MODELS FOR TEXTURE BASED VIDEO CODING

Marc Bosch †, *Fengqing Zhu* † *and Edward J. Delp* † ‡

†Video and Image Processing Lab (*VIPER*)
School of Electrical and Computer Engineering
Purdue University, West Lafayette, Indiana, USA

‡Tampere University of Technology
Department of Signal Processing
Tampere International Center for Signal Processing (TICSP), Tampere, Finland

ABSTRACT

One way to increase compression efficiency beyond the data rates achievable by modern video codecs is to not encode all the pixels. In particular, regions belonging to areas that the viewer will not perceive the specific details in the scene could be skipped or encoded at a much lower data rate. This approach can be extended by using a model of the Human Visual System and a statistical model of the texture pixels in a frame. The goal was to determine where "insignificant" texture regions or "detail-irrelevant" regions in the frame are located and then use a texture model for the pixels in the region. By "insignificant" pixels we mean regions in the frame that the observer will not notice are different without observing the original video sequence. We will describe the texture modeling approach we use and show that using this approach can increase the compression efficiency by more that 15% over more classical approaches.

Index Terms— coding efficiency, texture analysis, video coding.

1. INTRODUCTION

In the past two decades, conventional hybrid video codecs have been succeeded in increasing the video quality while reducing the data rate [1]. One interesting approach to reduce the data rate is to use a different coding method for pixels belonging to areas with large amount of details that are costly to encode. In early video coding systems this was achieved by either reducing the size of the frame and/or a combination of frame skipping. In MPEG-4, shape coding is used to code shapes in a frame after they are segmented. In many cases the shapes in a frame consist of important objects in the scene and poor segmentation of the objects can cause problems in the reconstructed video. Further developments introduce new coding techniques that focus on the semantic meanings of objects represented in the video sequence. As a result, one way to accomplish such a goal beyond the data rates achievable by modern codecs is to not encode all the pixels. In particular, regions belonging to areas that viewer does not perceive the specific details could be skipped or encoded at a much lower data rate. In 1959, Synthetic Highs was proposed [2] which introduced the concept of dividing an image into textures and edges. Two approaches were described to encode each type of structure in the image. This approach, used in image coding, was later extended by using a model of the Human Visual System (HVS) and a statistical model of the texture pixels in a frame [3, 4]. The goal is to determine where "insignificant" texture regions or "detail-irrelevant" regions in the frame are located and then use a texture model for the pixels in these regions. By "insignificant" pixels we mean regions in a frame that the observer will not notice what has been changed. The encoder fits the model to the image and transmits the model parameters to the decoder as side information which uses the model to reconstruct the pixels. Since a frame is not homogeneous one may need to use different types of models for various texture regions in a frame An example of texture based methods is described in [1, 5, 6], where coding methods are described using the concept that textures such as grass, water, sand, and flowers can be synthesized with acceptable perceptual quality instead of coding them using more classical approaches. The problem with using this approach in video is that if each frame is encoded separately the areas that have been reconstructed with the texture models will be obvious when the video is displayed. This then requires that the texture to be modeled both spatially and temporally. An example of such approach, reported by Wiegand and colleagues, is described in [1, 7], where a video coder was designed using the fact that textures such as grass, water, sand, and flowers can be synthesized with acceptable perceptual quality instead of coding them using mean square error. Since the latter has a higher data rate in order to represent the details in the textures which are not visually important, the approach can be used to increase the overall coding efficiency. The issues then are the trade-offs between data rate, modeling efficiency, and image quality.

2. MODEL OVERVIEW

2.1. Overview

Our goal is to not encode all the pixels in the sequence. Particularly, those pixels belonging to areas with large amount

This work was partially sponsored by the Academy of Finland as part of the Finland Distinguished Professor Program (FiDiPro). Address all correspondence to E. J Delp, delp@cs.tut.fi

of details that are costly to encode. This approach is a mid-level content-based video coding scheme. That is, regions with similar homogeneous motion, color and texture properties are processed together. The semantic interpretation of these regions is thereby irrelevant. Rather, it is assumed in the framework that video content can be divided into perceptually relevant and perceptually irrelevant textures. Consequently, for these perceptually irrelevant textures, the semantic meaning of the displayed texture is more relevant to the viewer than the specific details therein. In the present work it is assumed that for mean square error (MSE) distortion criteria is not suitable for efficient video coding, since irrelevant detail textures may be reproduced and they are costly to code. A general

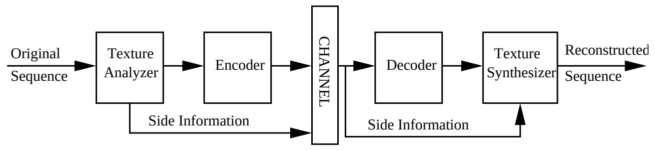


Fig. 1. Texture Coding System Overview.

