MERGING MPEG-7 DESCRIPTORS FOR IMAGE CONTENT ANALYSIS

Patrick Ndjiki-Nya, Oleg Novychny and Thomas Wiegand

Fraunhofer Institute for Telecommunications – Heinrich-Hertz-Institut
Image Processing Department
Einsteinufer 37, 10587 Berlin, Germany
{ndjiki/novychny/wiegand}@hhi.de

ABSTRACT

A segmentation algorithm for image content analysis is presented. We assume that the textures in a video scene can be labeled subjectively relevant or irrelevant. Relevant textures are defined as containing subjectively meaningful details, while irrelevant textures can be seen as image content with less important subjective details. We apply this idea to video coding using a texture analyzer and a texture synthesizer. The texture analyzer (encoder side) identifies the texture regions with unimportant subjective details and generates side information for the texture synthesizer (decoder side), which in turn inserts synthetic textures at the specified locations. The focus of this paper is the texture analyzer, which uses multiple MPEG-7 descriptors simultaneously for similarity estimation. The texture analyzer is based on a split and merge segmentation approach. Its current implementation yields an identification rate of up to 96% and an average gain of up to 10% compared to single descriptor usage.

1. INTRODUCTION

Textures like water, grass, trees, sand, etc. that are present in many video sequences are difficult to code due to the large amount of visible detail. We claim that the exact regeneration of such textures is not necessary if they are shown with limited spatial resolution and the original video is not known to the viewer . The viewer should just be able to identify the semantic category of the reconstructed textures, which is often not the case when a pre-filter is used or these are blurred due to strong quantization. We exploit this idea for video coding using a texture analyzer at the encoder side and a texture synthesizer at the decoder side.

The identification of detail-irrelevant texture regions (water, sand …), the creation of coarse masks corresponding to these regions, as well as the signaling of these masks as side information to the decoder are the main tasks of the texture analyzer. The texture synthesizer replaces the marked textures via inserting synthetic textures.

In [1] it is shown that detail-irrelevant textures can be represented using MPEG-7 descriptors [2],[3], instead of the mean squared error, as the coding distortion. Since the considered MPEG-7 descriptors evaluate overall similarity, the reproduced textures typically show different details than the original ones. These deviations between original and synthetic textures are not subjectively noticeable as long as the displayed spatial accuracy of the textures remains unchanged and are also much less annoying as if they were coded at a bit-rate which is equivalent to the bit-rate of the side information of the texture synthesizer. In [1], we show that substantial bit-rate savings can be achieved using our coding approach. The gains thereby increase with increasing video quality. E.g., bit-rate savings of up to 19.4% compared to an H.264/AVC video codec were measured for the Flowergarden test sequence (CIF resolution, 30 Hz progressive video and quantization parameter 16).

In this paper, we focus on the texture analyzer. Its segmentation strategy, the selected MPEG-7 descriptors for similarity estimation as well as the corresponding metrics and weighting strategies are developed.

A similar wavelet-based analysis-synthesis video coding approach was introduced by Yoon and Adelson [4] and by Dumitraş and Haskell [5]. The algorithms presented in [4],[5] are optimized for textures with absent or very slow global motion, whereas no such constraint is required for our system [1].

The remainder of the paper is organized as follows. In Section 2, we present the segmentation strategy of the texture analyzer. Finally, in Section 3 the experimental results are shown.

2. SEGMENTATION STRATEGY

The texture analyzer performs a split and merge segmentation of each frame of a given video sequence. This corresponds to a region-based segmentation for coarse detection of true regions [6].

2.1. Splitting step

The splitting step consists in analyzing a frame using a multi-resolution quadtree [7]. The latter encompasses several levels, the first level (level 0) being the original frame itself. At level 1, the original frame is split into 4 non-overlapping blocks, while it is split into 16 non-overlapping blocks at level 2, etc. The amount of blocks at level L is given by 4^L. Each block at level L-1 is split into four blocks at level L, since the amount of blocks per

column is always identical to the amount of blocks per row (e.g. four blocks per row/column at level 2).

