A Video Segmentation Algorithm For Content-Based Coding

Yong-Wan Kim\textsuperscript{a}, Poong-Min Kim\textsuperscript{a} and Yo-Sung Ho\textsuperscript{b}

\textsuperscript{a}Computer/Software Technology Institute in ETRI
1 Ueun-Dong Yusong-Gu, Taejon, 305-333, Korea

\textsuperscript{b}Kwangju Institute of Science and Technology (K-JIST)
572 Ssangam-Dong Kwangsan-Gu, Kwangju, 506-712, Korea

\section*{ABSTRACT}

This paper proposes a new segmentation algorithm based on morphological filtering and a watershed algorithm. By combining spatial and temporal information, we can get visually meaningful segmentation results. The proposed segmentation algorithm consists of six parts: preprocessing, image simplification, feature extraction, decision, region merging and postprocessing.

Keywords: MPEG-4, Segmentation, Morphology, Watershed Algorithm

\section*{1. INTRODUCTION}

In MPEG-4 video, we extract objects in the video sequence and process them based on their properties for supplying content-based interactivity. Segmentation is an indispensable tool for object-based coding. In object-based algorithms, we segment an image into a set of objects or regions according to its meaningful content, and estimate their parameters such as, color or regions. There have been several segmentation algorithms proposed to detect moving objects.

M. Hötter proposed an interactive approach in which an image is first divided into unchanged and changed areas. Each connected changed region defines an object, and motion parameters describing the motion of each object are computed. Within each object, regions of connected pixels whose motion is not well described by the estimated motion parameters are detected. The algorithm then proceeds recursively by estimating the motion of these objects [1]. G. Adiv proposed a segmentation algorithm based on the optical flow field [2]. However, this algorithm has a huge amount of computational load and this model requires explicit constraint, such as smoothness. P. Salembier proposed a segmentation algorithm using morphological tools, such as morphological filters and a watershed algorithm[3]. However, those methods produce false contours because they use only spatial information or only motion information [4]. Recently, a new method has been proposed to use the information both from the spatial and temporal domain[3,7].

In this paper, we propose an efficient spatio-temporal segmentation algorithm that provides us more accurate contours of objects than conventional schemes. Our proposed segmentation algorithm consists of six parts: preprocessing, image simplification, feature extraction, decision, region merging and postprocessing. We propose a change detection algorithm as a preprocessing, which enables us to prevent the foreground and the background from being merged and to reduce the computational load because we perform the segmentation step only on the foreground. We also propose a postprocessing step to eliminate large contour information parts with small texture information. We use morphological filters for the postprocessing. For these reasons, we propose an efficient spatio-temporal segmentation to get more meaningful and accurate segmentation results.

\section*{2. SPATIO-TEMPORAL SEGMENTATION}

As shown in Figure 1, our proposed segmentation algorithm consists of six parts: change detection, image simplification, feature extraction, boundary decision, region merging and postprocessing.

At first, the change detector differentiates intensity changed regions from unchanged regions. A segmentation algorithm is applied on only the changed regions. Unchanged regions are considered as one object and these are not carried out any further operations. The changed regions are simplified by morphological filtering. A marker extraction

E-mail: ywkim@creator.seri.re.kr

\begin{flushright}
-236-
\end{flushright}
is performed to find seeds, which are coherent in both the motion and luminance domains. Region boundaries are decided by a watershed algorithm that incorporates motion and luminance information simultaneously. With this result, regions having similar motions are merged. Finally, the result after region merging is postprocessed to eliminate redundant parts.

2.1. Change Detection

The conventional segmentation algorithms, which are based only on the spatial information or temporal information, may have false contours around objects. In some cases, the background and foreground can be merged. Recently, several attempts have been made to exploit information both from the spatial and the temporal domains in order to have more meaningful segmentation results. However, such conventional spatio-temporal segmentation algorithms are apt to merge the foreground into the background. In this paper, we propose to use change detection as a preprocessing step for segmentation to avoid such a situation. Unchanged regions are considered as one object of the background and segmentation is applied only to the changed regions.