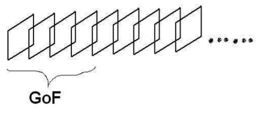
scheme for video coding using texture analysis and synthesis is illustrated in Figure 1. The texture analyzer identifies homogeneous regions in a frame and labels them as textures. This step can be done by using several texture segmentation strategies that will be introduced later. To ensure the temporal consistency of the identified textures throughout the video sequence, global motion models are used to warp texture regions from frame-to-frame. A set of motion parameters for each texture region is sent to the texture synthesizer at the decoder as side information. The output of the texture analyzer is passed to a conventional video codec, e.g. H.264, with synthesizable regions labeled as skip macroblocks. At the decoder, frames are partially reconstructed except for the synthesizable parts which are inserted later by the texture synthesizer using the texture side information.

2.2. Texture Analysis

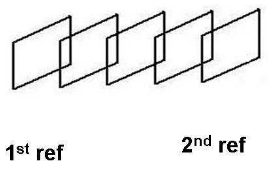
Spatial Analysis
The first step in implementing the system described in Figure 1 is to extract regions in a frame that correspond to the same texture. In texture analysis there are two major issues that need to be addressed, namely texture feature extraction and texture boundary detection or segmentation. Texture segmentation is often a two step process in which features are first obtained, followed by the segmentation step which is a "grouping" operation of the homogeneous regions based on the feature properties and a grouping metric [8, 9]. Feature extraction is used to measure local texture properties in an image. Typically, four approaches have been used for texture feature extraction: statistical or feature-based methods, model-based methods, transform or spatial-frequency methods and structural methods. In feature-based methods, characteristics of homogeneous regions are chosen as the texture features such as the co-occurrence matrix or geometrical features such as edges [8]. Model-based methods assume that the texture is described by a stochastic model and uses model parameters to segment the texture regions. Examples of such methods are found in [9] where a multiresolution Gaussian autoregressive model is described and in [4] where an image model is formulated using a seasonal autoregressive time series. Subband decomposition, especially the use of wavelets, is often seen in spatial-frequency methods [10]. Structural methods are based on the notion that textures are composed of primitives that are well-defined and spatially repetitive [11, 12]. Boundary detection or segmentation is followed to group the features into regions with similar texture properties. In our previous work [5], we have concentrated on examining spatial texture models that are based on simple features and statistical models, such as color and edges. We have also studied statistical and transform models, including gray level co-occurrence matrix and Gabor filter [6]. The color feature we examined is a color histogram defined in the Hue-Saturation-Value color space with fixed color space quantization. We used the Kirsch edge operator as our edge detector. The Gray Level Co-occurrence Matrix describes the spatial relation between pixels and their neighbors by means of the occurrence probability of each pair of gray levels of the image. Gabor filters describe various features related to the local power spectrum of a signal. Once we computed the characteristics of the image, segmentation is performed to divide the image into different regions based on their properties, these methods include direct segmentation techniques or classification methods.

Temporal Analysis
The spatial texture models referred in the previous section operate on each frame of a given sequence independently of the other frames of the same sequence. This yields inconsistency in the segmentation across the sequence and can be very noticeable when the video is viewed. One can address this problem by using spatial-temporal texture models [13] or using something similar to motion compensation for the texture models in each frame [1]. In order to maintain temporal consistency of the texture regions, the video sequence is first divided into groups of frames (GoF). Each GoF consists of two key frames (the first and last frame of the GoF) and several middle frames to be modeled with textures between the two key frames. This is illustrated in Figure 2.

The key frames will either be I or P frames when they are coded. For every texture region in each of the middle frames we look for similar textures in both key frames. The corresponding region (if it can be found in at least one of the key frames) is then mapped into the segmented texture region. There are three possible cases, the texture is only found in the first key frame, the last key frame, or it is found in both key frames. This is illustrated in Figure 3. In most cases, similar textures can be found in both key frames. In this case, the



(a) GOF



(b) Key Frames

Fig. 2. Illustration of Group of Frames(GOF). (a) A sequences of GOFs, (b) Two key frames within a GOF.

texture resulting from the smallest error will be considered.