A block at level L-1 is considered to have homogeneous content if its four sub-blocks at level L have "similar" statistical properties. An optional auxiliary requirement for homogeneity is that the $(2n+1)\times(2n+1)$ non-overlapping sub-blocks ($n \in \mathrm{N}$) of the considered block (level L-1) are pairwise similar. This additional condition allows better detection of details at block boundaries as well as in the middle of blocks. The similarity between two blocks is measured in terms of corresponding MPEG-7 descriptors as explained below. Inhomogeneous blocks are split further, while homogeneous blocks remain unchanged. The splitting stops, when the smallest allowed block size is reached, and the non-homogeneous areas of the considered frame are marked as being not classified. The smallest allowed block size can be set according to a priori knowledge concerning the size of the structures in the given video sequence.

The segmentation mask obtained after the splitting step typically shows a clearly over-segmented frame. Thus post-processing of the former is required, which leads to the second step implemented by the texture analyzer - the merging step.

2.2. Merging step

In the merging step, homogeneous blocks identified in the splitting step are compared pairwise and similar blocks are merged into a single cluster forming an homogeneous area itself. The merging stops if the obtained clusters are stable, i.e. if they are pairwise dissimilar. Typically the final number of clusters is considerably reduced by the merging step.

In addition to merging homogeneous texture regions, the corresponding MPEG-7 feature vectors can also be updated, which is an optional feature of the texture analyzer. Thus the merging of similar homogeneous texture regions can be taken into account in the feature space. If this feature is switched off, no feature vector update is done in case of the merging of two homogeneous texture regions.

2.3. Similarity estimation

The similarity assessment between two blocks is done based on MPEG-7 descriptors [2],[3]. We use the "Edge Histogram" (EH) texture and the "SCalable Color" (SCC) descriptors. These features have initially been developed for visual content representation with the main target being image retrieval.

2.3.1. Edge Histogram descriptor

The EH descriptor represents the spatial distribution of four directional edges (one horizontal, one vertical, and two diagonal edges) and one non-directional edge for 16 local, non-overlapping regions of a given image. The frequency of occurrence of each edge class is determined for each local region, which yields an 80 (16x5) dimensional feature vector.

We also use a global EH descriptor that can easily be derived from the MPEG-7 standard conforming EH descriptor described above. The global EH is of dimension five and represents an edge-class-wise addition of the 16 local histograms.

2.3.2. SCalable Color descriptor

The SCC descriptor is basically a color histogram in the HSV color space. HSV is a three-dimensional color space with the components Hue, Saturation and Value (luminance). The resolution (number of colors or bins) of the SCC descriptor can be varied from 16 to 256 colors. The number of possible colors is thereby doubled from resolution step to resolution step. We use the highest resolution step in order to achieve best possible segmentation results given the SCC descriptor.

The MPEG-7 standard conforming SCC histogram described above consists of 16 hues. Each hue has four corresponding saturation levels per given luminance value, which yields 64 bins per luminance value. Four luminance values are used in the reference SCC histogram, which leads to a total of 256 bins. If H_{vs}^{h} represents a color with quantized hue h (h = 0...15) at quantized saturation s (s = 0...3) and quantized value v (v = 0...3), then the colors in the reference SCC histogram are sorted in the following order:

$$H_{00}^{h}\cdots H_{03}^{h}\, H_{10}^{h}\cdots H_{13}^{h}\, H_{20}^{h}\cdots H_{23}^{h}\, H_{30}^{h}\cdots H_{33}^{h}.$$

The reference SCC descriptor was modified to achieve better segmentation results for images with varying saturations or luminance values of the same hue. The modifications consist in re-ordering the bins of the MPEG - 7 standard conforming SCC histogram, i.e. the dimension of the SCC histogram is not altered. The colors in the re-ordered SCC histogram are sorted in the following manner:

$$H_{00}^{0}\cdots H_{03}^{0}H_{13}^{0}\cdots H_{10}^{0}\, H_{20}^{0}\cdots H_{23}^{0}\, H_{33}^{0}\cdots H_{30}^{0}\cdots H_{30}^{1}\quad\cdots\quad H_{30}^{15}\cdots H_{00}^{15}.$$

As can be seen above, the re-ordering yields storing all variations of a given hue h in neighboring bins.