A block diagram of the change detection algorithm is shown in Figure 2. In the first step, we calculate frame differences of two successive frames, k and k+1. Then, absolute frame differences are added up in a measurement window of size $3 \times 3$ picture elements. Comparing the sum to a predefined threshold $T_{ch}$, the central picture element is assigned either to the changed state ($C_1 = 1$) if the sum exceeds or equals to the threshold $T_{ch}$, or to the unchanged state ($C_1 = 0$) if the result is below $T_{ch}$. In the next step, median filtering is applied in a measurement window of size $5 \times 5$ picture elements. This filtering is employed to smooth the boundaries between the changed and unchanged regions. In the last step of the change detection process, small isolated regions are eliminated below certain threshold.

We adjust the change detection threshold $T_{ch}$ for the next two frames k+1 and k+2, based on the change detection mask and the frame differences from the frames k and k+1. As an initial threshold value, $T_{ch} = 21$ has been chosen experimentally, where 256 is due to the quantization according to 8 bit per sample. Assuming that the luminance pixels of the unchanged regions of the two frames differ due to temporal noises only, the mean squared frame difference in the unchanged regions is a measure of the power of the noises. However, in order to obtain a proper threshold value as a function of the amplitude in the range of $0.0 \leq T_{ch} \leq 255.0$, we can use the standard deviation of the
mean squared frame difference as a new threshold $T_{eh}$. This procedure is repeated every two frames and has shown to converge very fast, i.e., the threshold $T_{eh}$ does not change significantly from the second or third calculation on [1].

With the use of change detection, we can reduce the amount of computational load in a large video format, such as CCIR601, and prevent the merging of the foreground and the background.

### 2.2. Image Simplification

<table>
<thead>
<tr>
<th>Morphological Function</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erosion</td>
<td>$\delta_n(f)(x) = \text{Min}{f(x - y), y \in M_n}$ (2.1)</td>
</tr>
<tr>
<td>Dilation</td>
<td>$\epsilon_n(f)(x) = \text{Max}{f(x + y), y \in M_n}$ (2.2)</td>
</tr>
<tr>
<td>Geodesic dilation of size 1</td>
<td>$\gamma^1(f, r) = \text{Min}{\gamma^1(f, r), r}$ (2.3)</td>
</tr>
<tr>
<td>Geodesic erosion of size 1</td>
<td>$\gamma^1(f, r) = \text{Min}{\gamma^1(f, r), r}$ (2.4)</td>
</tr>
<tr>
<td>Reconstruction By Dilation</td>
<td>$\varphi^{(rec)}(f, r) = \cdots \delta^1(\cdots \delta^1(f, r), r) \cdots, r)$ (2.5)</td>
</tr>
<tr>
<td>Reconstruction By Erosion</td>
<td>$\varphi^{(rec)}(f, r) = \cdots \epsilon^1(\cdots \epsilon^1(f, r), r) \cdots, r)$ (2.6)</td>
</tr>
<tr>
<td>Opening By Reconstruction</td>
<td>$\gamma^{(rec)}(\epsilon_n(f), f)$ (2.7)</td>
</tr>
<tr>
<td>Closing By Reconstruction</td>
<td>$\gamma^{(rec)}(\delta_n(f), f)$ (2.8)</td>
</tr>
<tr>
<td>Open_Close By Reconstruction</td>
<td>$\varphi^{(rec)}(\delta_n(\gamma^{(rec)}(\epsilon_n(f), f)), \gamma^{(rec)}(\epsilon_n(f), f)))$ (2.9)</td>
</tr>
</tbody>
</table>

Table 1. List of Morphological Functions

The objective of image simplification is to make the signal easier to segment. It controls the type and amount of information that is removed from the sequence before marker extraction and decision. Image simplification is generally used to eliminate the noises and to remove parts of the signal which is of no interest for the segmentation process. The most classical simplification tool in signal processing is a linear lowpass filter. However, it is well known that this filter blurs image edges and does not preserve the contour information. For segmentation applications, it is of course of prime importance to preserve the contour information. Finding a simplification tool able to preserve the contour information is a very active field of research. Many nonlinear filters such as median, rank-order and morphological filters have been proposed. However, even if these filters can provide good results for 1D signals, their
performances deteriorate badly for 2D signals. Most of the time, the results are strongly influenced by the choice of the window size of the filter [6].

