Sprite-Based Video Coding Using On-line Segmentation

Regis Crinon and Ibrahim Sezan

SHARP Laboratories of America
5755 NW Pacific Rim Blvd.
Camas, Washington 98607, USA

ABSTRACT

We address the problem of on-line sprite-based video coding in cases where scene segmentation is not available a priori or transmission of such segmentation information cannot be afforded due to low bit rate requirements. We propose an on-line segmentation method that can be integrated into an MPEG4 on-line sprite based video codec. The proposed method uses macroblock types as well as motion compensated residuals to perform the on-line segmentation. It produces a background mosaic without requiring a priori foreground-background segmentation information. Our results demonstrate the coding efficiency and functionality benefits of the proposed approach.

1. INTRODUCTION

Sprite (mosaic) based video compression refers to building a mosaic image of a particular object within the scene (e.g., background) and using it as reference in predictive coding to increase the resulting compression efficiency. This is an added new dimension to classical video coding where predictive coding is limited to inter-frame coding only. The benefit of sprite-based coding using a background mosaic is realized when a part of the background that has been visible in the past (hence included in the mosaic) is uncovered at a current time instant. In this case, prediction from the mosaic increases the coding efficiency. Both off-line and on-line sprite-based video coding are currently being considered in the MPEG4 standardization [1,2]. In off-line coding, the background mosaic is constructed and transmitted to the decoder prior to coding and then used by directly warping it to form the background. In subsequent steps only warping parameters are transmitted. In on-line sprite-based coding, the background mosaic is generated during transmission using a priori background-foreground segmentation information, both at the encoder and the decoder at the same time. A fundamental difference in on-line mosaic based coding is the fact that predictive coding in reference to the dynamic mosaic is performed rather than direct warping and copying the contents of a static mosaic. Image mosaics were also used in [3] in an MPEG2-like video codec. The current methods in MPEG4 do utilize explicit foreground-background segmentation information that is assumed to be known a priori. This information is transmitted via binary shape mask coding. In this paper, we consider the cases where (i) segmentation information is not available a priori; or (ii) transmission of shape information may consume a significant amount of bits in lower bit rate environments.

We propose an on-line mosaic building technique that is fully integrated to an MPEG4 encoder and decoder, and is also capable of implicitly performing foreground rejection in creating the background mosaic. The advantages of the proposed technique are (i) ability of using sprite-based coding even when a priori segmentation information is not available (i.e., increased versatility); (ii) creating a background mosaic only (without foreground) in the absence of explicit a priori segmentation information (i.e., increased functionality), and (iii) increasing visual image quality in low bit rates by not requiring explicit coding of shape information. Resulting background mosaics created by byproduct of compression can be stored and subsequently used as representative images of the video bitstream. In the proposed codec, the background video object plane (VOP) is assumed to move with the dominant image motion. On-line segmentation of the background and building of the associated mosaic is performed by making use of the dominant motion in the scene, and the macroblock type chosen by the encoder. Each macroblock can be coded ‘intra’, ‘non-intra’ or ‘mosaic’ type, as specified in [1], where the choice by the encoder is made on the basis of motion-compensated prediction error.

In the following, we first describe the proposed method of on-line segmentation in a sprite-based MPEG4 video codec. We then present the simulation results reflecting the compression efficiency and added functionality benefits of the proposed approach.

2. ON-LINE SEGMENTATION

On-line segmentation is achieved by distinguishing the background and foreground pixels in a video frame — a rectangular VOP. The background mosaic is reconstructed using only the background pixels as they become visible. Automatic discrimination between background and foreground pixels is performed on the basis of a macroblock (16x16 luminance pixels) using its motion and coding type, called the “macroblock type” determined by the encoder. We now explain this process in some detail. We define the following quantities. We assume that the current frame is at time $t$ and the decoded version of the previous frame at time $t-1$ is available at both encoder and the decoder.

$(j,k)$: The position of the macroblock in the VOP being encoded. The coordinates $(j,k)$ represent the upper left corner of the macroblock. The size of a macroblock is $B_h \times B_v$ pixels, where $B_h$ is the horizontal dimension and $B_v$ is the vertical dimension of the macroblock, respectively.

