VIDEO CONTENT ANALYSIS USING MPEG-7 DESCRIPTORS

Patrick Ndjiki-Nya, Oleg Novychny and Thomas Wiegand

Fraunhofer Institute for Telecommunications – Heinrich-Hertz-Institut
Germany

ABSTRACT

A video content analysis tool for video coding is presented. The underlying assumption of our approach is 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 contain 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 inserts synthetic textures at the specified locations. The focus of this paper is the texture analyzer, which uses MPEG-7 descriptors for similarity estimation. The texture analyzer is based on a split and merge segmentation approach and also provides solutions concerning temporal mapping of identified detail-irrelevant textures. The current implementation of the texture analyzer yields an identification rate of up to 93%.

1. INTRODUCTION

Textures like water, grass, trees, sand, etc. 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 mandatory if they are shown with limited spatial resolution and the original video is not known to the viewer . He 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 Ndjiki-Nya et al. (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 as 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. The segmentation strategy as well as the MPEG-7 similarity criteria, including the selected descriptors and metrics, are elaborated. A technique for ensuring temporal consistency of the identified texture regions is also presented.

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).

Analysis-synthesis-based codecs have also been introduced for object-based video coding applications (e.g. cp. Wollborn (6)). The purpose of the analyzer and synthesizer modules in this case is usually the identification and appropriate synthesis of moving objects (6). Such approaches can be seen as complementary to ours as the texture analyzer presented in this paper tends to identify background textures.

The remainder of the paper is organized as follows. In Section 2 we present the segmentation strategy of the texture analyzer, while in Section 3 temporal consistency of the identified detail-irrelevant texture regions is addressed. Finally, in Section 4 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 (cp. Freixenet et al. (7)).

Printed and published by the IEE, Michael Faraday House, Six Hills Way, Stevenage, Herts SG1 2AY, UK

2.1. Splitting step

The splitting step consists in analyzing a frame using a multi-resolution quadtree (cp. Malki et al. (8)). The latter encompasses several levels, with 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 splitted into 4 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, i.e. an odd number of sub-blocks of the current block (level L-1) is considered in x and y directions (e.g. 3x3, 5x5 ...). 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 step features two break conditions:

1. *The smallest admissible block size* is user-specifiable and can be set according to a priori knowledge of the size of the structures in the given video sequence or run-time constraints.
2. *The block status*: The texture analyzer stops splitting a given frame if all corresponding blocks are labeled "homogeneous".

Samples that are still unlabeled after the first break condition is fulfilled are labeled „unclassifiable".

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 a homogeneous area itself. The merging stops if the obtained clusters are stable, i.e. if they are pairwise dissimilar. The final number of clusters is typically 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. I.e. the feature vector of one of the homogeneous texture regions composing a cluster, resulting from the merging of similar homogeneous texture regions, is used as representative of the cluster.

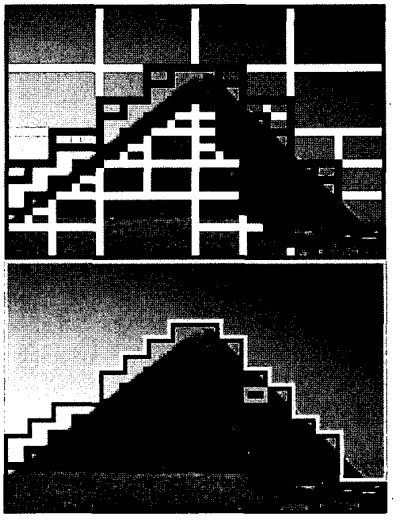

Figure 1: Segmented image after the splitting step (top) and after the merging step (bottom)

Figure 1 illustrates the split/merge steps by showing the segmentation masks of a frame after the splitting (top) and after the merging (bottom) steps. Regions labeled „unclassifiable" are marked by a black border, while classified regions are marked by a non-black border. It can be seen that the number of homogeneous clusters can be substantially reduced after the merging step.

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. The Edge Histogram descriptor is preferred to the Homogeneous Texture descriptor because the reference implementation of the latter works only for images sized at least 128x128 samples. Taking the Texture Browsing descriptor into account is not feasible due to its incompatibility with our task (2),(3). The SCC descriptor is used because of its scalability and as a starting point for first evaluations of the texture analyzer. Note that the MPEG-7 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 (cp. Figure 2) 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.

Figure 2: Edge classes identifiable by the „Edge Histogram" descriptor

We also use a global EH descriptor that can easily be derived from the MPEG-7 standard conforming EH descriptor delineated above. The global EH is of dimension five an 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 assess the best possible segmentation results given the SCC descriptor.
The MPEG-7 standard conforming SCC histogram described above consists of 16 hue values. Each hue value 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 luminances of the same hue. The modifications consist in re-ordering the bins of the standard conform 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.

