REAL-TIME RECURSIVE MOTION SEGMENTATION OF VIDEO DATA

Rimmert Wittebrood and Gerard de Haan
Philips Research Laboratories, Eindhoven, The Netherlands

ABSTRACT
Segmentation is a hot topic for video compression and interpolation. We introduce a recursive algorithm that enables real-time motion segmentation of standard definition video on a DSP. The evaluation of the separate optimization steps is included.

1 INTRODUCTION
We reported on object based motion estimation (ME) for software video format conversion in an earlier publication [3]. By far the most critical part of this function is the video segmentation module, which assigns individual motion models to image regions in our estimator pixel blocks. In this paper, we shall report on progress: in Section 2, we summarise our object based ME showing the advantages of recursive segmentation over best match selection. We introduce and evaluate the use of penalties in this recursive algorithm (Section 3), and a complexity reduction technique called “block hopping” (Section 4). Finally, in Section 5, we draw our conclusions.

2 FULL-SEARCH VS. 3DRS
A parameter vector \( \hat{p}_O = (\hat{t}_{xO, y}, \hat{z}_{xO}, \hat{z}_{yO}) \) is associated with every image object \( O \). This parameter vector describes the translation and zooming of the object. The relation between the parameter vector of an object \( O \) and the motion (displacement) vector of the object at position \( \hat{x} = (x, y)^T \) in the image is:

\[
\hat{D}_O(\hat{x}) = \left( \begin{array}{c}
\hat{t}_{xO, y} + \hat{z}_{xO, x} \\
\hat{t}_{yO, y} + \hat{z}_{yO, y}
\end{array} \right)
\]

How well a motion model fits for a given block \( B(\hat{X}) \) can be expressed by the following error measure:

\[
e_O(\hat{X}) = \sum_{x \in B(\hat{t})} \left| F(\hat{x}, x) - F(\hat{x} - \hat{D}_O(\hat{x}), x - 1) \right|
\]

Here \( n \) is the image number of the luminance image \( F \), and \( \hat{x} \) is a pixel in the center of block \( B(\hat{X}) \). Note that \( \hat{X} \) and \( \hat{x} \) are associated with positions on the block- and pixel-grid respectively.

A segmentation is obtained if this error measure is minimized over all motion models:

\[
M(\hat{X}) = \arg\min_O (e_O(\hat{X}))
\]

Here \( M(\hat{X}) \) is the so-called segmentation map. Since this segmentation resembles the vector assignment of a full search (FS) block matcher, similar problems occur. The main problems are the inconsistency of the resulting vector field, and the high computational complexity and required bandwidth. In the past, much research was aimed at reducing the disadvantages of a FS block matcher; surveys are given in [2, 1]. The most successful block matcher (for video format conversion purposes) both in accuracy and consistency is the 3D recursive search (3DRS) block matcher [1]. The approach used in this block matcher is also applicable here. Instead of minimising eq.2 over all motion models, prediction is used to limit the number of motion models evaluated in the minimization.

Fig.1 Comparison between a FS (top) and a 3DRS segmentation (bottom). Eight motion models are used. Different motion models are identified with different grey levels.
sequences as in [1, 3]). On average 8 motion models were defined for every sequence.

Table 1 Quality of segmentation. Results for FS, 3DRS, 3DRS with penalties and 3DRS with penalties and block hopping.

<table>
<thead>
<tr>
<th>Measure</th>
<th>FS</th>
<th>3DRS</th>
<th>3DRS PEN</th>
<th>3DRS BH</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2SE</td>
<td>412</td>
<td>379</td>
<td>372</td>
<td>373</td>
</tr>
<tr>
<td>SI</td>
<td>1.34</td>
<td>0.79</td>
<td>0.67</td>
<td>0.51</td>
</tr>
<tr>
<td>Complexity</td>
<td>8</td>
<td>1.8</td>
<td>1.8</td>
<td>1.4</td>
</tr>
</tbody>
</table>

As well as improving consistency, 3DRS segmentation reduces the computational complexity. Instead of all motion models, only 4 candidates have to be evaluated. Moreover, the increased spatial and temporal consistency illustrate high probabilities of identical spatial and temporal candidates, for which only one $\varepsilon_{O}^{p}(\hat{x})$ has to be calculated. When identical candidates are evaluated only once, the average number of times an $\varepsilon_{O}^{p}(\hat{x})$ had to be calculated for a given block was found to be 1.8 for the 3DRS method, against 8 for the FS method, see Table 1. Table 1 shows that the accuracy, consistency and complexity of the 3DRS segmentation are all better than the FS segmentation.

3 PENALTIES

A further improvement of the consistency of the segmentation results from penalising temporal and random candidates. The following error measure results:

$$\varepsilon_{O}^{p}(\hat{x}) = \varepsilon_{O}(\hat{x}) + S_{O}(\hat{x})$$

(4)

In which penalty $S_{O}(\hat{x})$ is usually zero for spatial candidates, small for temporal candidates and large for random candidates.

Figure 2 shows how the accuracy and consistency of the segmentation depends on the penalty values.

The penalty values considerably improve accuracy and consistency. Table 1 gives results of the segmentation without (3DRS) and with penalties (3DRS PEN) on temporal and random candidates.

4 BLOCK HOPPING

From Figure 3 it can be seen that only a small number of temporal and random candidates have a segmentation error eq.2 smaller than the error $T$ of the spatial candidate.

A considerable reduction in the number of calculations is obtained when the segmentation errors of remaining candidates are not calculated if the spatial candidate has a low enough segmentation error ($<T$). This is shown in Figure 3. Accuracy and consistency with block hopping are still better than with the FS, see Table 1. On average only 1.36 motion models have to be checked per block.

5 CONCLUSIONS

The functional optimizations of the motion segmentation in this presentation result in an accurate and computationally simple segmentation algorithm, applicable in real-time scan rate conversion systems of consumer quality on a DSP [3].

REFERENCES