A large number of morphological tools are listed in Table 1. Those morphological transformations depend on two basic sets of transformations, erosion and dilation. If \( f(x) \) denotes an input signal and \( M_n \) a window (or a flat structuring element) of size \( n \), the erosion and dilation by \( M_n \) are given by Eq.(2.1) and Eq.(2.2).

The second set of erosion and dilation involves geodesic transforms. They are always defined with respect to a reference function \( r \). The geodesic dilation of size one in Eq.(2.3) is defined as the minimum value between the dilation of size one of the original function \( f \) and the reference function \( r \). The geodesic erosion in Eq.(2.4) is defined by duality. Geodesic dilations and erosions of arbitrary size are defined by iterations. For example, the geodesic dilation of infinite size, which is also called reconstruction by dilation, is given by in Eq.(2.5). The most popular filter by reconstruction is the opening by reconstruction of erosion in Eq.(2.7). Of course, by duality, a closing by reconstruction in Eq.(2.8) can be defined.

Finally, Open_Close by reconstruction in Eq.(2.9) is a combination of opening by reconstruction and closing by reconstruction. These filters have a simplification effect on the signal, but preserve the contour information [4].

2.3. Feature Extraction

Markers are binary signals identifying the presence of homogeneous regions, which will be precisely delimited by the decision step. This step aims at finding flat regions (regions of a constant gray level value and regions of a constant motion value). If we use the spatial and temporal information adequately, we can get a more accurate segmentation result. Feature extraction identifies the presence of homogeneous regions from the spatial and temporal information. A spatial marker makes use of the simplified images and a temporal marker makes use of the motion vector field. They identify the interior regions and do not intend to solve the problem of contour localization. The output of the marker extraction step is a set of labeled markers.

The marker extraction consists of labeling the interior of large flat zones. After the simplification for size-oriented segmentation, we can detect the spatial and temporal marker individually. We select as spatial markers the flat regions, whose sizes are greater than a given threshold. Temporal markers are extracted in the same way. The final markers are the common areas of both spatial markers and temporal markers. Figure 3 illustrates the marker extraction operation.

2.4. Boundary Decision

Once the markers have been defined, the decision can be taken by the watershed algorithm. A large number of pixels are not yet assigned to any region. Assigning these pixels to a given region can be viewed as a decision process that precisely defines the partition.

The classical morphological decision tool is the watershed. It is generally used on the morphological gradient of the image for segmentation. However, in the case of moving images, the use of the gradient results in a much larger loss of information. Indeed, in the temporal direction, the thickness of the gradient depends on the motion of the objects and can become very large. Therefore, the use of the gradient should be avoided and the watershed algorithm
Figure 4. Watershed Algorithm with a Hierarchical Queue

has to be modified to work directly on the signal, but not on its gradient [5]. This modified watershed algorithm is a region growing process. The set of markers are extended until they occupy all the available space. During the extension, pixels of the uncertainty areas are assigned to one of markers.

In the paper, the distance is defined as a weighted sum of the absolute difference between the pixel gray level value and the mean gray level value of the pixels belonging to the neighboring region, and the motion difference between the estimated motion vector at a certain pixel and the motion vector generated at the certain pixel by the parametric motion model of the region.

\[
\text{similarity measure} = \alpha \times \text{intensity difference} + (1 - \alpha)k \times \text{motion difference}
\]

where \(k\) is a scaling factor and \(\alpha\) is a weighting factor. We can select \(\alpha\) value using the frame difference, because \(\alpha\) is dependent on the amount of motion difference. If frame difference becomes larger, \(\alpha\) value becomes smaller and vice versa. The watershed algorithm works in two steps: queue initialization and region growing.

(a) The initialization puts in the queue the location of all pixels corresponding to the interior of a region in the labeled marker. These pixels have the highest priority (distance 0) because they certainly belong to their respective regions.