The re-ordering has a positive impact on the segmentation results for textures with varying saturations or/and luminance values of the same hue and given an adequate metric. The same applies to the reference SCC histogram for textures with varying hues and constant luminance and saturation.

2.3.3. Merging Edge Histogram and SCalable Color

The detection of detail-irrelevant textures can be optimized by using both, the SCC and EH descriptors, for similarity assessment. The relevance of the decisions made by the above features depends on the content of the blocks to be examined. We use two weighting strategies to determine the relevance of each feature for content analysis of the four sub-blocks of a given block (cp. Section 2.1.).

The first weighting approach is based on the dynamic weighting mechanism described in [8]. It assumes that the feature with the highest variance in the feature space is more likely to lead to the best homogeneity decision. The feature weights are determined as follows:

$$w = \frac{\sigma}{\bar{d}}\,\mathrm{I};\ \bar{d} = \frac{1}{6}\sum_{i=0}^{2}\sum_{j=i+1}^{3} d_{ij}\ ;\ \sigma = \frac{1}{6}\sum_{i=0}^{2}\sum_{j=i+1}^{3}\left| d_{ij} - \bar{d}\right| \quad (1)$$

where w represents the weight of the considered feature. $\bar{d}$ is the mean distance between the feature vectors of sub-blocks i and sub-blocks j. The sub-blocks are numbered increasingly from left to right and from top to bottom starting with zero. $\bar{\sigma}$ represents the mean deviation from the mean distance $\bar{d}$. The mean relevance index $\bar{I}$ of the considered features is determined for the four given sub-blocks. We use saturation for SCC and the detected edge type for EH. I.e. EH is considered to be relevant (I=1) if more than only non-directional edges are seen (I=0 else), as non-directional edges are non-specific. The determined weights are normalized by the total sum of weights.

The second weighting approach is not variance-based and uses the normalized relevance indexes as weights.

2.3.4. Thresholds and metrics

Two blocks are considered to be similar if the distance between the corresponding feature vectors lies below a given threshold. In case of combined use of SCC and EH, the overall distance between two blocks is given by the weighted sum of the single distances:

$$w_{SCC} d_{SCC} + w_{EH} d_{EH} \le w_{SCC} T_{SCC} + w_{EH} T_{EH} \qquad (2)$$

where $d_{SCC/EH}$ represent the distances between the feature vectors of the considered blocks, while $T_{SCC/EH}$ and $w_{SCC/EH}$ are the corresponding similarity thresholds and weights respectively. The thresholds are determined as a proportion of the maximum possible distance between two feature vectors. The maximum distance depends both on the selected metric and the chosen descriptor. A threshold of zero means that two feature vectors are seen as similar if and only if they are identical, while a threshold of one indicates that any two feature vectors will be seen as similar, as no distance can be greater than the maximum one.

Two metrics are used to determine the distance between feature vectors: the ℓ_1 norm (EH, SCC) and the Earth Mover's Distance [9] (SCC only). If we define the bin population of the first of two histograms as hills and the corresponding population of the second histogram as valleys, then EMD represents the minimum "earth" transportation cost from the hills to the valleys. The greater the distance between provider (histogram #1) and receiver bin (histogram #2), the higher the transportation costs. Histograms with different locations of most of the "earth" concentration will be labeled as very different, while histograms with similar shapes and noisy deviations will be seen as similar. EMD is robust against noise, scaling and shift because it mainly compares the shapes of the histograms. This makes EMD eligible for compensating lighting variations, when used in combination with the SCC descriptor.

3. EXPERIMENTAL RESULTS

In our experiment, we evaluate the quality of the segmentation results obtained using the texture analyzer in combination with SCC and EH. A test set of 51 images is used. 15 of the images are gray-level images and are used only for evaluation of EH. For each image, a reference mask is manually generated by first splitting the former into 16x16 non-overlapping macroblocks. The macroblock grid is thereby imposed by the coding scenario [10]. Macroblocks containing a homogeneous texture are marked "classifiable", while the others are labeled "unclassifiable". The "classifiable" blocks are then manually clustered and those containing the same homogeneous texture are assigned to the same class. The reference masks are then compared to the best masks generated by the texture analyzer. The best mask, i.e. the mask with the least segmentation errors, for a given texture analyzer configuration is obtained by varying the similarity thresholds (cp. Section 2.3.4.) using a fixed step width, thereby minimizing segmentation errors.