MBType $(j,k)$: The macroblock type. This quantity can take the value INTRA, INTER1V (one motion vector for the whole macroblock), INTER4V (four motion vectors for each of the 8x8 blocks in the macroblock) and MOSAIC. The INTER4V and
**Step 1: Initialization of the mosaic:** The content of the mosaic is initialized with the content of the first VOP at time $t = t_0$. The shape map of the mosaic is initialized to 1 over a region corresponding to the rectangular VOP shape. The value 1 indicates that texture content has been loaded at this location in the mosaic.

$$S_s(R, t_0) = \begin{cases} \text{VO}_t(r, t_0) \text{ if } \text{VO}_t(r, t_0) = 1 \\ 0 \quad \text{otherwise} \end{cases}$$

where $S_s$, $S_r$, $\text{VO}_t$, $\text{VO}_r$ represent the mosaic shape map, the mosaic texture, the decoded VOP shape map (rectangular here) and the decoded VOP texture fields, respectively. The mosaic shape $S_s$ and the decoded VOP shape $\text{VO}_t$ are binary alpha maps. In the mosaic shape map, the values 0 and 1 mean that the mosaic content is not determined and determined at this location, respectively. The position vectors $R$ and $r$ represent the pixel position in the mosaic and in the VOP, respectively. These spatial coordinates differ by an offset $\Delta$ as described in [1].

**Step 2: Acquire the next VOP (time $t + 1$) and select macroblock types:** Assuming that the current time is $t + 1$, the mosaic that has been updated last at time $t$ exists. Residuals from previous VOP prediction are obtained by using conventional block matching. The global backward mapping $\mathcal{W}_b$ is used on every macroblock to get residuals resulting from motion prediction. Comparing the residuals, the encoder selects the macroblock type resulting in the least amount of residual or INTRA type when prediction is poor. At this time, the global backward mapping is also used to compute and record the residuals $f(j, k)$ obtained by predicting from the previously decoded VOP.

**Step 3: Encode and decode the VOP:** The encoder encodes and decodes the VOP at time $t + 1$. The decoder decodes the bitstream to generate the same decoded VOP.

**Step 4: Create binary map to detect macroblocks belonging to foreground:** For every macroblock $(j, k)$ in the current rectangular-shaped VOP, build an initial object segmentation map $g(j, k)$, defined as follows, given the macroblock types and coded residuals of macroblocks at time $t + 1$.

```c
/* Test whether macroblock is of type MOSAIC */
if( MBType(j,k) == MOSAIC )
{
    /* Set segmentation map to 0 */
    g(j,k) = 0
}

/* Test whether macroblock is of type INTER4V */
else if( MBType(j,k) == INTER4V )
{
    /* Set segmentation map to 1 */
    g(j,k) = 1
}

/* For macroblock types other than INTER4V and MOSAIC (i.e., INTRA and, INTER1V */
else
{
    /* Compare residual from Global Motion Estimation on previous VOP against macroblock residual */

```
Given a macroblock position \((j,k)\), let \(\ell = \left[ \frac{j + l}{k + p} \right]\) where the variables \(l\) and \(p\) are such that \(0 \leq l \leq B_h - 1\) and \(0 \leq p \leq B_v - 1\). The variables \(j + l\) and \(k + p\) are used to denote the position of each pixel within the macroblock \((j,k)\).

For every value \(l\) and \(p\) in the range specified above, do

/* Test whether pixel belongs to VOP and whether mosaic content is already determined at this location */
if (\(\text{VO}_l(\ell_r,t+1) == 1\) & & \((S_l(\ell_r,t) == 1)\))
{
    /* Test whether macroblock has been classified as foreground macroblock */
    if (\(h(j,k) == 1\))
    {
        /* Warp mosaic forward but do not change its content */
        \(S_l(\ell_r,t+1) = W_f\{ S_l(\ell_r,t) , w\}\)
    }
    else
    {
        /* Warp mosaic forward and update it by blending current VOP content in */
        \(S_l(\ell_r,t+1) = (1 - \alpha) W_f\{ S_l(\ell_r,t) , w\} + \alpha \text{VO}_l(\ell_r,t+1)\)
        /* Set mosaic shape to 1 to signal that content has been determined */
        \(S_l(\ell_r,t+1) = 1\)
    }
}
/* Test whether pixel belongs to VOP and whether mosaic content is undetermined at this location */
else if (\(\text{VO}_l(\ell_r,t+1) == 1\) & & \((S_l(\ell_r,t) == 0)\))
{
    /* Set mosaic content to current VOP content */
    \(S_l(\ell_r,t+1) = \text{VO}_l(\ell_r,t+1)\)
    /* Set mosaic shape to 1 to signal that content has been determined */
    \(S_l(\ell_r,t+1) = 1\)
}

Step 7: Acquire the next VOP; Go to Step 2 and repeat the process till all VOPs are processed.