(b) The flooding assigns pixels to regions following a region growing procedure. To constrain the current segmentation to the segmentation obtained in the previous levels, the algorithm checks if the region and the pixel under consideration are compatible, that is, if they belong to the same partition class in the previous segmentation. Two incompatible pixels should not be parts of the same region. The flooding extracts a pixel from the queue. If the pixel does not belong yet to a region, we know that at least one of its compatible neighbors belongs to a region. Therefore, all compatible neighboring regions are examined, the similarity measure between these neighboring regions and the current pixel are assessed, and the pixel is assigned to the region giving the highest certainty. Of course, if there is only one compatible neighboring region, the pixel is directly assigned to it. Note that a new pixel has been assigned to a region, the average gray level value of the region and motion parameter should be updated in order to accurately compute its similarity measure with respect to new pixels.

2.5. Region Merging

During the process of boundary decision, the image can be oversegmented i.e., it may have too many segments. Elimination of redundant regions is necessary for saving bits of coding region shapes and motion parameters. The
region merging process has two important components: the definition of the region similarity measure, and its usage. Several similarity measures have been proposed for spatio-temporal region merging. They differ in the way that they exploit the spatial information and the temporal information given by the motion parameters. The second issue is to use the similarity measure in order to determine whether regions should be merged.

In this paper, we use only the temporal information given from the motion parameters to determine whether the regions should be merged [2]. If the regions created in the boundary decision are consistent with the same affine motion, they should be merged together. Consistency with a motion parameter is detected by computing, using the least squares techniques, optimal parameters and related error values for sets of adjacent components.

2.6. Postprocessing

As shown in Figure 6, the result in region merging may have large contour information regions with small texture information. In the MPEG-4 coding framework, these parts produce a large number of coding bits. In order to encode the objects efficiently, these parts should be eliminated. By using a postprocessing operation, we can save the coding bit of objects to transmit the contour and texture information of objects. These redundant parts can be eliminated by several different techniques. In this paper, we used morphological opening and closing filtering to eliminate these parts reasonably for proposed postprocessing. In the extreme case such as a fishing rod, we do not encode the contour of a fishing rod and only encode the texture information of fishing rod with the background together.

3. EXPERIMENTAL RESULTS

3.1. Segmentation Results with Conventional Schemes

Experiments have been carried out on successive frames of video sequences of QCIF format, "Miss America" and "Carphone". Figure 7 (a) and (b) show a hierarchically structured segmentation algorithm by Hötter [1]. We use the 80th and 83th frames of Miss America sequence. We use the 50th and 53th frames of Carphone. Our algorithm is based on the hypothesis that a region to be segmented is defined by a set of uniform motion and position parameters signified as mapping parameters. Simulation results show the inexact contour information of objects because this algorithm is based on change detection.
3.2. Segmentation Results with the Proposed Scheme

In this section, we present experimental results of our proposed algorithm.

(a) Test Frames

Figure 8(a),(b) and Figure 9(a), (b) show the test frames of "Miss America" and "Carphone".

(b) Simplified Image

Figure 8(c) and Figure 9(c) show the simplified image by morphological filtering. The size of structuring element can be selected arbitrarily. However, a flat $11 \times 11$ structuring element is selected by Salembier's simulation work for the QCIF resolution. In our scheme, the size of structuring element is not important. Because we are targeting for the low bit rate environments, region merging is performed by parameters with large difference errors. As show in Figure 8(c)and Figure 9(c), morphological open_close by reconstruction achieves much better flatness, but at the expense of the contour information.

(c) Boundary Decision

Figure 8(d) and Figure 9(d) show the result of boundary decision process. After joint marker extraction, the watersheds algorithm decides the boundary of spatio-temporal regions. The spatio-temporal scheme gives the accurate region boundaries for moving object.

(d) Region Merging

Figure 8(e) and Figure 9(e) show the result of region merging process. We remove the redundancy by merging the spatio-temporal regions having the same motion into one moving object. This corresponds to motion-based region merging where each object is described by the affine motion model.

(e) Final Results

Figure 8(f) and Figure 8(f) show the final segmentation result by the proposed scheme. We see that our spatio-temporal segmentation gives more accurate result compared to Hötter's algorithm. Hötter's algorithm based on only the temporal information produces sometimes inaccurate shape information.