3.1. SCalable Color descriptor

The evaluation of the SCalable Color descriptor is done using a total of 36 images selected in consideration of the lighting conditions, the presence/absence of details in the images (useful for evaluation of detail identification potentialities of the texture analyzer) and a "good" coverage of the HSV color space. The most important configurations of the texture analyzer in combination with SCC are shown in Tab. 1. SCC_RO represents the re-ordered version of the reference SCC histogram (cp. Section 2.3.2.).

	Descriptor	Metric	Update after merging	Detail Identi-fication
Config. #1	SCC	EMD	No	Yes
Config. #2	SCC	EMD	No	No
Config. #3	SCC	EMD	Yes	No
Config. #4	SCC_RO	EMD	No	No
Config. #5	SCC_RO	EMD	Yes	No
Config. #6	SCC_RO	l_1	No	Yes
Config. #7	SCC_RO	l_1	No	No

Tab. 1: Evaluated configurations of the texture analyzer in combination with the „SCalable Color" descriptor

Figure 1 depicts the correctly identified image area for the evaluated configurations of the texture analyzer. The strongly overlapping notches of the box plots indicate that none of the evaluated configurations is statistically significantly better than the others. Configurations #2 and #3 yield the best average texture identification rate of 71% (median value, horizontal line within the corresponding boxes) given the test data. The configurations with EMD (configs. #1 to #5) as the metric lead to the best results as expected (cp. Sections 2.3.2. and 2.3.4.).

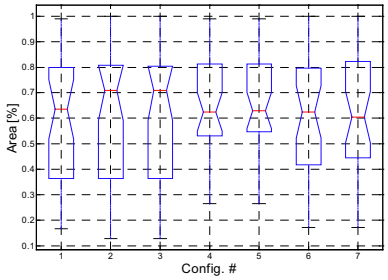

Figure 1: Correctly identified image area for seven configurations of the texture analyzer in combination with the „SCalable Color" descriptor

Less than 25% of the test images lead to a correctly identified image area smaller than 37% (lower quartile corresponds to lower bound of the box), whereas more than 80% of the image area is correctly segmented for more than 25% of the test images (upper quartile

corresponds to upper bound of the box) for configurations #2 and #3. The whiskers drawn from the lower (upper) quartile to the smallest (biggest) correctly identified area cover the range of the data.

Within the false segmentation class, we distinguish between false negatives and false positives. False negatives are image areas that are marked "classifiable" in the reference mask and labeled "unclassifiable" in the best texture analyzer mask, while false positives correspond to the other possible mismatches. Considering only the false positive macroblocks as erroneous leads to an average non-erroneous identified area of 99.64%.

3.2. Edge Histogram descriptor

The evaluation of the Edge Histogram descriptor is done using 21 images selected in consideration of the texture resolution, orientation and coarseness. The most important configurations of the texture analyzer in combination with EH are shown in Tab. 2. EH_GL represents the global version of the reference EH descriptor (cp. Section 2.3.1.). Note that the ℓ_1 norm is used for all configurations as recommended in the MPEG-7 standard [2],[3].

	Descriptor	Metric	Update after merging	Detail Identi-fication
Config. #1	EH	l_1	No	Yes
Config. #2	EH	l_1	No	No
Config. #3	EH	l_1	Yes	Yes
Config. #4	EH	l_1	Yes	No
Config. #5	EH_GL	l_1	No	Yes
Config. #6	EH_GL	l_1	No	No

Tab. 2: Evaluated configurations of the texture analyzer in combination with the „Edge Histogram" descriptor

Figure 2 depicts the correctly identified image area for the evaluated configurations of the texture analyzer. Configuration #6 leads to significantly better results than the others.