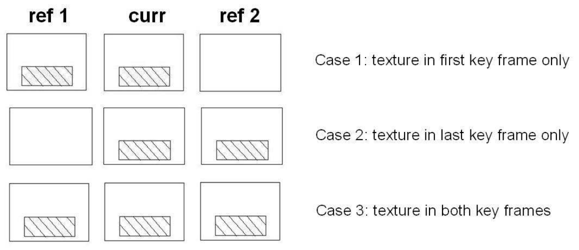

ref 1
curr
ref 2
Case 1: texture in first key frame only
Case 2: texture in last key frame only
Case 3: texture in both key frames

Fig. 3. Choice of Key Frame.

The texture regions are warped from frame-to-frame using a motion model to provide temporal consistency in the segmentation as illustrated in Fig.4. The mapping is based on a global motion assumption for every texture region in the frame, *i.e.*, the displacement of the entire region can be described by just one set motion parameters. We modified a 8-parameter (i.e. planar perspective) motion model to compensate the global motion [14]. This can be expressed as:

$$\begin{aligned} x' &= \frac{a_1 + a_3 x + a_4 y}{1 + a_7 x + a_8 y} \\ y' &= \frac{a_2 + a_5 x + a_6 y}{1 + a_7 x + a_8 y} \end{aligned} \quad (1)$$

Where (x, y) is the location of the pixel in the texture frame and (x', y') is the corresponding mapped coordinates. The planar perspective model is suitable to describe arbitrary rigid object motion if the camera operation is restricted to rotation and zoom. It is also suitable for arbitrary camera operation, if the objects with rigid motion are restricted planar motion. In practice these assumptions often hold over a short period of a GoF [5].

When an identified texture region in one of the texture frames is warped toward the key frame of the GoF, only the pixels of the warped texture region that lie within the corresponding texture region of the key frame of the GoF are used for synthesis. Although this reduces the texture region in the texture frame, it is more conservative and usually gives better results in terms of visual quality.

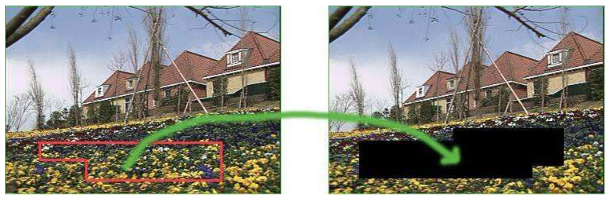


Fig. 4. The Motion Model is Used For Warping the Texture From The Key Frame to Texture Frame.

The motion parameters $(a_0, a_1, \ldots, a_8)$ are estimated using a simplified implementation of a robust M-estimator for global motion estimation [14]. The weighting function is simplified to a rectangular function, i.e. a point is either weighted fully or considered as outliers and are discarded:

$$w(\epsilon) = \begin{cases} 1 & \epsilon^2 < c\mu \\ 0 & \epsilon^2 > c\mu \end{cases} \quad (2)$$

where ϵ, the residual, is obtained from the difference of the luminance between the actual and the warped pixels. The sum of squares of all the residuals except the outliers is minimized. Then μ is the average sum of squares of all N points within region R,

$$\mu = \frac{1}{N} \sum_{p \in R} \epsilon^2 \quad (3)$$

c is used to adjust the sensitivity of the optimization. With an increasing value of c, more pixels in both structure and unstructured areas are used for the iteration.

We estimated the set of parameters using an iterative Gauss-Newton method. The process of warping and optimization is done for both key frames, hence two sets of motion parameters are estimated (each set corresponds to a key frame) and the set that has smallest MSE between the synthesized and original texture region is used We used as a “stopping criterion” the first minimum obtained in the cost function μ.

Unstructured image regions can have a bad impact on the motion parameter estimates results. The residuals will be small in unstructured areas even if the motion parameters estimates are poor. However, areas such as edges are likely to result in large areas even if the estimation is good. To ensure the adequate performance of the estimator, the unstructured regions are detected and excluded from the estimation. The pixels in the unstructured regions are obtained by:

$$V = \begin{cases} 1 & |I_x| > d \cap |I_y| > d \\ 0 & \text{else} \end{cases} \quad (4)$$

In some cases a better estimate of the motion parameters may be obtained by first interpolating pixels and then doing the motion estimation [15]. The interpolation method is the same as the H.264 sub-pixel interpolator. We used bilinear interpolation for half pixels for the Y samples on all frames

