
           </BinCounts>
           </EdgeHistogram>
        </VisualDescriptionScheme>
    </DescriptionUnit>
</Mpeg7>
```

4.8.2 Grouping Technologies

4.8.2.1 Situation-based clustering

4.8.2.1.1 General

A simple but very effective structure is to group images by the occasion on which they were taken. This is natural for the user since they will often remember the context of the situation much better than a date, time or explicit label attached to the picture. It is possible to automatically cluster images into such “situations” by using MPEG-7 visual description, together with the time stamp of the image. Based on the assumption that each situation is contiguous in time, the organisational structure can be represented by the time-sequence of images, with a flag or marker to indicate the boundaries between situations (*cf*. Figure AMD3.9). This provides the user with a simple, intuitive and effective means to browse through their collection, without placing any

additional burden on them to spend time organising it. Two methods are presented for situation based clustering.

Boundaries (vertical bars) are inserted between adjacent images in the sequence to denote the grouping.

Figure AMD3.9 — One representation of the grouped sequence of images

4.8.2.1.2 Use scenario

This kind of clustering can easily be implemented in traditional photo-browsing software applications. For the user, it is very simple to use – the extraction and matching of MPEG-7 descriptors and detection of the boundaries is fully automatic, so the tool is essentially "one-click". Of course, some users may choose to adjust and refine the automatic output to match their individual preferences. This process would still be far easier than organising all the photos manually.

The clustering information can be used to access and manipulate the image content in a variety of ways:

- Browsing:
  - Display a cluster of images per page, or
  - Display a single thumbnail / icon for each cluster
- Annotation
  - User can easily assign a single label to all the images in a cluster
- Sharing:
  - User can select images by cluster and…
  - Print
  - Copy
  - Upload to website

4.8.2.1.3 Method1: Simple Linear Clustering

4.8.2.1.3.1 General

This method achieves good clustering performance with minimal complexity. The additional computation (after extracting and matching MPEG-7 visual descriptors) consists of a simple weighted linear summation. It is therefore well-suited to applications where MPEG-7 descriptors have been extracted from images but resources are not available for higher-level processing (for example, in low-complexity devices). The input parameters to the algorithm are also simple and therefore easy to adapt - for example, to different applications or user preferences.

4.8.2.1.3.2 Tools to be used

Six tools defined in ISO/IEC 15938-3 shall be instantiated in StillRegionFeatureDS:

- Dominant Color (DC)
- Scalable Color (SC)
- Color Layout (CL)
- Color Structure (CS)
- Homogeneous Texture (HT)
- Edge Histogram (EH)

Also, capturing date/time information should be included. If an image is encoded in Exif file format (JEITA CP-3451), this information can be obtained from the Exif header.

- EXIF DateTime tag (ID36867)

Alternatively, the same information can be captured using

- CreationInformation/CreationCoordinates/Date (mpeg7:TimeType)

4.8.2.1.3.3 Clustering Algorithm

The images are ordered by their time stamps and each potential boundary in the sequence is evaluated in turn. To determine the presence or absence of a boundary, a number of pair-wise comparisons are made amongst images lying in a window either side of the transition. This neighbourhood and the comparisons used are illustrated in Figure AMD3.10.

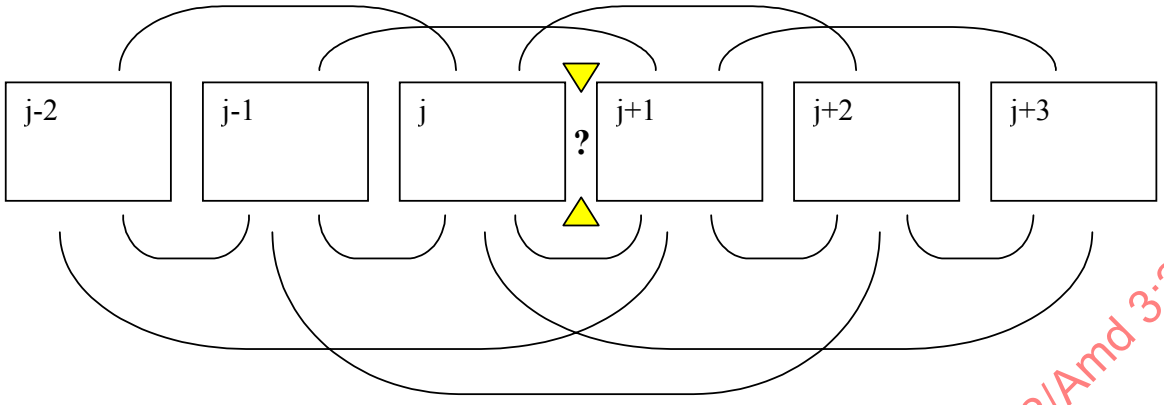


Figure AMD3.10 — Neighbourhood comparisons evaluated to determine if a boundary is present

Comparison of images consists of computing the descriptor distances (by the respective methods suggested in ISO/IEC 15938-8 TR) and calculating the time difference. The latter is measured on a logarithmic scale, to compress the range of this feature and allow meaningful comparisons. Time distance is therefore defined as:

$$D_T(i,i+1) = \ln\left(10^{-5} + \left(T_{i+1} - T_i\right)\right)$$

The unit of time used for T_i is days. The natural logarithm is applied to normalise the range of time distances – potential time differences will vary over several orders of magnitude. After this transformation, the variation of the time distance is comparable to the remaining features. The constant, 10^{-5}, meanwhile, chooses the minimum scale of the distance – just under one second, in this case. It also ensures that $\ln(0)$ does not occur.

The input to the algorithm includes the first-, second- and third-order distances in a short time interval around the boundary to be tested. Here "first-order" refers to the difference, for any given feature, between two images that are adjacent in the sequence – *i.e.* $D_F(i,i+1)$. A second-order distance is the difference between two images that are separated by one other image – *i.e.* $D_F(i,i+2)$. Similarly, a third-order distance is the difference between two images that are separated by two other images – *i.e.* $D_F(i,i+3)$. The total measurement of difference between images j and $j+1$ is now:

$$D(j,j+1) = \sum_F \left\{ \sum_{i=-2}^{2} \alpha_{Fi} D_F(j+i, j+i+1) + \sum_{i=-2}^{1} \beta_{Fi} D_F(j+i, j+i+2) + \sum_{i=-2}^{0} \gamma_{Fi} D_F(j+i, j+i+3) \right\}$$

This is a summation over a set of 12 distance measurements for each of 6 visual features (the outer summation being over the set of features, F). For time difference, only the first order distances are used, adding 5 more distance measurements, to give a total set of 77 numbers. These are weighted by 77 weights α, β, γ, the recommended values of which are given in Table AMD3.1.

