Coding Transparency in Object-Based Video

Aghito, Shankar Manuel; Forchhammer, Søren

Published in:
Proceedings of the 7th Nordic Signal Processing Symposium

Link to article, DOI:
10.1109/NORSIG.2006.275236

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

Coding Transparency in Object-Based Video

Shankar Manuel Aghito[†] *and Søren Forchhammer*

Technical University of Denmark
Department of Communications, Optics and Materials
DTU, Ørsteds Plads 345V, 2800 Kgs. Lyngby
DENMARK
Tel: +45 4525 3641[†], Fax: +45 4593 6581
E-mail: sma@com.dtu.dk[†], sf@com.dtu.dk

ABSTRACT

A novel algorithm for coding gray level alpha planes in object-based video is presented. The scheme is based on segmentation in multiple layers. Different coders are specifically designed for each layer. In order to reduce the bit rate, cross-layer redundancies as well as temporal correlation are exploited. Coding results show the superior efficiency of the proposed scheme compared with MPEG-4.

1. INTRODUCTION

In MPEG-4 object-based video the shape and the transparency of video objects are represented by gray level alpha plane sequences, allowing composition of multiple objects. Shape coding has been studied intensively [1], resulting in the adoption of Context-based Arithmetic Encoding (CAE) into MPEG-4 [2]. The Digital Straight Line Segments Coder (DSLSC) [3, 4] was recently proposed as a very efficient alternative to CAE. Although the use of transparency is desirable for smooth composition, little documented work has been aimed at efficient coding of the transparency information [5, 6]. In MPEG-4 transparency is coded with the same techniques used for texture (i.e. luminance and chrominance), namely the Shape Adaptive Discrete Cosine Transform (SA-DCT). Since the characteristics of texture and alpha planes are very different, a coder designed for alpha plane should increase the efficiency.

In this work several new strategies for coding the transparency of video objects are presented. This includes a segmentation of the source data in multiple layers, and novel algorithms to encode the different layers. The segmentation scheme is described in Section 2. The schemes for coding the different layers are described in Sections 3 and 4. Results compared with MPEG-4 are reported in Section 5. Conclusive remarks are given in Section 6.

2. PROPOSED ARCHITECTURE

The composition of video objects is shortly described. Consider a video resolution of N_c columns and N_r rows; given a single frame of a video object, let $\alpha(x,y)$ be the alpha component, with values $0 \leq \alpha \leq 255$ and domain $D = \{1,\ldots,N_c\} \times \{1,\ldots,N_r\}$; let $Y(x,y)$ and $Y_B(x,y)$ be the corresponding luminance component and the luminance of the background object, respectively. The luminance of the composed scene is given by

$$Y_C(x,y) = Y(x,y)\frac{\alpha(x,y)}{255} + Y_B(x,y)\frac{255-\alpha(x,y)}{255}. \quad (1)$$

In the proposed scheme the alpha plane $\alpha(x,y)$ is segmented in background layer (L_0), opaque layer (L_{255}) and intermediate layer (L_{int}), as follow:

$$\begin{aligned} L_0 &= \{(x,y) \in D \mid \alpha(x,y) = 0\}, \\ L_{255} &= \{(x,y) \in D \mid \alpha(x,y) = 255\}, \\ L_{\text{int}} &= \{(x,y) \in D \mid 0 < \alpha(x,y) < 255\}. \end{aligned} \quad (2)$$

Note that the background layer L_0 is the complement of the binary shape layer, indicated as $\overline{L}_0$. The coding flow is depicted in Fig. 1. First, the binary shape is directly encoded with DSLSC [3, 4]. In order to exploit the correlation between layers, L_{255} is encoded referencing L_0, and L_{int} is encoded referencing both L_0 and L_{255}, as described in the following sections. Implementation details can be found in [7].