(f) Comparison of the Various Size of the Structuring Elements

There are simulation results for various size of morphological structuring elements. If the size of structuring element is greater than the structuring size 11 in the QCIF format, significant difference between segmentation results no longer exists. However, the size of the structuring element is much larger, the computational load is much heavier and more interior information is removed. If the size of the structuring element is decreased, the number of segment regions is increased. If we process the segmentation steps, we should select an appropriate size of the structuring element according to the input image format.
(g) Comparison of the Shape Coding of the Segmentation Results

In Table 3 and Table 5, we see the improvement of coding efficiency by postprocessing. We use the chain coding for shape coding of results. Therefore, the number of bits for shape coding is three bits per point for shape coding. However, we can use a differential chain coding to save the coding bits \([8]\). Each chain code contains a start point data, the first chain code, and the subsequent differential chain codes. Since a chain code has a cyclic property, a differential chain code can be expressed in the range form \(-3\) to \(4\) by the following definition:

\[
d = \begin{cases} 
  c_n - c_{n-1} + 8, & \text{if } c_n - c_{n-1} < -3 \\
  c_n - c_{n-1} - 8, & \text{if } c_n - c_{n-1} > 4 \\
  c_n - c_{n-1}, & \text{otherwise}
\end{cases}
\]

where \(d\) is the differential chain code, \(c_n\) is the current chain code, and \(c_{n-1}\) is the previous chain code. A Huffman code is used to encode the differential chain code \(d\). Here, we can ignore the huffman header information. The current chain code \(c_n\) is reconstructed as follows:

\[
c_n = (c_{n-1} + d + 8) \mod 8
\]

There are VLC tables for the differential chain code in Table 2 and Table 4. As shown in the VLC tables, we can reduce the length of the VLC tables. This enables us to save the coding bits for shape coding of segmentation results.

4. CONCLUSIONS

Segmentation is an indispensable tool for content-based video coding. In this paper, we have presented a new segmentation that processes segmentation by combining the spatial information and the motion information. After a brief introduction of several conventional segmentation algorithms, a new morphological spatio-temporal segmentation algorithm has been described. The elementary structure employed in this segmentation algorithm contains six steps. In each step, the related theories as well as efficient implementations have been discussed. Morphological filters have been demonstrated particularly suitable for the simplification operation. A efficient marker extraction method has been found. The modified watershed algorithm has been used for boundary decision. A region merging method has also been introduced as a bottom-up approach.

REFERENCES

<table>
<thead>
<tr>
<th>d</th>
<th>Hötter's Result</th>
<th>After Merging</th>
<th>Final Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-2</td>
<td>1111</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-1</td>
<td>110</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>1110</td>
<td>1110</td>
<td>111</td>
</tr>
<tr>
<td>3</td>
<td>1110</td>
<td>1111</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2. VLC Table for Shape Coding (Miss America)

<table>
<thead>
<tr>
<th></th>
<th>% of Coding Points</th>
<th>% of Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hötter's Result</td>
<td>489</td>
<td>831</td>
</tr>
<tr>
<td>After Merging</td>
<td>423</td>
<td>735</td>
</tr>
<tr>
<td>Final Result</td>
<td>407</td>
<td>617</td>
</tr>
</tbody>
</table>

Table 3. Number of Coding Bits for Shape Coding (Miss America)

<table>
<thead>
<tr>
<th>d</th>
<th>Hötter's Result</th>
<th>After Merging</th>
<th>Final Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-2</td>
<td>10111</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-1</td>
<td>11</td>
<td>100</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>11</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>1010</td>
<td>1010</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>101101</td>
<td>1011</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>101100</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4. VLC Table for Shape Coding (Carphone)

<table>
<thead>
<tr>
<th></th>
<th>% of Coding Points</th>
<th>% of Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hötter's Result</td>
<td>1529</td>
<td>1911</td>
</tr>
<tr>
<td>After Merging</td>
<td>926</td>
<td>1673</td>
</tr>
<tr>
<td>Final Result</td>
<td>851</td>
<td>1374</td>
</tr>
</tbody>
</table>

Table 5. Number of Coding Bits for Shape Coding (Carphone)
Figure 8. Simulation Results (Miss America)

Figure 9. Simulation Results (Carphone)