Table AMD3.1 —Weights for distance calculation. (Feature distances are first normalized for zero mean and unit variance)

	Dominant Color	Scalable Color	Color Layout	Color Structure	Homogeneous Texture	Edge Histogram	Time
α_0	0,0583	0,2598	0,4546	0,2661	-0,0718	0,4890	2,8952
α_1	0,1976	-0,0077	-0,0986	-0,3279	-0,1370	-0,3108	-0,2911
α_2	-0,0425	0,1117	0,0543	0,0594	-0,0089	-0,1642	-0,0035
α_{-1}	0,2718	-0,1835	-0,0640	-0,1153	0,0102	-0,3534	-0,3748
α_{-2}	0,0085	-0,0259	-0,0539	-0,1419	-0,0725	0,0951	-0,0786
β_{-1}	-0,0249	-0,0107	0,4662	0,3828	0,0567	-0,2351	
β_0	0,1718	-0,0788	-0,0086	0,2190	0,2653	0,2186	
β_{-2}	0,0958	-0,2618	-0,0520	-0,0652	-0,0496	0,1157	
β_1	0,2785	0,0072	-0,3648	-0,1872	-0,0611	0,1439	
γ_{-1}	-0,1955	0,1203	-0,0767	-0,0567	0,0148	0,1178	
γ_{-2}	0,0324	0,1808	-0,2327	0,2665	0,0167	0,2029	
γ_0	-0,1199	0,0196	0,0477	0,1841	0,0288	0,1436	

The output, *D*, is the real-valued indicator of boundary confidence – *i.e.*, higher values of *D* indicate a stronger belief that there is a boundary at the candidate position. A binary boundary indicator is obtained by comparing this number to a threshold. The threshold may be adjusted for sensitivity depending on the image collection, the particular application, or the preferences of the user. A value of around 3.35 could be recommended to produce a good balance between false positives (mistakenly detected boundaries) and false negatives (missed boundaries).

4.8.2.1.4 Method2: Clustering based on Visual Semantic Hints

4.8.2.1.4.1 General

The proposed method achieves good clustering performance on similar situations based upon the visual semantic hints. If the visual semantic hints are used for adaptive feature selection, they eventually help to reduce computational complexity while achieving reasonable clustering performance. For example, a low performance device like a mobile phone, which can only extract a limited number of MPEG-7 descriptors, can apply this method while maintaining a reasonable clustering performance.

4.8.2.1.4.2 Tools to be used

Seven tools defined in ISO/IEC 15938-3 shall be instantiated in StillRegionFeatureDS:

- Dominant Color (DC)
- Scalable Color (SC)
- Color Layout (CL)
- Color Structure (CS)
- Homogeneous Texture (HT)
- Texture Browsing (TB)
- Edge Histogram (EH)

Also, capturing date/time information should be included. If an image is encoded in Exif file format (JEITA CP-3451), this information can be obtained from the Exif header.

- EXIF DateTime tag (ID36867)

Alternatively, the same information can be captured using

- CreationInformation/CreationCoordinates/Date (mpeg7:TimeType)

4.8.2.1.4.3 Semantic Hint Extraction

We determine the weight of each visual feature using 'visual semantic hints', which are automatically extracted from a series of photos, in order to improve the Situation/View-based Photo clustering performance. The visual semantic hints of image represent the visual characteristics that are perceived by human visual system. The visual semantic hints used are as follows:

1) Colorfulness (CoF) hint: it represents degree of a visual sensation according to the purity of colors on photo. Figure AMD3.11 shows some exemplary photos with high degree of colorfulness.

Figure AMD3.11 — Exemplary photos with Colorfulness semantics.

To extract the CoF semantics, we utilize Scalable Color Descriptor.

$$\mathbf{F}_{SCD} = \{f_1, f_2, f_3, \cdots, f_j, \cdots, f_{N_{SCD}}\}, \text{ where } N_{SCD} \in \{16,32,64,128,256\}$$

where f_j represents the number of colors belonging to each HSV bin, i.e. it could be obtained by inverting Harr transform coefficients.

Whether a photo has CoF semantics is determined by using magnitude of the HSV bins. The colorfulness of a color is represented by averaging the magnitude of each HSV bin. Thus power of the magnitude is represent the CoF semantics. The CoF semantics of a photo is also set to high or low CoF for simple way. It is defined as,

$$CoF = \begin{cases} high, & \frac{1}{N_{SCD}} \sum_{j=1}^{N_{SCD}} |f_j|^2 > th_{CoF} \\ low, & otherwise \end{cases}$$

where th_{CoF} is a threshold to detect whether the power of the magnitude is sufficient to represent the CoF semantics. The threshold, th_{CoF}, was heuristically set to an average of 'colorfulness' values in given database. The th_{CoF} was set to 13.56.

2) Color Coherence (CoC) hint: it represents degree of a visual sensation according to spatial coherency of colors on photo. Figure AMD3.12 shows some exemplary photos with high degree of color coherency.

Figure AMD3.12 — Exemplary photos with Color Coherence semantics

To extract the CoC semantics, we utilize Dominant Color Descriptor.

$$\mathbf{F}_{DCD} = \{(\mathbf{c}_j, p_j, u_j), s\}, \text{ where } j = 1,2,3,\cdots,N_{DCD}$$

where N_{DCD} is the number of dominant colors. Each dominant color values $\mathbf{c}_j$ is a vector of corresponding color space component values. The percentage p_j (normalized to a value between 0.0 and 1.0) is the fraction of pixels in the image corresponding $\mathbf{c}_j$. The optional color variance u_j describes the variation of the color values of the pixels in a cluster around the corresponding representative colors. The spatial coherency s is a single number that represents the overall spatial homogeneity of the dominant colors in the images.

Whether a photo has CoC semantics is determined by using the percentage of each dominant color and the spatial coherency. To capture the human visual perception about the CoC semantics of photo, the CoC semantics is captured by the number of dominant colors over large portion of image and high spatial coherency. The CoC semantics of a photo is set to high or low CoC for simple way. It is defined as,

$$CoC = \begin{cases} high, & \left\{\sum_{j=1}^{N_{DCD}} B\left(p_j > th^1_{CoC}\right)\right\} > th^2_{CoC} \text{ and } s > th^3_{CoC} \\ low, & otherwise \end{cases}$$

where $B(p_j > th^1_{CoC})$ is 1 if $(p_j > th^1_{CoC})$ is true and it is 0 otherwise. th^1_{CoC} is a threshold to detect whether each dominant color take large portion of image region. th^2_{CoC} is a threshold to detect whether there are sufficient number of dominant colors that take large portion of image region. The th^3_{CoC} is a threshold to detect whether the dominant colors has spatially high coherence over the image region. The threshold, th^1_{CoC}, th^2_{CoC} and th^3_{CoC}, were heuristically set to 0.5, 7.0, and 0.31, respectively.

3) Level of Detail (LoD) hint: it represents degree of a visual sensation for objects on photo appearing more or less detailed. It is also one of the importance semantics since, for example, the photos of the mountains generally have higher level of detail than the close-up photos of human face. Figure AMD3.13 shows some exemplary photos with the LoD semantics. High LoD photos contain more detail such as edge rather than low LoD photos.