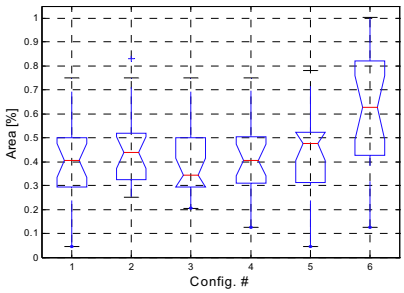

Figure 2: Correctly identified image area for six configurations of the texture analyzer in combination with the „Edge Histogram" descriptor

An average identification rate of 62.5% can be measured here. Considering only the false positive macroblocks as erroneous leads to an average non-erroneous identified area of 96.5%.

3.3. Merging of SCalable Color and Edge Histogram

The merging of the SCalable Color and Edge Histogram descriptors is evaluated using a total of 36 images (cp. Section 3.1.). Configurations #2 and #6 are used for SCC and EH respectively (cp. Sections 3.1. and 3.2.).

Figure 3 shows the correctly identified image area for both weighting approaches (cp. Section 2.3.3.) and the best texture analyzer configuration in combination with SCC. It can be seen that both weighting strategies yield better results than configuration #2 of SCC. However, the weighting mechanism that is not based on variance leads to the best results. An average identification rate of 81% can be measured here, which corresponds to a gain of 10% compared to configuration #2 of SCC.

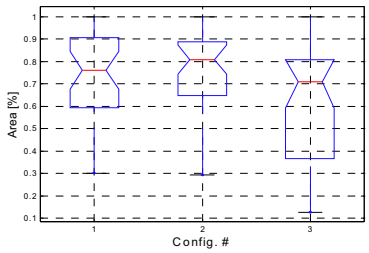

Figure 3: Correctly identified image area for the variance-based (left) and the not variance-based (middle) weighting strategies as well as for configuration #2 of the „SCalable Color" descriptor (right)

Considering only the false positive macroblocks as erroneous leads to an average non-erroneous identified area of 96%.

4. CONCLUSIONS

We have presented a segmentation algorithm for image content analysis. Two MPEG-7 descriptors, a texture (Edge Histogram) and a color descriptor (SCalable Color), are merged for similarity estimation. Our experiments show that the average texture identification rate increases by up to 10% for the best weighting strategy compared to the best single descriptor. Further weighting strategies will be explored in order to improve the homogeneous texture identification performance.

5. REFERENCES

[1] P. Ndjiki-Nya, et al., "Improved H.264 Coding Using Tex-ture Analysis and Synthesis", *Proc. ICIP 2003*, Barcelona, Spain, September 2003.

[2] ISO/IEC JTC1/SC29/WG11/N4362, "MPEG-7 Visual Part of eXperimentation Model Version 11.0", Sydney, Australia, July 2001.

[3] ISO/IEC JTC1/SC29/WG11/N4358, "Text of ISO/IEC 15938-3/FDIS Information technology – Multimedia content description interface – Part 3 Visual", Sydney, Australia, July 2001.

[4] S.-Y. Yoon and E. H. Adelson, "Subband texture synthesis for image coding", *Proc. SPIE on HVEI III*, Vol. 3299, pp. 489-497, San Jose, USA, January 1998.

[5] A. Dumitraş and B. G. Haskell, "An Encoder-Decoder Texture Replacement Method with Application to Content-based Movie Coding", *To be published in IEEE Trans. on CSVT.*

[6] J. Freixenet, et al., "Yet Another Survey on Image Segmentation : Region and Boundary Information Integration", *Proc. ECCV*, Part III, Vol. 2352, pp. 408-22, Copenhagen, Denmark, May 2002.

[7] J. Malki, et al., "Region Queries without Segmentation for Image Retrieval by Content", *VISUAL'99*, pp.115-22, 1999.

[8] Y. Luo, et al., "Extracting meaningful regions for content-based retrieval of image and video", *VCIP 2001*, Vol. 4310, pp. 455-464, San Jose, USA, January 2001.

[9] Y. Rubner, et al., "A Metric for Distributions with Applications to Image Databases", *ICCV'98*, pp.207-214, 1998.

[10] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced Video Coding for Generic Audiovisual Services", 2003.