3. CODING THE OPAQUE LAYER

The opaque layer L_{255} is encoded using the knowledge of the background L_0, as illustrated in Fig. 1, exploiting the strong correlation among the two layers which is observed for a large class of alpha planes. The proposed technique consists in representing L_{255} as a morphological erosion of $\overline{L}_0$. The algorithm is organized in a block-based manner, and can be iterated for decreasing block sizes. The strength of the erosion is the key information to be encoded locally for each

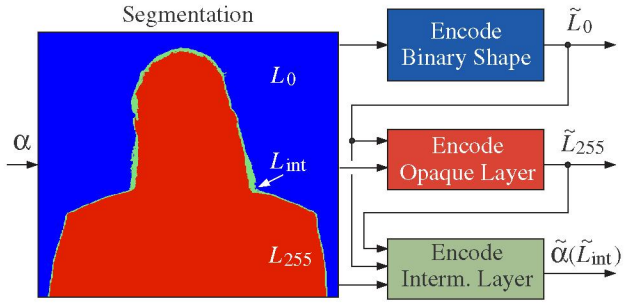

Fig. 1. The proposed architecture.

Table 1. The different block types and the assigned variable length codes.

type	description	C_1	C_2
TR	all pixels are in L_0	-	-
OP	all pixels are in L_{255}	11	-
SK	not TR, not OP, $e(\epsilon_{\text{opt}}) > e_{\max}$	10	1
ER	not TR, not OP, $e(\epsilon_{\text{opt}}) \leq e_{\max}$	0	0

block. The intra (each frame coded independently) algorithm is firstly presented, and the modifications for inter coding (exploiting temporal correlation) are given afterwards.

3.1. Block classification

Blocks that are entirely within L_0 are not processed, since they were encoded by the binary shape coder. These blocks are referred to as *transparent* (TR). Blocks that are entirely in L_{255} are classified as *opaque* (OP). The remaining blocks are classified as *eroded* (ER) or *skipped* (SK), as follow. Local (on each block) approximations of L_{255} are obtained by eroding $\overline{L_0}$, using filled circles as structuring element. The optimal radius ϵ_{opt} is found within a predefined set $S_\epsilon = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_{N_\epsilon}\}$ (N_ϵ is a power of 2), such that the approximation error is minimized. If the approximation error $e(\epsilon_{\text{opt}})$ is not greater than a preselected threshold $e_{\max}$, then the block is marked as ER, otherwise the block is SK. For each block that is not TR, the distance d_B from the block to L_0 is defined as the shortest Euclidean distance from pixels in the block to pixels in L_0. The algorithm searches for the integer $d_B^\star$, such that all blocks with distance d_B not smaller that $d_B^\star$ are OP. If $d_B^\star$ is found, a 0 is coded followed by a binary representation of $d_B^\star$. In this case only blocks with distance $d_B < d_B^\star$ need to be encoded. If $d_B^\star$ cannot be found, a 1 is inserted, indicating that all non TR blocks must be coded. The classification criteria and the variable length codes (VLC) for each block type are listed in Table 1. The code C_1 is used by default, while C_2 is used when in every block there is at least one pixel $p \in L_0$, since in this case there are no OP blocks. After the classification is coded, the different blocks are encoded as follow.

Table 2. The construction of the variable length code used to represent Δ_ϵ for predictively encoded ER blocks. I$(\cdot,\cdot)$ indicates *innovation codes* [8].

Δ_ϵ	code
-1	00
+1	01
0	10
$1 < \Delta_\epsilon < N_\epsilon$	11 0 I$(N_\epsilon - \Delta_\epsilon - 1, N_\epsilon - 2)$
$-N_\epsilon < \Delta_\epsilon < -1$	11 1 I$(N_\epsilon + \Delta_\epsilon - 1, N_\epsilon - 2)$

3.2. Coding opaque blocks

Opaque blocks are implicitly encoded in the classification process described above. No additional information is required for these blocks.

3.3. Coding eroded blocks

The optimal radius ϵ_{opt} is coded for each ER block. Blocks are processed in raster scan order. A differential scheme is employed. For each ER block the candidate prediction blocks are, in the given order: the block above, the block to the left, and the block in the up-right position. The first available candidate which is ER is selected as prediction block. If there are no ER candidates, ϵ_{opt} is simply coded by a $\log_2(N_\epsilon)$ bits representation of its index in the ordered set S_ϵ, indicated as $\mathtt{i}(\epsilon_{\text{opt}}, S_\epsilon)$. If a prediction block is found, ϵ_{opt} is encoded differentially with respect to the optimal radius ϵ_p of the prediction block. The difference between the indexes $\Delta_\epsilon = \mathtt{i}(\epsilon_{\text{opt}}, S_\epsilon) - \mathtt{i}(\epsilon_p, S_\epsilon)$, is coded using the VLC defined in Table 2.