(a) (b)

Figure AMD3.13 — Exemplary photos with Level of Detail semantics, (a) photos with high Level of Detail semantics and (b) photos with low Level of Detail semantics

The basic idea to measure the LoD is that the photos of high LoD semantics have much detail description in their contents since the neighbor pixel values of them are abruptly and frequently changed. This means the photo has much high frequency components. On the other hands, the photos of low LoD semantics have relatively small detail description in their contents.

Nowadays, JPEG compression is very popular in digital photographs since it does not much degraded image quality, and also drastically reduces the file size of the photo. In order to measure LoD semantics of a photo, we define 'a relative compression ratio per pixel'. In general, the photos of high LoD semantics may have lower compression ration than the photos of low LoD semantics since they have relatively smaller spatial redundancy among inter-pixels.

In JPEG compression, a loss is caused by quantization. It is common that each photo may have compressed with different quantization table. Thus before extracting LoD semantics from photos, all photos to be clustered should be decompressed and then compressed with the same quantization table.

The file size of the JPEG compressed photo is obtained from its file header. And the file size is normalized by the total number of pixels. The LoD semantics is defined as,

$$LoD = \begin{cases} high, & \left. \frac{f_{fs}}{f_{iw} \times f_{ih} \times f_{cd}} \right|_{\mathbf{Q}} > th_{LoD} \\ low, & otherwise \end{cases}$$

where f_{fs} is the file size of photo, f_{iw} is image width of photo, f_{ih} is the image height of photo, and f_{cd} is the color depth of photo. **Q** is the given quantization table. Baseline JPEG quantization table, as seen in Figure AMD3.14, is recommended. The threshold, th_{LoD}, was heuristically set to an average of 'level of detail' values in given database. th_{LoD} was heuristically set to 0.499.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a) Luminance quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b) Chrominance quantization table

Figure AMD3.14 — Baseline JPEG quantization table for LoD semantic hint

4) Homogeneous Texture (HoT) hint: it represents degree of a visual sensation according to homogeneous texture on photo. HoT semantic hint can express how regular the textures of objects on a photo repeatedly form clumped region. Figure AMD3.15 shows some exemplary photos with high regular texture patterns.

Figure AMD3.15 — Example photos with Homogeneous Texture semantic hint

To extract the HoT semantic hint, we utilize Texture Browsing Descriptor.

$$\mathbf{F}_{TBD} = \{f_r, f_{d1}, f_{d2}, f_{s1}, f_{s2}\}$$

where f_r is regularity or structure of the texture. f_{d1} and f_{d2} are two dominant directions of the texture. f_{s1} and f_{s2} are two dominant scales to capture the coarseness of the texture.

The homogeneous texture of a photo can be represented by averaging the magnitude of each edge bin. Thus power of the magnitude represents the HoT semantic hint. The HoT semantic hint of a photo is also set to high or low for simple way. It is defined as,

$$HoT = \begin{cases} high, & f_r > th_{HoT} \\ low, & \text{otherwise} \end{cases}$$

where th_{HoT} is a threshold to detect sufficient regularity (or homogeneity) of the texture on image region. The threshold, th_{HoT}, was heuristically set to an average of 'homogeneous texture' values in given database. The th_{HoT} was set to 2.

5) Heterogeneous Texture (HeT) hint: it represents degree of a visual sensation how continuous or strong the boundaries on photo. Figure AMD3.16 shows some exemplary photos with strong boundaries.

Figure AMD3.16 — Example photos with Heterogeneous Texture semantic hint

To extract the HeT semantic hint, we utilize Edge Histogram Descriptor.

$$\mathbf{F}_{EHD} = \{f_1, f_2, f_3, \cdots, f_j, \cdots, f_{N_{bin}=80}\}$$

where f_j represents magnitude of edge bin.

The heterogeneous texture of a photo can be represented by averaging the magnitude of each edge bin. Thus power of the magnitude represents the HeT semantic hint. The HeT semantic hint of a photo is also set to high or low for simple way. It is defined as,

$$HeT = \begin{cases} high, & \frac{1}{N_{EHD}} \sum_{j=1}^{N_{EHD}} |f_j|^2 > th_{HeT} \\ low, & \text{otherwise} \end{cases}$$

where th_{HeF} is a threshold to detect whether the power of the magnitude is sufficient to represent the HeT semantic hint. The threshold, th_{HeT}, was heuristically set to an average of 'heterogeneous texture' values in given database. The th_{HeT} was set to 227.98.

4.8.2.1.4.4 Hierarchical Thresholding to Detect Situation Change

In a sequential series of photos, two adjacent photos have a variety of time and visual differences. The situation change is detected with hierarchical thresholding. Situation changes on a hierarchy are more frequent on the higher level of hierarchies, i.e., a situation group can be divided into more than two groups in the higher hierarchy. That is, the finest situation changes are made in the highest hierarchy. Figure AMD3.17 shows an illustration of the hierarchical situation detection

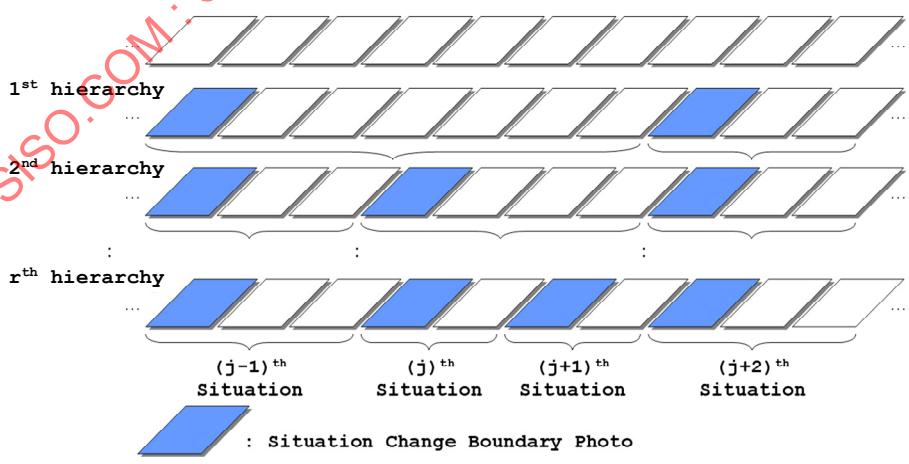


Figure AMD3.17 — Hierarchical situation change detection

Given multiple features, similarity distances between photos are measured. The time difference between the $(i)^{th}$ and the $(j)^{th}$ photos is measured as,

$$D_{time}(i,j) = \frac{\log\{\mathbf{F}_{time}(i) - \mathbf{F}_{time}(j) + C_{time}\}}{D_{time\_max}}$$

where $\log\{\mathbf{F}_{time}(i) - \mathbf{F}_{time}(j) + C_{time}\}$ is a time scale function, C_{time} is a constant to avoid zero for input of the log scale function. D_{time_max} is maximum time difference in a series of input photos to be clustered. Time difference is non-linear and the value of time similarity distance increases as the time difference increases. The value of the time similarity distance is scaled, so that it is less sensitive to the large time difference and is consistent in the same situation.