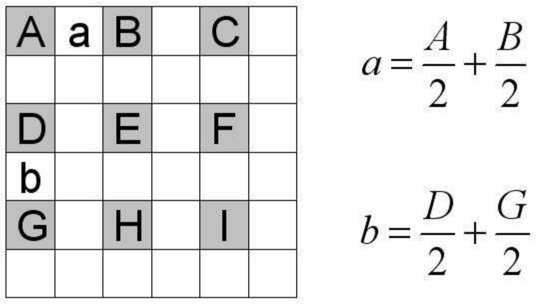

A
a
B
C
D
E
F
b
G
H
I
$a = \frac{A}{2} + \frac{B}{2}$
$b = \frac{D}{2} + \frac{G}{2}$

Fig. 5. Bilinear Interpolation.

for our motion model. An example of the interpolated Y samples is shown in Figure 5. The set of motion parameters along with the flag to indicate the corresponding key frame are sent to decoder as side information.

2.3. Texture Synthesis and Integration into H.264/AVC

At the decoder, reference frames and non-synthesizable parts of other frames are conventionally decoded. The remaining parts labeled as synthesizable regions are skipped by the encoder and their values remain blank in the conventional coding process. The texture synthesizer is then used to reconstruct the corresponding missing pixels. With the assumption that the frame to frame motion can be described using a planar perspective model, then given the motion parameter set and the control parameter that indicated which frame (first or last frame of the GoF) is used as the key frame, the texture regions can be reconstructed by warping the texture from the key frame toward each synthesizable texture region identified by the texture analyzer. As for the integration process the texture models described in the previous sections have been integrated into the H.264/AVC JM 11.0 reference software. In our implementation, the video sequence is first divided into groups of frames (GoF). Each GoF consists of two reference frames (first and last frame of the considered GoF) and several middle frames between the two reference frames. The reference frames are conventionally coded as I or P frames; the middle frames are encoded as B frames that are candidates for texture synthesis. For every texture region in each of the middle frames, the texture analyzer looked for similar textures in both reference frames. The corresponding area (if it can be found in at least one of the reference frames) is then mapped into the segmented texture region based on a global motion model. When a B frame contains identified synthesizable texture regions, the corresponding segmentation masks, motion parameters as well as the control flag to indicate which reference frame is used are transmitted as side information to the decoder. All macroblocks belonging to a synthesizable texture region are handled as skipped macroblocks in the H.264/AVC reference software. Hence, all parameters and variables used for decoding the macroblocks inside the slice, in decoder order, are set as specified for skipped macroblocks.

This includes the reconstructed YUV samples that are needed for intra prediction, the motion vectors and the reference indices that are used for motion vector prediction. After all macroblocks of a current frame are completely decoded, texture synthesis is performed in which macroblocks belonging to a synthesizable texture region are replaced with the textures identified in the corresponding reference frame.

Table 1. Data Rate Savings Obtained for Different Sequences. Using a Quantization Parameter of 24.

Sequence	Original data rate [kb/s]	Data Rate Savings Texture-based
Tabletennis	840.49	15.22%
Flowergarden	2399.31	20.23%
Coastguard	2593.82	16.02%
Football	1962.14	16.45%

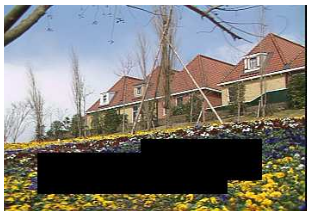
3. RESULTS AND CONCLUSION

We used our spatial and temporal texture models to examine several video sequences. We are interested in measuring the performance of the models we tested in terms of coding efficiency, for detailed results see table 1. To estimate the new data rate for each test sequence, we substract from the original data rate (coded with the H.264 test coder) the data rate savings for macroblocks that are not coded using H.264 codec. The data rate used to construct the side information, which is no more than 1.25Kb per frame, is then added to this amount to obtain the new data rate. Such side information contains the coarse texture masks (typically about 800 bits), 8 motion parameters (256 bits) which ensure temporal consistency at the decoder and 1 control flag (1 bit) to indicate which frame is used as the reference frame (the first or the last frame of the GOF). In our experiments, we used the following parameters for the Main Profile H.264/AVC codec: Quantization Parameter for intra and inter frames = 16, 20, 24, 32 and 44; 1 reference frame; 3 B frames; CABAC; rate distortion optimization; no interlace; constant channel; 30 frames per second.

Figures 6 and 7 shows how the textures are generated in each frame using the motion model. To reduce the blockiness effect along the edges, we used a low pass filter on the edge pixels. A visual artifact may be created if the intensity of the light in the environment changes over time. This happens in Coast Guard sequence. To fix this problem, we transmit the difference between the average intensity of the texture region in the current frame and the reference frame as additional side information. At the synthesizer this will be added to the intensity of the synthesized pixels.