3.4. Coding skipped blocks

Skipped blocks could not be approximated as an erosion of the shape $\overline{L_0}$. The procedure described above is iterated for decreasing block sizes; each skipped block (level 1) is divided in 4 sub-blocks; the sub-blocks are also classified as TR, OP, ER or SK and coded as described above. Any skipped sub-blocks (level 2) is divided again, and so on. The iteration stops at the desired level. In our implementation we used 4 levels, with blocks of size 162×180, 81×90, 27×45, and 9×45. If there are skipped sub-blocks also at the last level, these sub-blocks are encoded with DSLSC.

3.5. Inter coding

Temporal correlation is exploited by introducing and additional block type, i.e. *inter* blocks. The algorithm is modified as follow: for each non TR block, the motion compensated prediction (MCP) is first calculated, using motion vectors estimated on the binary shape layer (hence without requiring a overhead). If the result of motion compensation is good enough, i.e. if the error is within $e_{\max}$, then the block is classified as

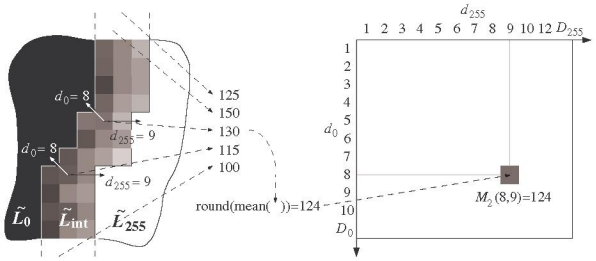

Fig. 2. The construction of the distance map M_2. Each element $M_2(j,k)$ is calculated using (4), as illustrated in this example for $j = 8$ and $k = 9$.

inter and represented as the MCP, otherwise the intra algorithm is used.

4. CODING THE INTERMEDIATE LAYER

Several strategies for coding the transparency levels $\alpha(L_{\text{int}})$ in the intermediate layer are proposed, given that the (coded versions of the) two previous layers L_0 and L_{255} are known.

4.1. SA-DCT with modified support

Since L_{255} is known, the conventional SA-DCT algorithm may be utilized to encode only the pixels in L_{int}, instead of all the pixels in $\overline{L_0}$ as done in MPEG-4.

4.2. Absolute distances map

The transparency levels $\alpha(L_{\text{int}})$ are represented as a function of their relative position with respect to L_0 and L_{255}. The basic idea is to replace all the pixels with a certain relative position with their average value. The information to be encoded is only the average value for each possible relative position.

Let $d^E(a,b)$ be the Euclidean distance between two pixels a and b. Let $d_i^E(a) = \min_{b \in L_i} d^E(a,b)$ be the Euclidean distance between a pixel $a \in L_{\text{int}}$ and $L_i, \forall i \in \{0, 255\}$. The parameter of interest is defined as

$$d_i(a) = \begin{cases} 9 - N_i(a) & \text{if } d_i^E(a) < 2 \\ \lceil d_i^E(a)/Q_i \rceil + 8 & \text{if } d_i^E(a) \geq 2, \end{cases} \quad (3)$$

where $N_i(a)$ is the number of 8-connected neighbors of the pixel a which are equal to i, Q_i is the quantization step used for layer L_i, and $\lceil \cdot \rceil$ is upward rounding. Using d_i instead of d_i^E allows for a more precise differentiation of the pixels contiguous to L_i, while the information for the other pixels is quantized. Let $D_0 = \max_{p \in L_{\text{int}}} d_0(p)$ and $D_{255} = \max_{p \in L_{\text{int}}} d_{255}(p)$. Let the matrix M_2 be constructed with elements

$$M_2(j,k) = \lceil \mathtt{m}\{\alpha(p) \mid p \in L_{\text{int}},\ d_0(p) = j,\ d_{255}(p) = k\} \rfloor$$
$$\forall (j,k) \in \{1,2,\ldots,D_0\} \times \{1,2,\ldots,D_{255}\}, \quad (4)$$

where $\lceil \cdot \rfloor$ is rounding to the nearest integer; $\mathtt{m}(\cdot)$ is the mean value, or 0 if the argument is empty. The matrix M_2 is the information to code, since it contains the average α value for every possible pair (d_0, d_{255}).