Content-based similarity between the $(i)^{th}$ and the $(j)^{th}$ photos is also defined as,

$$\mathbf{D}_{content}(i,j) = \{\mathbf{D}_f(i,j) \mid f \in \mathbf{F}\} = \{\Theta\{\mathbf{F'}_f(i) - \mathbf{F'}_f(j)\} \mid f \in \mathbf{F}\}$$

where Θ is a similarity measurement function, such as L1 or L2 norm distance measure, for a given low-level feature *f*.

Given the inter-photo similarity distances, similarity distance between the photos that belong to two adjacent situations, is measured. Figure AMD3.18 shows an example of determining whether the situation of the $(i)^{th}$ photo in the $\mathbf{S}_{(r-1)}$ has been changed or not. The $\mathbf{S}_{(r-1)}$ is the situation change results of the $(r-1)^{th}$ hierarchy.

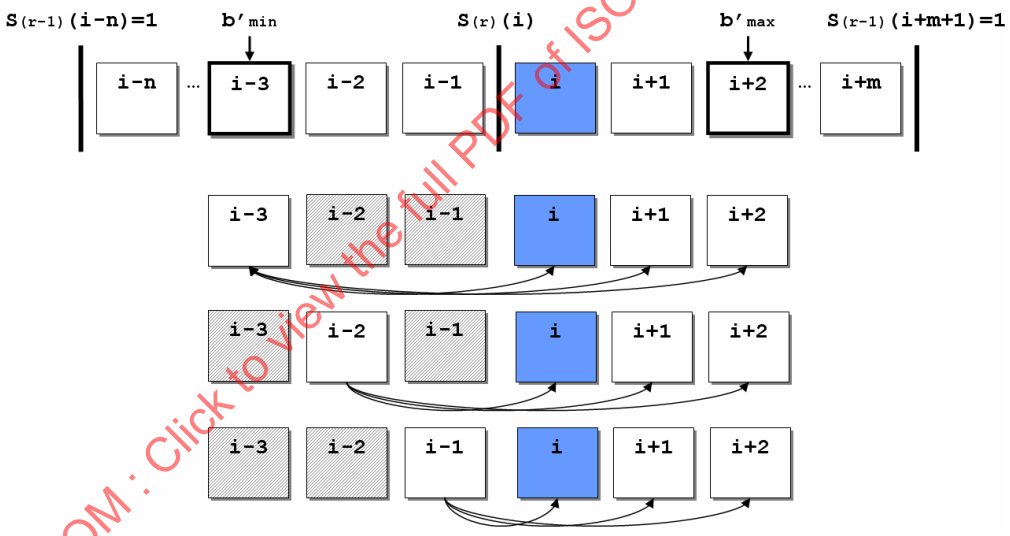


Figure AMD3.18 — Inter-situation similarity measurement

As shown in Figure AMD3.18, assume that a situation group is made from the $(i-n)^{th}$ photo to $(i+m+1)^{th}$ photo in the $(r-1)^{th}$ hierarchy. Then, inter-situation similarity is measured in a comparison bound, defined as,

$$B_r(i) = [b_{min}, b_{max}]$$

where b_{min} is the photo on minimum bound and b_{max} is the photo on maximum bound. Initially, the comparison bound is the situation change boundary in the $(r-1)^{th}$ hierarchy, e.g., in the Figure AMD3.18, b_{min} is $(i-n)^{th}$ photo and b_{max} is $(i+m+1)^{th}$ photo.

The comparison bound $B_r(i)$ is updated as finding the two most similar photos within the bound. This is to avoid unnecessary comparisons of photos that are not similar to the $(i)^{th}$ photo. Provided that the $(i)^{th}$ photo, the minimum bound b'_{min} is updated to the most similar one among the photos which were taken in prior to the

$(i)^{th}$ photo. Similarly, the maximum bound b'_{max} is updated to the most similar one among photos which were taken in posterior to the $(i)^{th}$ photo. The updated bound is measured as,

$$B'_r(i) = [b'_{\min}, b'_{\max}] = \left[\arg\min_j \{D(i,j) | b_{\min} \le j < i\}, \ \arg\min_j \{D(i,j) | i < j \le b_{\max}\} \right]$$

Given the bound $B'_r(i)$, the inter-situation similarity is measured. Similarity between two photos in the bound is measured as,

$$D_f(j_1, j_2)\big|_i = \frac{v_f(i) \times D_f(j_1, j_2)}{\sum_{f \in \mathbf{F}} v_f(i)}$$

where $v_f(i)$ is importance value of the feature *f*. The importance value for each feature was heuristically set as shown in the following Table AMD3.2. The default importance value for each feature is 1.0.

Table AMD3.2 — An importance value table for each feature and semantic hints

	v_{HTD} = 1.5	v_{EHD} = 2.7	v_{CLD} = 3.2	v_{CSD} = 2.0	v_{SCD} = 1.5	v_{DCD} = 2.5
HoT	High		-	-	-	-
HeT	-	High	-	-	High	-
LoD	-	-	Low	High	-	Low
CoF	-	-	-	High	High	-
CoC	-	-	High	-	-	High

The inter-situation similarity $Z_r(i)$ is finally measured by three terms of inter-photo similarity and it is as follows,

$$Z_r(i) = \alpha \cdot D_f(i, b'_{\min}) + \beta \cdot D_f(i, b'_{\max}) + \gamma \cdot \frac{\sum_{j=b'_{\min}}^{i-1} \left[\sum_{k=i}^{b'_{\max}} \left\{ \sum_{f \in \mathbf{F}} D_f(j,k) \right\} \right]}{M}$$

where α, β, and γ are importance values of each term. Without any prior knowledge, α, β, and γ can be set to 1.0. The first term is similarity distance between the $(i)^{th}$ photo and the minimum bound b'_{min}. The second term is similarity distance between the $(i)^{th}$ photo and the maximum bound b'_{max}. The third term is the sum of similarity distances among a group of photos which were taken in prior to the $(i)^{th}$ photo and a group of photos which were taken in posterior to the $(i)^{th}$ photo.

If there is a situation change in the $(i)^{th}$ photo, the first term would be relatively lower than that of no situation change case. The second term and the third term would be relatively higher than the first term.

In the $(r)^{th}$ hierarchy, the inter-situation similarity is determined by,

$$S_r(i) = \begin{cases} true, & Z_r(i) > th_r \\ false, & otherwise \end{cases}$$

where th_r is threshold to determine situation change on the $(i)_{th}$ photo. If the inter-situation similarity of the (i)th photo $Z_r(i)$ is bigger than the threshold, the $(i)^{th}$ photo is regarded as a situation change.