Figure 3: Bins 241 to 256 of the „SCalable Color" (top) and modified „SCalable Color" (bottom) histograms

Figure 3 depicts the colors represented by the bins 241 to 256 of the SCC and re-ordered SCC histograms. There are obvious differences between the two histograms. While the SCC sub-histogram (top) shows 16 hues with constant saturation and luminance, the modified SCC sub-histogram (bottom) depicts all variations of a given hue. I.e. hue is constant, whereas saturation and luminance vary.
The re-ordering has a positive impact on the segmentation results for textures with varying saturations or/and luminances of the same hue and given an adequate metric, as said above. The same applies to the reference SCC histogram for textures with varying hues and constant luminance and saturation.

2.3.3. Thresholds and metrics. Two blocks are considered to be similar if the distance between the corresponding feature vectors lies below a given threshold:

$$\begin{aligned} d\,(\overrightarrow{SCC_1}, \overrightarrow{SCC_2}) &\le T_{SCC} \\ d\,(\overrightarrow{EH_1}, \overrightarrow{EH_2}) &\le T_{EH} \end{aligned} \qquad (1)$$

where $\overrightarrow{SCC_i}/\overrightarrow{EH_i}$ (i=1,2) represent the feature vectors of the considered blocks, while T_{SCC} and T_{EH} are the similarity thresholds. 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. The thresholds are manually optimized for some key frames of a given sequence. The optimal threshold is then used for all frames of the video. The texture analyzer presented here can therefore be seen as a semi-automatic segmentation algorithm.
Two metrics are used to determine the distance between feature vectors: the ℓ_1 norm (EH, SCC) and the Earth Mover's Distance (EMD) (SCC only, cp. Rubner et al. (9)). 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.

The splitting and merging steps segment each frame of a given sequence independently of the other frames of the same sequence. This yields inconsistent temporal texture identification. Thus a mapping of textures identified in a frame to textures identified in previous frames of the same sequence is required.

3. TEMPORAL CONSISTENCY

Temporal consistency of detected synthesizable textures is ensured by setting up a texture catalog that contains information about the textures present in the given sequence. The texture catalog is initialized with the feature vectors of the textures identified in the first frame of the sequence. In case no texture is identified in the starting frame, the catalog is initialized with the textures of the first frame where at least one texture is found. The textures identified in the following frames are first compared to the indexed texture(s) and mapped to one of them if similar. The former are added to the texture catalog otherwise.

Note that the segmentation masks available at this stage must be adapted to the macroblock grid (e.g. 16x16 samples) for simple integration into the H.264/AVC codec (10).

4. 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 150 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 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 T_{SCC} and T_{EH} (cp. Section 2.3.3.) and using a fixed step width, thereby minimizing segmentation errors.

4.1. SCalable Color descriptor

The evaluation of the SCalable Color descriptor is done using a total of 135 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 Table 1. SCC_RO represents the re-ordered version of the reference SCC histogram (cp. Section 2.3.2.).

TABLE 1: Evaluated configurations of the texture analyzer with the „SCalable Color" descriptor as similarity criterion

	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

Figure 4 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. Configuration #2 yields the best average texture identification rate of 70% (median value, horizontal line within the corresponding boxes) given the test data.

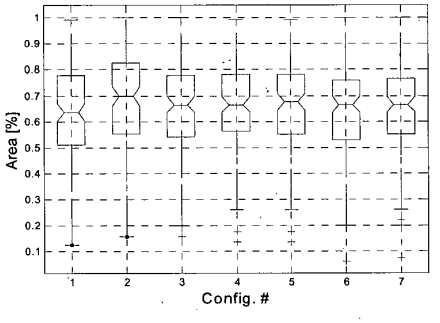

Figure 4: Correctly identified image area for seven configurations of the texture analyzer with the „SCalable Color" descriptor as similarity criterion

The configurations with EMD (configs. #1 to #5) as the metric lead to the best results in most of the cases as expected (cp. Sections 2.3.2. and 2.3.3.). Less than 25% of the test images lead to a correctly identified image area smaller than 55% (lower quartile corresponds to lower bound of the box), whereas more than 78% 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 configuration #2. The whiskers drawn from the lower (upper) quartile to the smallest (biggest) correctly identified area cover the range of the data. Note that statistical outliers are represented by crosses below or above the whiskers.

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 92%. Some segmentation results are shown in Figure 5.