4.2.1. Coding the matrix M_2

First, unimportant rows (or columns) of M_2 are removed, i.e. if the number of pixels in L_{int} with the corresponding d_0 (or d_{255}) is less than a selected threshold. Let M_2' be the reduced matrix, and L_{int}' the corresponding subset of L_{int}. The reduced matrix M_2' can be encoded with any lossless/lossy image coding scheme, e.g. MPEG-4 SA-DCT, adaptive quantization [6] (optional) followed by lossless/lossy shape adaptive BMF [9], with shape given by the support of M_2'.

4.2.2. Reconstruction of the layer L_{int}

M_2' is decoded, giving the approximation $\widetilde{M'}_2$; interpolating the removed rows and columns from the remaining ones the full size matrix $\widetilde{M}_2$ is reconstructed. The transparency levels are reconstructed as

$$\widetilde{\alpha}(p) = \widetilde{M}_2(d_0(p), d_{255}(p)), \forall p \in L_{\text{int}}. \quad (5)$$

Finally, a simple low pass linear filter is applied.

4.2.3. Extensions

A more precise reconstruction is obtained augmenting the model by adding a third variable: the three dimensional matrix M_3 is constructed in similar way as in (4) by considering, beside d_0 and d_{255}, also a third variable representing the orientation of the pixels in L_{int} with respect to the other two layers.

If a higher quality is desired, the obtained reconstruction of $\widetilde{\alpha}(L_{\text{int}})$ is utilized as prediction for further coding. The MPEG-4 SA-DCT scheme is utilized to encode the prediction error.

Although the current implementation provides only intra coding, inter coding should be considered since it is reasonable to expect high correlation between matrixes obtained from consecutive alpha plane frames.

4.3. Quantization and context-based coding

The α levels are quantized in the spatial domain and then coded with context-based arithmetic encoding. The idea was introduced in [5], refined in [6], and it is here further extended by inserting it in the proposed three layer architecture: hence it applies only to the pixels in L_{int}; the quantization is the same implemented in [6]; more efficient context-based arithmetic encoding is obtained by using the (shape adaptive modified version of the) BMF coder [9]. Inter coding is achieved by sharing the statistics of the arithmetic encoder over a number of consecutive frames.

5. RESULTS

Experimental results demonstrating the efficiency of the proposed coding techniques are presented. Six test sequences (720×486 pixels, 30 Hz) from the MPEG-4 test set are utilized: akiyo, children, bream, weather, layers 5 (robot) and 7 of total destruction. Two representative cases are analyzed in Fig. 3, which provides the rate-distortion performance of the different algorithms, for the sequences bream (intra case) and robot (inter case). The bit rates include coding of L_{255} and L_{int} (calculated on the first 10 frames of each sequence); the PSNR is measured within the shape $\overline{L_0}$. In the legend, MPEG-4 indicates the SA-DCT scheme as applied in the standard, i.e. to all the pixels in $\overline{L_0}$. All the other solutions use the opaque layer coder as described in Section 3, with the parameter $e_{\max} = 10$. DM indicates the distance map method (Q_1 and Q_{255} are set to 1), MS SA-DCT indicates the modified support SA-DCT algorithm described in Section 4.1, and Q+BMF indicates the adaptive quantization followed by BMF coding presented in Section 4.3. All the proposed algorithms outperform MPEG-4. The best performance at the low rates is typically obtained by DM, while Q+BMF is the best at higher rates. For mid-range rates, these two algorithms provide typical bit rate reductions of $50-70\%$ in intra mode and $40-70\%$ in inter mode, with respect to MPEG-4; the MS SA-DCT is less efficient, but still provides rates 40% smaller than those obtained with MPEG-4. More detailed results are presented in [7].