The threshold value decreases as the hierarchy increases. The lower threshold makes more situation boundaries. The threshold is defined as,

$$th_r = th_{init} - \Delta th_r$$

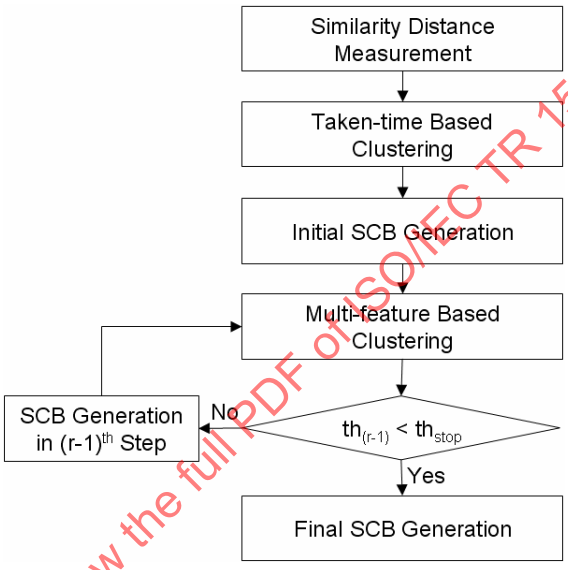
where th_{init} is initial threshold at the first hierarchy, and Δth_r is variation of the threshold in the $(r)^{th}$ hierarchy. th_{init} and Δth_r were set to 0.7 and 0.02, respectively

Situation change detection is finished when the threshold meet the following condition.

$$th_r < th_{stop}$$

where the th_{stop} is minimum criteria to stop the situation change detection. The th_{stop} was set to 0.5.

Figure AMD3.19 shows the situation change detection using hierarchical thresholding. The initial situation change boundaries are determined by using taken-time feature only, since the taken-time is the most useful to group each photo roughly. e.g., time difference more than several hours. Provided that the initial boundaries, the situation change boundary at the next hierarchies is measured by using all multiple features that are available.



*SCB: situation change boundary

Figure AMD3.19 — Hierarchical thresholding procedure

4.8.2.1.5 Condition of Use

The situation boundary detector is well suited to the task of organising a consumer digital photo collection, by breaking it up into small, manageable groups. Time information, which is routinely embedded in image files by digital cameras, is very useful in achieving this goal. While the same kind of grouping could be generated using visual features alone, the method proposed here would require some arbitrary external ordering to determine the sequence of the images in this case.

4.8.2.1.6 DDL instantiation examples

```
<Mpeg7 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mpeg:mpeg7:schema:2004">
   <DescriptionUnit xsi:type="StillRegionType">
      <CreationInformation>
         <Creation>
            <Title />
            <CreationCoordinates>
               <Date>
                  <TimePoint>2006-12-05T10:13+00:00
```

```
                    </TimePoint>
                </Date>
            </CreationCoordinates>
        </Creation>
    </CreationInformation>
    <VisualDescriptionScheme
        xsi:type="StillRegionFeatureType">
        <DominantColor>
            <SpatialCoherency>10</SpatialCoherency>
            <Value>
                <Percentage>6</Percentage>
                <Index>12 3 8</Index>
                <ColorVariance>0 0 0</ColorVariance>
            </Value>
            <Value>
                <Percentage>4</Percentage>
                <Index>28 23 20</Index>
                <ColorVariance>0 0 1</ColorVariance>
            </Value>
            <Value>
                <Percentage>7</Percentage>
                <Index>25 11 11</Index>
                <ColorVariance>0 0 0</ColorVariance>
            </Value>
            <Value>
                <Percentage>4</Percentage>
                <Index>25 16 15</Index>
                <ColorVariance>0 0 0</ColorVariance>
            </Value>
            <Value>
                <Percentage>4</Percentage>
                <Index>19 7 10</Index>
                <ColorVariance>0 1 0</ColorVariance>
            </Value>
            <Value>
                <Percentage>3</Percentage>
                <Index>28 16 14</Index>
                <ColorVariance>0 0 0</ColorVariance>
            </Value>
            <Value>
                <Percentage>1</Percentage>
                <Index>17 11 17</Index>
                <ColorVariance>1 1 1</ColorVariance>
            </Value>
        </DominantColor>
        <ScalableColor numOfCoeff="32"
            numOfBitplanesDiscarded="3">
            <Coeff>
                472 -309 -326 100 41 44 49 54 98 -85 -77 55
                10 20 50 25 67 30 26 18 63 13 29 17 -63 15
                -36 16 -33 3 14 7
            </Coeff>
        </ScalableColor>
        <ColorStructure colorQuant="3">
            <Values>
                0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
                19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
                35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
                51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
                67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82
                83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
```

```
                99 100 101 102 103 104 105 106 107 108 109 110
                111 112 113 114 115 116 117 118 119 120 121 122
                123 124 125 126 127
            </Values>
        </ColorStructure>
        <ColorLayout>
            <YDCCoeff>25</YDCCoeff>
            <CbDCCoeff>22</CbDCCoeff>
            <CrDCCoeff>46</CrDCCoeff>
            <YACCoeff5>16 8 11 15 15</YACCoeff5>
            <CbACCoeff2>21 29</CbACCoeff2>
            <CrACCoeff2>12 10</CrACCoeff2>
        </ColorLayout>
        <HomogeneousPattern>
            <Average>19</Average>
            <StandardDeviation>20</StandardDeviation>
            <Energy>
                103 87 99 130 97 73 112 109 122 132 108 102 105
                113 106 141 103 111 78 76 82 117 88 70 69 61 48
                68 48 53
            </Energy>
            <EnergyDeviation>
                106 84 94 130 94 75 107 104 117 128 100 99 97
                107 92 132 90 106 76 64 78 110 83 65 64 52 39
                72 35 47
            </EnergyDeviation>
        </HomogeneousPattern>
        <TextureBrowsing>
            <Regularity>irregular</Regularity>
            <Direction>90Degree</Direction>
            <Scale>medium</Scale>
            <Direction>0Degree</Direction>
            <Scale>fine</Scale>
        </TextureBrowsing>
    </VisualDescriptionScheme>
  </DescriptionUnit>
</Mpeg7>
```


4.8.2.2 Category based clustering

4.8.2.2.1 Use scenario

The users often feel that it is nuisance and hard to browse their photos in some meaningful orders when they arrange their photos to the digital photo album. This manual cataloguing is quite time-consuming, tedious, erroneous, and inconsistent, so that it has been a big hurdle for users to use digital cameras. Thus the categorizing lot of photos in some automatic manner is strongly needed, in which general users would get some easy ways to browse groups of photos that are semantically linked together like landscape, architecture, night-scene, indoor, people, etc. The category classification also improves capabilities for effectively searching and filtering the desired photos by reducing search ranges.