### 3. RESULTS

All simulations were performed using the MPEG4 test sequence STEFAN, considering the following cases. We have used Momusys Video VM8.0 (Verification Model) MPEG4 software.
Case 1: The object segmentation is known. Sprite-based coding is not invoked.
Case 2: The object segmentation is known. On-line sprite-based coding is used for the background Video Object.
Case 3: The object segmentation is not known. On-line sprite-based coding is used with the proposed on-line segmentation method.

For Cases 2 and 3, the size of the mosaic is set to 720x300 pixels; motion is modeled as affine and estimated using the MPEG4 VM software; and the blending factor is set to unity. In Case 3, the threshold $\theta$ is set to 1.05 and the blending factor $\alpha$ is set to 0.5. The macroblock neighborhood system $\Omega$ is a $3 \times 3$ region ($M=3$) centered about the macroblock of interest. The value of the rank $\rho$ is 6. The PSNR results (for luminance) for cases 2 and 3 are furnished in Table 1. In Case 2, where a priori segmentation map is used with 2 Video Objects, both foreground (fg) and background (bg) performance are specified. In case of on-line segmentation, the PSNR figure applies to the entire video frame.

Table 1: Summary of Results

<table>
<thead>
<tr>
<th>$q$</th>
<th>Given Segmentation</th>
<th>On-Line Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1028 Kbps</td>
<td>1032 Kbps</td>
</tr>
<tr>
<td></td>
<td>29.60 dB (bg)</td>
<td>29.56 dB</td>
</tr>
<tr>
<td></td>
<td>27.44 dB (fg)</td>
<td>27.42 dB</td>
</tr>
<tr>
<td>27</td>
<td>378 Kbps</td>
<td>383 Kbps</td>
</tr>
<tr>
<td></td>
<td>24.35 dB (bg)</td>
<td>24.34 dB</td>
</tr>
<tr>
<td></td>
<td>22.79 dB (fg)</td>
<td>22.78 dB</td>
</tr>
</tbody>
</table>

In Table 1 above, each entry shows the bit rates followed by the Luminance PSNR value obtained for the rectangular frames (cases 3 and 4) or for both the background and the foreground objects, respectively (cases 1 and 2).

The results indicate that coding efficiency provided by on-line sprite-based coding with automatic segmentation is comparable to the one obtained in the case of sprite-based coding applied to the background Video Object (Case 2) at the higher bit rate. On-line sprite-based coding with automatic segmentation becomes clearly favorable at lower bit rates (384 Kbps) when bits spent to code the shape information (i.e., the segmentation map) becomes a relatively larger overhead. At 1028 Kbps, the shape information represents about 5% of the transmitted data and the quantizer level is equal to $q=12$ in both cases. At lower bit rates, the shape information represents about 12.5% of the transmitted data. This relative increase in the amount of shape information penalizes the conventional on-line sprite-based coder and as a result, the quantizer level must be coarser ($q=29$) compared to the on-line segmentation-based coder where no shape information is transmitted ($q=27$). (Larger $q$ implies coarser quantization and hence lower image quality).

We also like to note the benefits of sprite-based coding in general, with or without on-line segmentation. The comparable bit rate (1047 Kbps) was reached with MPEG4 VM without sprites at $q=13$ (compared to $q=12$). At the lower bit rate, the MPEG4 VM without sprites is coded at 404 Kbps at $q=31$ compared to sprite-based and on-line sprite based coding at comparable rates but at $q=29$ and $q=27$, respectively, where on-line segmentation is clearly superior to all.

In sprite-based compression, coding efficiency improvements come from scene re-visitation (when the camera pans back and forth), uncovering of background (when a foreground object covers and uncovers a portion of the background), and global motion estimation (no local motion vectors to transmit).

A sample mosaic generated on-line is shown in Figure 1. Further visual results will be shown during our presentation.

4. SUMMARY

The proposed approach improves sprite-based coding further in terms of coding efficiency as well as functionality. In particular, it (i) improves coding efficiency, especially at low bit rates, by not requiring explicit shape coding; and (ii) does not require prior segmentation. In other words, on-line sprite-based coding with proposed on-line segmentation can be realized “entirely on line”. From functionality point of view, the proposed approach also readily provides a background mosaic as a result of automatic, on-line segmentation. The background mosaic can be used as a representative image of the video sequence in indexing. Indeed, the foreground objects can be superimposed on the background mosaic in any desired fashion to express the scene dynamics.

5. REFERENCES