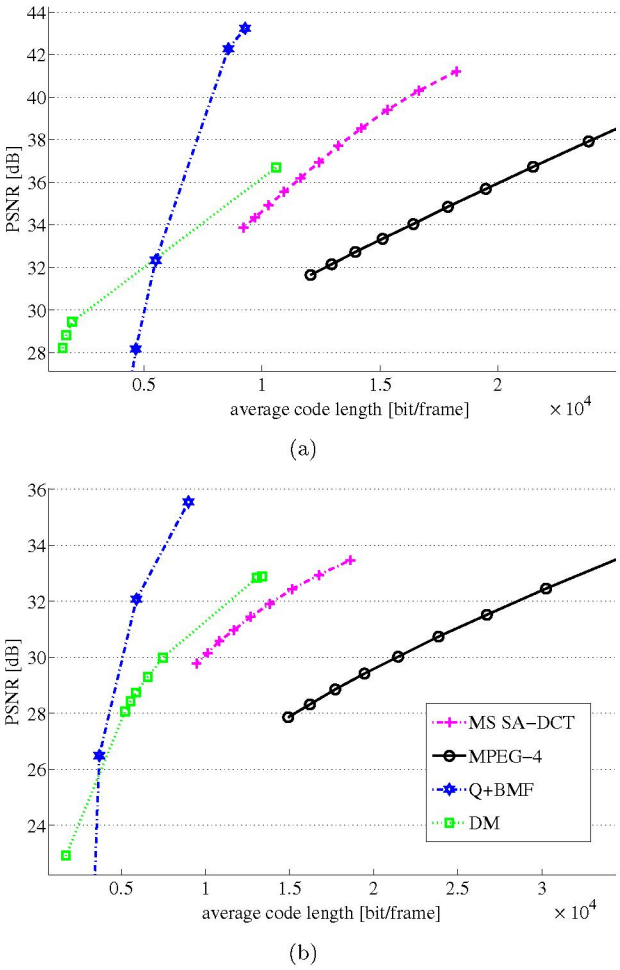

Fig. 3. The proposed methods compared to MPEG-4: the sequence bream coded in intra mode(a), and the sequence robot coded in inter mode (b).

6. CONCLUSIONS

A new architecture for encoding both the shape and the transparency information in gray level alpha planes for object-based video was proposed. The data is segmented in multiple layers: the binary shape, the opaque layer and the intermediate layer. The shape is encoded with DSLSC [3, 4]. The opaque layer is encoded with a new strategy based on block-based partitioning and morphological erosion of the previously encoded binary layer. Several techniques are proposed for coding the intermediate layer: the modified support SA-DCT, a new algorithm based on calculation of the distance map respect to the two previous layers, and a simple scheme based on adaptive quantization and context based arithmetic encoding. The proposed techniques outperform those of the MPEG-4 standard.

In the future, the impact of the proposed techniques on the overall rate-distortion performance in object-based video should also be evaluated, i.e. taking also into account coding of texture information.

7. REFERENCES

[1] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann and G. M. Shuster, "MPEG-4 and rate-distortion-based shape-coding techniques," *Proc. IEEE*, vol. 86, no. 6, June 1998, pp. 1126-1154.

[2] ISO/IEC Int'l Standard 14496-2, "Information technology – Coding of audio-visual objects – Part 2: Visual," 1999.

[3] S. M. Aghito and S. Forchhammer, "Context based coding of bi-level images enhanced by digital straight line analysis," *IEEE Trans. Image Processing*, to be published.

[4] S. M. Aghito and S. Forchhammer, "Efficient coding of binary shape sequences for object based video," in *Proc. of International Workshop VLBV'05*, Sep. 2005, 4 pages.

[5] L. Piron and M. Kunt, "Differential coding of alpha planes with adaptive quantization," *ITG Fachberichte*, pp. 713-718, 1997.

[6] S. M. Aghito and S. Forchhammer, "Context based coding of quantized alpha planes for video object," in *Proc. MMSP*, Dec. 2002, pp. 101-104.

[7] S. M. Aghito, *Algorithms for Object-Based Video Coding*, PhD Thesis, COM-DTU, Lyngby, Denmark, 2006.

[8] P. J. Ausbeck Jr., "The piecewice-constant image model," *Proc. IEEE*, vol. 88, no. 11, Nov. 2000, pp. 1779-1789.

[9] D. Shkarin, "BMF version 2.0." Available: http://www-lat.compression.ru/ds/bmf_2_hz.rar.