4.8.2.2.2 Tools to be used

Five tools defined in ISO/IEC 15938-3 shall be instantiated in StillRegionFeatureDS:

- Scalable Color (SC)
- Color Layout (CL)

- Color Structure (CS)
- Homogeneous Texture (HT)
- Edge Histogram (EH)

Segment related tools defined in ISO/IEC 15938-5 shall be instantiated in StillRegionDS to create several sub-regions.

- SpatialLocator
- SpatialDecomposition

4.8.2.2.3 Overall procedure

The overall procedure of the proposed photo categorization is shown in Figure AMD3.20.

For the training, we employed the local and global concept learning mechanism using local-concept lexicon. The training model is generated and stored in the category model DB.

For the image classification, the input image region is divided by 10 sub-regions in terms of the photographic region template set, which consists of 1 center, 4 edges, 2 horizontal, 2 vertical, and a whole image regions. Multiple low-level features of the local region are used in learning and detecting local concepts. Once the local concept detectors have been built, confidence values for each sub-region are measured for all the local concepts. In order to find out the most confident local concepts on the overlapped regions, a local concept merging is carried out. This aims to reduce classification error due to image localization with a fixed block size. A global concept can be a composite of one or more local concepts, meaning that it represents higher-level of semantics than the local concepts. In this paper, the global concept detectors are trained with the confidence vectors of the local concepts. The confidence vectors represent how a region is related with the local concepts. With the confidence vectors, the global concept detectors measure how much the region is related with the global concepts.

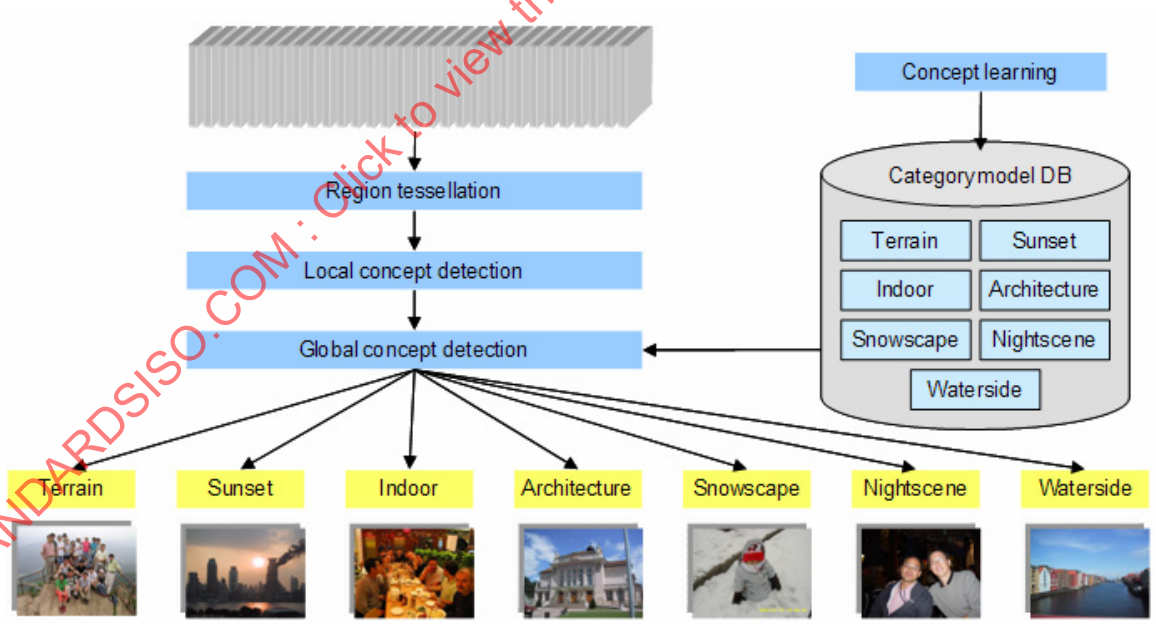


Figure AMD3.20 — Overall procedure of the photo categorization

4.8.2.2.4 Training of concepts

For local concept learning, local concept lexicon is defined first. There are 34 local concept lexicon defined as in Figure AMD3.21. Let us denote the local concept lexicon as V_l, where l = 1, 2, 3,…, L and L is the number of local concepts. The local concept should have relatively lower level of semantics than global (category) concept.

The local semantic concepts are trained with low-level visual feature vectors extracted from the local regions of image. A SVM is employed as a concept detector. The SVM is a binary classifier used to find the decision function of optimal linear hyperplane given a labelled data that is linearly separable in the feature space **H**. In the SVM, the input feature in the space **X** is mapped to a feature space **H** via a nonlinear mapping $\phi(\cdot): \mathbf{X} \to \mathbf{H}$ that allows ones to perform nonlinear analysis of the input data using a linear method. In general SVM, a kernel is designed to map the input data space **X** to the feature space **H**. With the 'kernel trick' property, the kernel can be considered as measures of similarity between two feature vectors without explicit computation of the map $\phi(\cdot)$. The kernel (***K***) is considered as a simple dot product similarity measure between two feature vectors as follows:

$$K(\mathbf{X}_a, \mathbf{X}_b) = \langle \phi(\mathbf{X}_a), \phi(\mathbf{X}_b) \rangle = \phi(\mathbf{X}_a) \cdot \phi(\mathbf{X}_b) = \frac{\sum_{x_b \in \mathbf{X}_b} \sum_{x_a \in \mathbf{X}_a} (x_a \cdot x_b)}{|\mathbf{X}_a||\mathbf{X}_b|}$$

where $\phi(\cdot)$ maps nonlinear input feature vector **X** into linear feature space **H**.