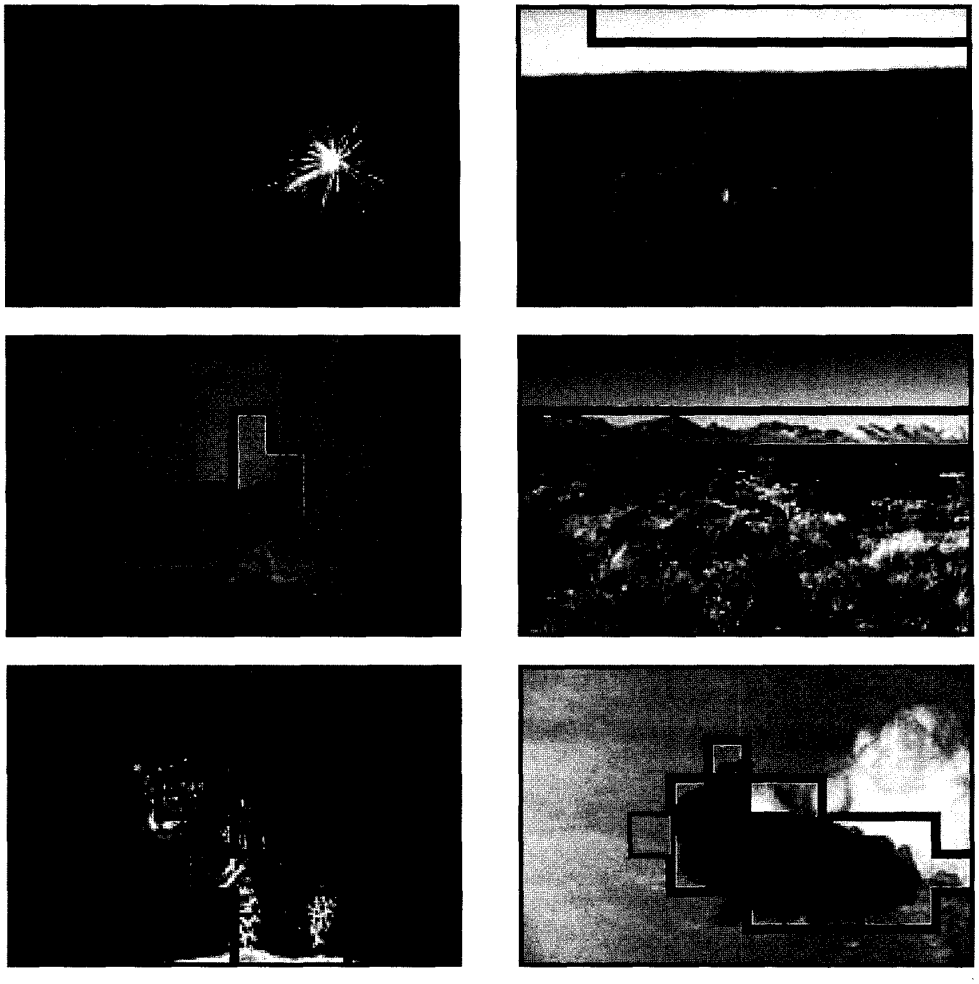

Figure 5: Some segmentation results obtained with configuration #2 of the texture analyzer (cp. Table 1)

4.2. Edge Histogram descriptor

The evaluation of the Edge Histogram descriptor is done using 120 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 Table 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).

TABLE 2: Evaluated configurations of the texture analyzer with the „Edge Histogram" descriptor as similarity criterion

	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

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

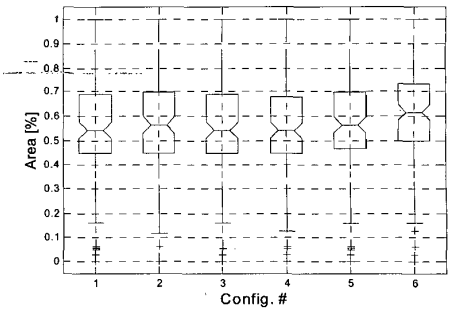

Figure 6: Correctly identified image area for six configurations of the texture analyzer with the „Edge Histogram" descriptor as similarity criterion

An average identification rate of 61% can be measured here. Considering only the false positive macroblocks as erroneous leads to an average non-erroneous identified area of 93%. Some segmentation results are shown in Figure 7.