(a) Reference Frame

(b) Current Frame


(c) Current Frame with Mask

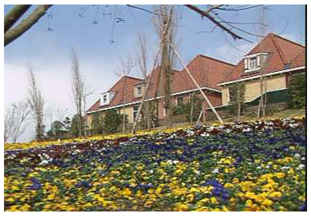
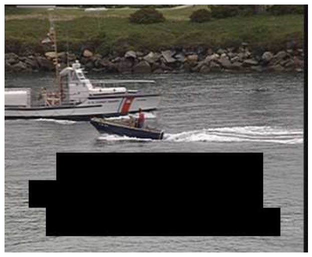

(d) Reconstructed Frame

Fig. 6. Example of motion model applied to Flower Garden sequence (a) Frame 61 used as the Reference Frame, (b) Frame 64 used as the Current Frame, (c) Frame 64 with Mask for texture region and (d) Reconstructed Frame 64 with synthesized texture.

Methods to separate "insignificant" regions from the frame were investigated and were incorporated into a conventional video codec (e.g. H.264) where the regions modeled by the global motion model were not coded in an usual manner. We have shown that we can reduce the data rate by as much as 15% when compared with more classical approaches such as H.264.

(a) Reference Frame

(b) Current Frame


(c) Current Frame with Mask

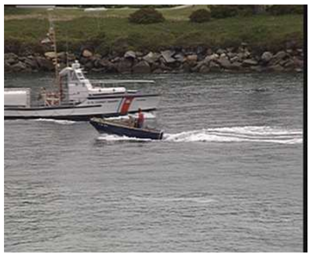

(d) Reconstructed Frame

Fig. 7. Example of motion model applied to Coast Guard sequence (a) Frame 41 used as the Reference Frame, (b) Frame 45 used as the Current Frame, (c) Frame 45 with Mask for texture region and (d) Reconstructed Frame 45 with synthesized texture.

4. REFERENCES

[1] P. Ndjiki-Nya, C. Stuber, and T. Wiegand, "Improved video coding through texture analysis and synthesis," *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*, Lisboa, Portugal, April 2004.

[2] W. F. Schreiber, C. F. Knapp, and N. D. Kay, "Synthetic highs, an experimental tv bandwidth reduction system," *Journal of Society of Motion Picture and Television Engineers*, vol. 68, pp. 525–537, August 1959.

[3] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 549–574, April 1985.

[4] E. J. Delp, R. L. Kashyap, and O. Mitchell, "Image data compression using autoregressive time series models," *Pattern Recognition*, vol. 11, pp. 313–323, June 1979.

[5] F. Zhu, K. Ng, G. Abdollahian, and E. J. Delp, "Spatial and temporal models for texture-based video coding," *Proceedings of the Visual Communications and Image Processing Conference*, San Jose, California, January 2007.

[6] M. Bosch, F. Zhu, and E. Delp, "Spatial texture models for video compression," *Proceedings of ICIP, IEEE International Conference on Image Processing*, San Antonio, Texas, September 2007.

[7] P. Ndjiki-Nya, B. Makai, A. Smolic, H. Schwarz, and T. Wiegand, "Improved h.264/avc coding using texture analysis and synthesis," *Proceedings of ICIP, IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.

[8] I. Elfadel and R. Picard, "Gibbs random fields, cooccurrences, and texture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 24 – 37, January 1994.

[9] M. L. Comer and E. J. Delp, "Segmentation of textured images using a multiresolution gaussian autoregressive model," *IEEE Transactions on Image Processing*, vol. 8, pp. 408 – 420, March 1999.

[10] J. R. Smith and S.-F. Chang, "Quad-tree segmentation for texture-based image query," *accpeted for ACM Multimedia*, San Francisco, CA., October 1994.

[11] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[12] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Reading, MA: Addison-Wesley, 1992.

[13] C.-H. Peh and L.-F. Cheong, “Synergizing spatial and temporal texture,” *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1179–1191, October 2002.

[14] A. Smolic and J. Ohm, “Robust global motion estimation using a simplified m-estimator approach,” *Proceedings of the IEEE International Conference on Image Processing*, Vancouver, Canada, September 2000.

[15] *Text of Committee Draft of Joint Video Specification*, ITU-T and ISO/IEC ITU-T Recommendation H.264—ISO/IEC 14496-10 AVC(MPEG-4 Part 10), 2002.