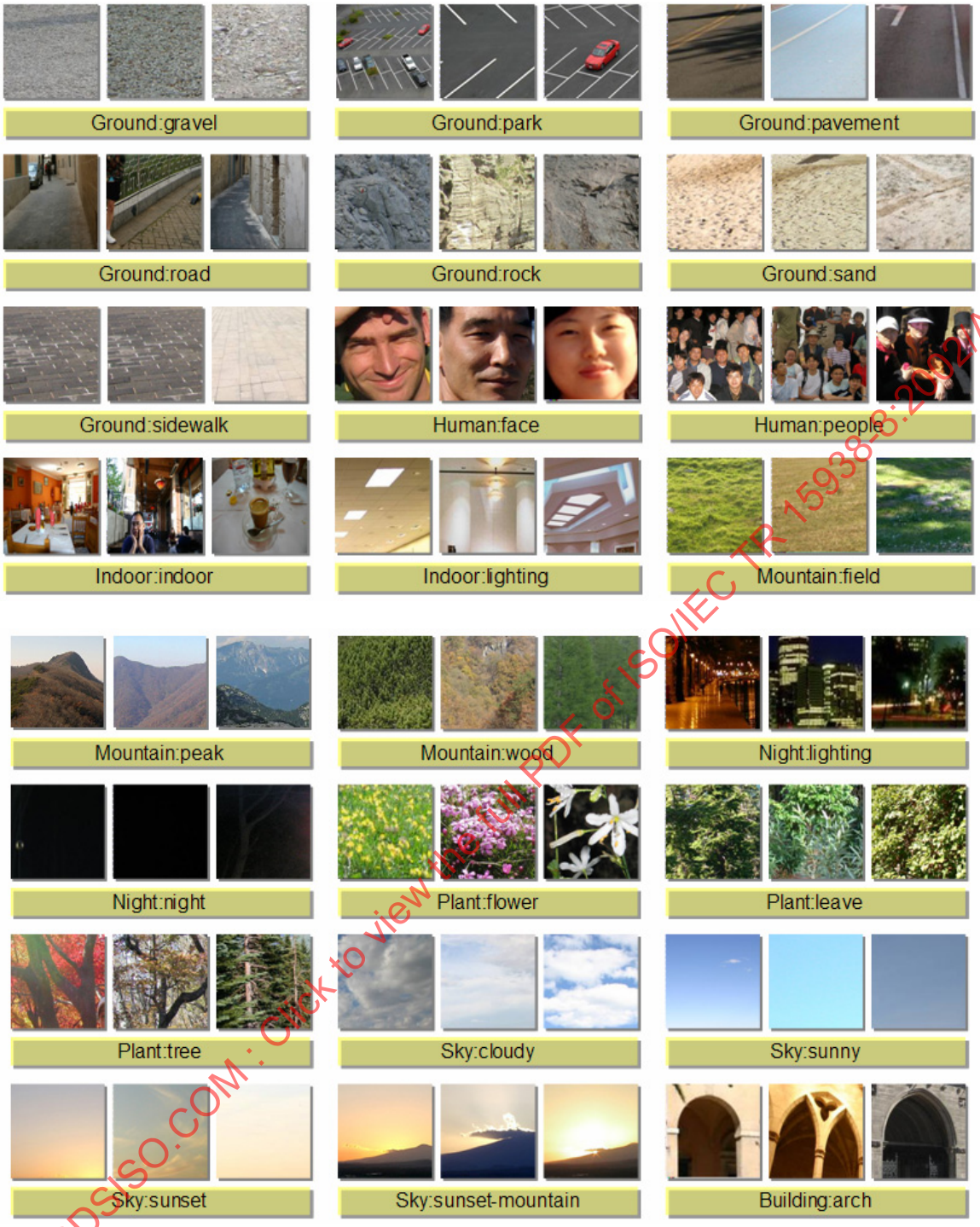
Given the kernel, the SVM for each concept is trained with low-level features **X** of training data. To solve the SVM, an optimal hyperplane is found to correctly classify the training data, i.e., it satisfies $\phi^T(\mathbf{X}_i) \cdot \mathbf{w} + b > 0$ for every training sample $\mathbf{X}_i$ with positive class label y_i = 1 and $\phi^T(\mathbf{X}_i) \cdot \mathbf{w} + b < 0$ for every training sample $\mathbf{X}_i$ with negative class label y_i = -1. The optimization problem to find the optimal hyperplane is a quadratic problem that can be solved by converting it into the Wolfe dual problem. The optimization problem can be written as,

$$\min_{\mathbf{w},b} \frac{\|\mathbf{w}\|^2}{2} \text{ subject to } y_i \{\phi^T(\mathbf{X}_i) \cdot \mathbf{w} + b\} \geq 1$$

By solving the optimization problem, the optimal hyperplane f_l to predict the l concept of unseen data **X** is formed as follows:

$$f_l(\mathbf{X}) = \sum_{i=1}^{k} a_i y_i K(\mathbf{X}_i, \mathbf{X}) + b$$

where a_i is support vector that is always positive and k is the number of support vectors.



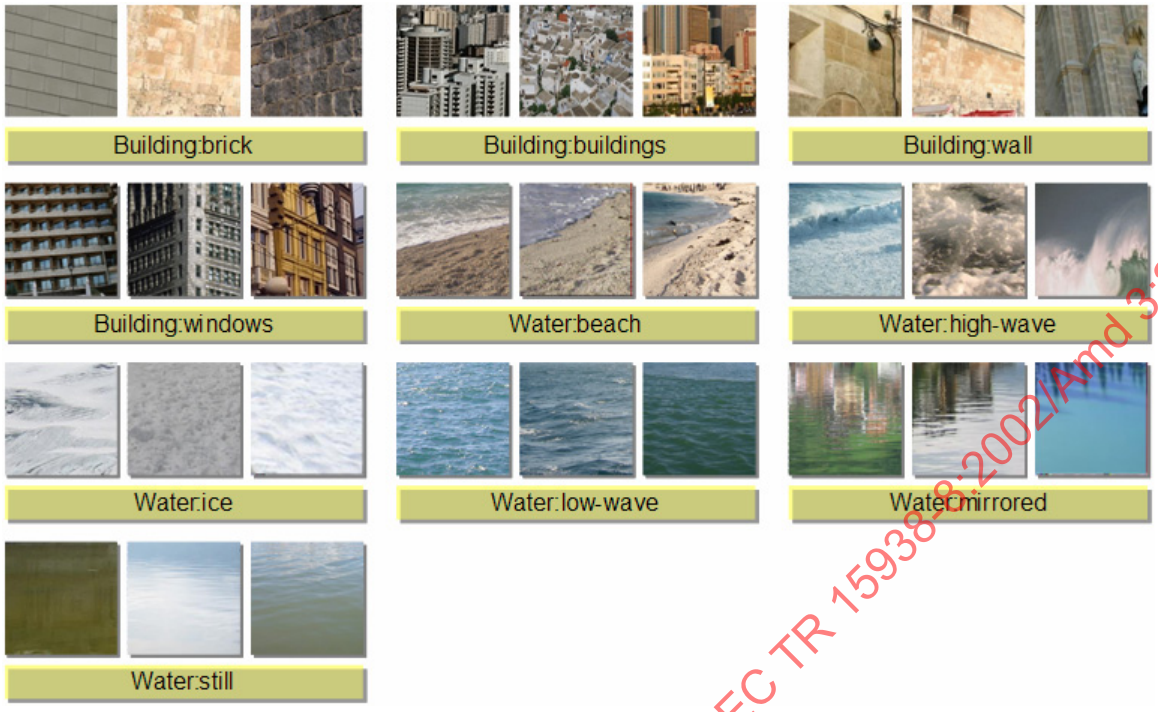


Figure AMD3.21 — Training images for 34 local concepts

Figure AMD3.22 demonstrates how the local concepts are trained with SVM. First, the MPEG-7 visual descriptors are extracted from the training images of each local concept. Each descriptor set is fed to the SVM and generates support vectors. Finally, the support vectors formed by each descriptor are gathered to form a set of support vectors to represent the local concept.