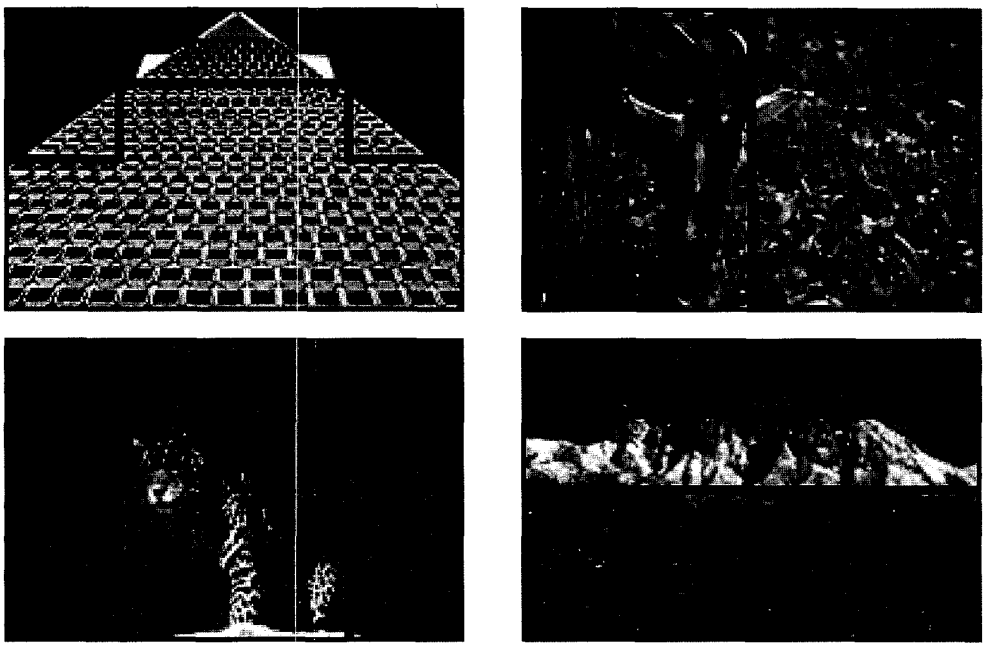

Figure 7: Some segmentation results obtained with configuration #6 of the texture analyzer (cp. Table 2)

4.3. Temporal consistency

In the temporal consistency experiment, we evaluate the performance of the texture catalog (cp. Section 3) responsible for the correct temporal identification of a given texture.

We segment 151 frames for each of the three considered test sequences (Flowergarden, Concrete and Canoe). Configuration #2 of the texture analyzer (Table 1) is used for segmentation. The temporal texture mapping implemented by the texture catalog is consistent for all frames of the test sequences. The area covered by the biggest detail-irrelevant texture region

found is plotted for each of the sequences (cp. Figure 8).

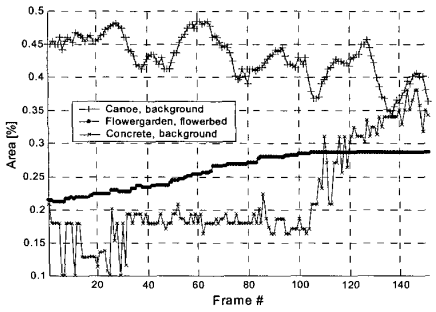

Figure 8: Biggest identified detail-irrelevant frame area w.r.t. time of the test sequences Canoe, Flowergarden and Concrete

The area covered by the flowerbed in the Flowergarden sequence continuously rises with time, which is reflected by the increase of the corresponding curve (cp. Figure 8). The increase of the identified stone wall area in the Concrete video is even more substantial (cp. Figure 8) and relates to the fact that a zoom out occurs in this sequence. The resolution with which the stones are shown thus continuously decreases and the wall becomes more and more detail-irrelevant. The texture identification at the beginning of the sequence is quite poor, since the stones in the background are shown with a very high resolution. It can be seen in Figure 8 that the background (rocks, trees, tree trunks) of the Canoe sequence covers almost 50% of the frame area in some frames.
Note that ensuring temporal stability of the borders between synthesized and natural textures is beyond the scope of this paper. However, a solution concerning this issue was already proposed in (1).

5. CONCLUSIONS

A segmentation algorithm for content-based video coding was presented. The underlying assumption of our approach is that the textures in a video scene can be labeled subjectively relevant or irrelevant. Two MPEG-7 descriptors, a texture (Edge Histogram, EH) and a color descriptor (SCalable Color, SCC), are used for similarity estimation. Our experiments show that the average area of the correctly identified detail-irrelevant textures represents 61% (EH) and 70% (SCC) of the total area of the considered test images. Considering false negatives as non-erroneous segmentation even yields an identification rate of 93% (EH) and 92% (SCC). I.e. the proportion of false positives is very low for both descriptors.

6. REFERENCES

1. Ndjiki-Nya P., Makai B., Blättermann G., Smolic A., Schwarz H. and Wiegand T., 2003, Proc. of ICIP, Vol. 3, 849-852.

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

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

4. Yoon S.-Y. and Adelson E. H., 1998, Proc. of SPIE on HVEI III, Vol. 3299, 489-497.

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

6. Wollborn M., 1994, IEEE Transactions on CSVT, Vol. 4, 236-245.

7. Freixenet J., Muñoz X., Raba D., Martí J. and Cufí X., 2002, Proc. of ECCV, Vol. 2352, 408-22.

8. Malki J., Boujemaa N., Nastar C. and Winter A., 1999, Proc. of VISUAL, 115-22.

9. Rubner Y., Tomasi C. and Guibas L. J., 1998, Proc. of ICCV, 207-214.

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