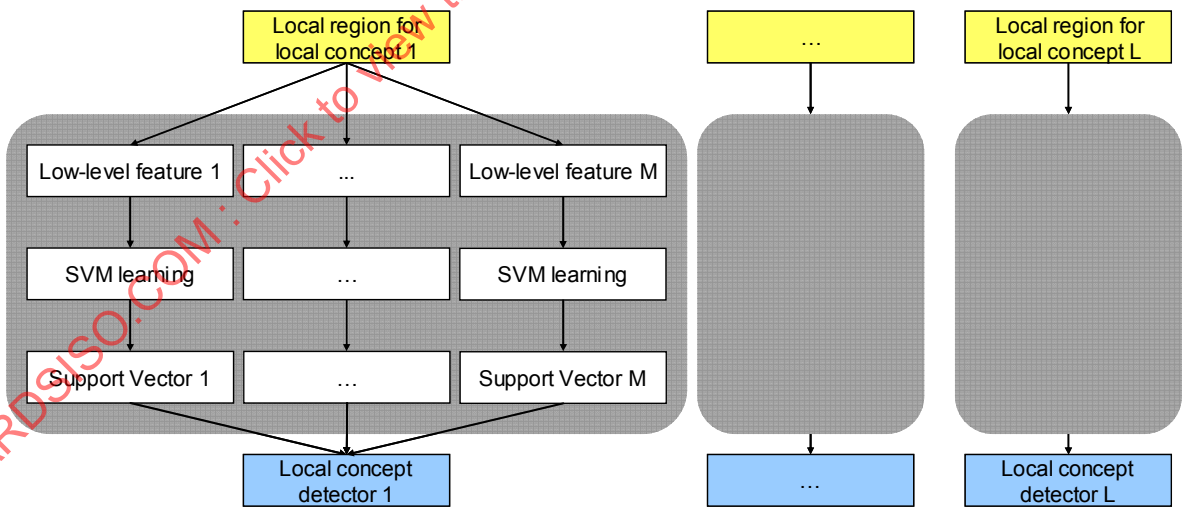


Figure AMD3.22 — Local concept detectors generation process

Similar to local concept learning, let us denote the global concept lexicon as V_n, where n = 1, 2, 3,…, N and N is the number of global concepts. The kernel for the global SVM is the same as that for the local SVM. By solving the optimization problem similar to local concept learning, the optimal hyperplane f_n to predict the global concept V_n of unseen data **Y** is formed as follows:

$$f_n(\mathbf{Y}) = \sum_{i=1}^{k} a_i y_i K(\mathbf{Y}_i, \mathbf{Y}) + b$$

As a whole process, the ground truth regions are selected from the training images for each global concept (Figure AMD3.23). Then, the 5 descriptors are extracted from each ground truth region. Each set of descriptors are fed into the local concept detectors and generate the confidence vectors, which again, are fed to the SVM and generate support vector as a global concept detector.

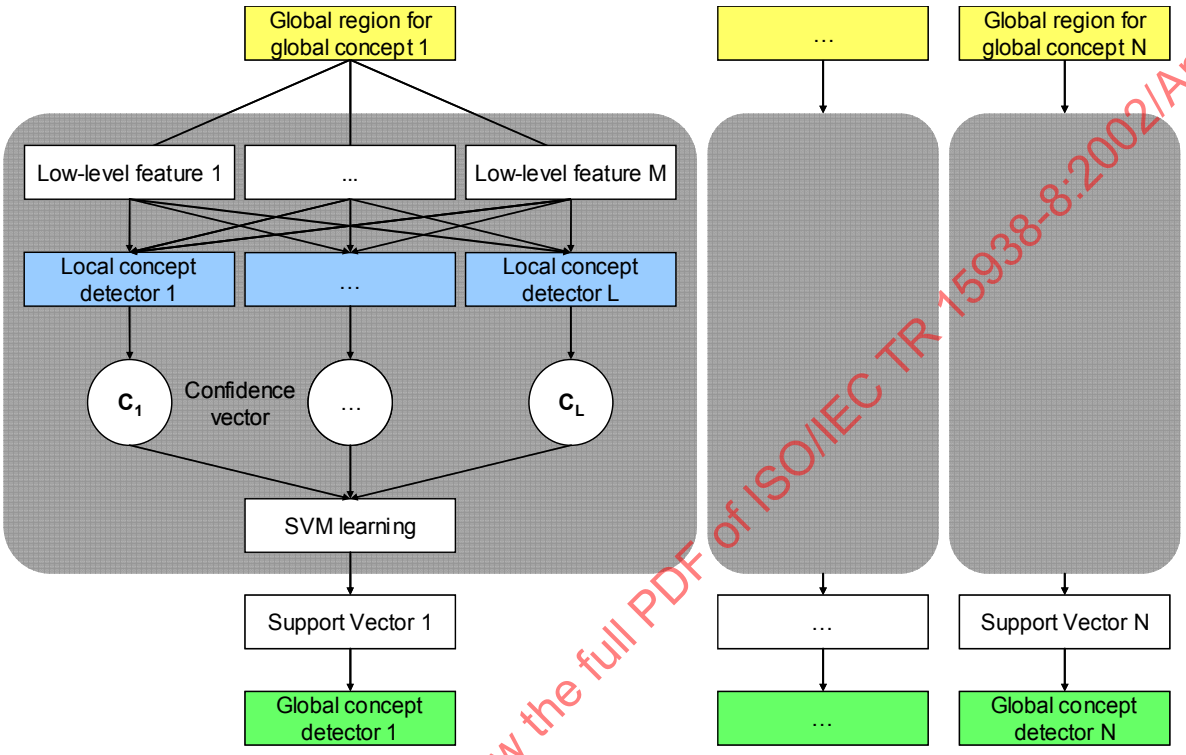


Figure AMD3.23 — Global concept detectors generation process

4.8.2.2.5 Automatic category classification

4.8.2.2.5.1 Region tessellation

In order to detect local semantics of image, an entire image region is divided into several sub-regions using photographic region templates suitable for home photographs. There may be some objects in foreground and scenery in background. Without any help of object detection, the location and contour of foreground objects are unpredictable. Thus, we assume that foreground objects could be located in any sub-region of image. In other words, semantic objects or scenes in home photo could be located in center, edge, horizontal, vertical or a whole region.

To find meaningful region templates, three important requirements are considered; one is that the region template should be large enough to detect local image semantic, another is that the region template should be small enough not to be time-consuming in practice, and the other is that the template should be scalable to support multi-resolution.

Figure AMD3.24 shows the proposed photographic region template. The region template set is composed of 10 sub-regions; they are 1 center region (T_1 in Figure AMD3.24), 4 edge regions (T_2, T_3, T_4, and T_5 in Figure AMD3.24), 2 horizontal regions (T_6 and T_7 in Figure AMD3.24), 2 vertical regions (T_8 and T_9 in Figure AMD3.24), and a whole region (T_{10} in Figure AMD3.24). The 4 edge regions are totally parts of the