An FPGA-Based Region-Growing Video Segmentation System with Boundary-Scan-Only LSI Architecture

Takashi Morimoto, Hidekazu Adachi, Kousuke Yamaoka, Kazutoshi Awane, Tetsushi Koide, and Hans Jürgen Mattausch
Research Center for Nanodevices and Systems, Hiroshima University
1-4-2 Kagamiyama, Higashi-Hiroshima-shi, 739-8527, Japan
Telephone: +81(82) 424–6265, Fax: +81(82) 424–3499, Email: \{koide, hjm\}@sxsys.hiroshima-u.ac.jp

Abstract—This paper presents a boundary-scan-only (BSO) video segmentation architecture and its FPGA-based prototype system for 80×60 video images (30 fps). In the proposed BSO architecture, an input image is divided into a number of small image blocks. Then only image blocks, that have boundary pixels in the currently grown region, are processed with a block-sized pixel-parallel processing unit. This enables large-sized video segmentation with the compact processing unit, so that FPGA-based real-time video segmentation can be realized. We have developed an evaluation system for 80×60 real-time image segmentation with a standard FPGA device in 130 nm CMOS technology, and also evaluated its performance with video images. From this FPGA implementation result, QVGA-sized real-time image segmentation is expected to become possible with a state-of-the-art FPGA device in 90nm CMOS technology.

I. INTRODUCTION

Recently, methods for object-oriented image processing such as object recognition, object tracking [1], [2], MPEG-7 [3], and object-based image compression [4], [5] have been proposed. In near future, demands of the object-oriented processing is predicted to be much increased. In all of these methods, image segmentation is an indispensable first step for object extraction (see Fig. 1). Furthermore real-time processing is required if it deals with video signals.

Many image segmentation algorithms have already been proposed [6], [7], [8]. However, most of these algorithms are designed for software implementation and have large complexity. Therefore real-time processing (30 fps) with a general-purpose processor is difficult even with the latest generation high-end processor [9]. To satisfy this real-time requirement, some segmentation hardwares, which use the frame difference between two consecutive frames [10], or the frame difference with a registered background image [11], [12], [13] have been proposed. These approaches are very useful, because it enables moving object extraction with simple calculations. However, these algorithms based on temporal information, such as frame difference, cannot simultaneously detect still objects. Furthermore, they cannot be applied to the case of a moving camera.

To overcome this problem, several video segmentation hardwares based on spatial information of the image have been proposed [14], [15]. A pixel-based fully-parallel video segmentation LSI for QVGA size images, which realizes over 300 fps video segmentation, has been reported [14]. On the other hand, an image-scan video segmentation architecture for video frame rate (30 fps) applications has been proposed [15]. In this architecture, an input image is divided into some small image blocks and each image block is processed sequentially by block-sized parallel processing elements. However the processing time of this regularly image-scan approach from top to bottom of the input image is proportional to the number of pixels and it takes much more time as increasing the input-image size.

In this paper, we newly propose a boundary-scan-only (BSO) video segmentation architecture, which realizes efficient image-scan processing to only image blocks including region-growing boundary pixels. We also verify the proposed architecture with a real-time image segmentation prototype system for images of 80×60 pixels. In Section II, we explain the proposed image-scan video segmentation architecture. Next, boundary-scan-only (BSO) scheme is presented in Section III. Then in Section IV, an FPGA-based video segmentation prototype system with BSO architecture is shown. Finally, we conclude this paper and describe future work in Section V.

![Fig. 1. Video/image segmentation and object-oriented image processing.](image-url)
II. IMAGE-SCAN VIDEO SEGMENTATION ARCHITECTURE

A conceptual diagram of our proposed image-scan video segmentation architecture [15] is shown in Fig. 2. An input image (e.g. $6 \times 10$ pixels) is divided into some small image blocks (e.g. $6 \times 2$ pixels). The pixels in each block are processed in parallel with a small image-segmentation processing array of the block size in sequential scan mode from top to bottom of the input image. Between the processing steps of two blocks the processing results of the finished block are stored in memories and the processing status of next block, which is obtained in the previous scan of the block, is loaded into the processing elements of the array. These storing and loading steps require memories with high access bandwidth and they are realized by using on-chip memories. In general, the block size is variable, therefore we can consider trade-off between processing time and hardware amount for applications.

The image-scan architecture consists of an embedded memory part and a processing part as shown in the block diagram of Fig. 3. The processing part is further subdivided into a weight calculation circuit and an image segmentation unit. The weight calculation circuit calculates connection-weights from the pixel data (e.g. RGB values) of the input image, which stand for the similarity among neighboring pixels. The image segmentation unit is an $m \times 2$ array of image segmentation elements (ISE), each corresponding to a processing element for one pixel. The image segmentation unit executes the scan-mode region-growing by sequential processing of the image blocks. For this purpose the previous cell (pixel) states of ISE and the connection weights are first loaded from memories and after processing the new cell (pixel) states are stored again for the block processing during the next scan. The loading and storing operations interact with the memory part. In the image segmentation unit, a seed pixel (leader cell) for region-growing is selected, then region-growing from this seed pixel is carried out with the ISE-array in parallel. After several scans the current region cannot grow anymore, this region is defined as one segment and a label number, as extracted-region identified number, is assigned to all of its pixels. Then the next seed pixel is searched and used for a new region-growing process. In this way, all pixels are segmented into meaningful regions.

In the memory part (indicated by the dashed-line of Fig. 3) there are five kinds of memories, which are called excitation flag, segmented flag, leader cell flag, connection-weight, and label number memory, respectively. The first 3 memories store the intermediate cell (pixel) states during the region-growing processes and together with the connection-weight memory is connected to the image segmentation unit as shown in Fig. 3. All cell status and connection-weight data for segmentation processing of $m \times 2$ pixels block are stored under same address in each memory, so that they can be accessed easily and simultaneously in one clock cycle.

III. BOUNDARY-SCAN-ONLY SCHEME (BSO)

The above described architecture realizes region-growing video segmentation by step-by-step image-scan processing from top to bottom of the input image. Actually, image blocks which have boundary pixels of the currently grown region only must be scanned, and processing of the other image blocks, that is non-boundary blocks, can be skipped. Thus we exploit this characteristics, and propose boundary-scan-only (BSO) scheme to realize reduction of processing time and power-consumption of read/write memory access.

The proposed BSO architecture can be realized by adding the following two circuits to the original image-scan architecture [15], (1) a region-growing boundary detector, and (2) an address generator of on-chip memories, which realizes addressing to region-growing boundary image blocks. To realize the region-growing detector, 1 bit flag register corresponding to each image block is used for storing the boundary block status. Based on these flag registers, only the addresses of the boundary image blocks are generated and inputted to each on-chip memory. These two circuits are additionally implemented in the main controller (Fig. 3).

We have verified the effectiveness of the proposed BSO architecture by functional simulations with Verilog-HDL. Figure 4 shows the processing times of architectures with BSO and without BSO. For QVGA and VGA sized video segmentation, over 80% reduction of region-growing processing time can be
archived by BSO scheme. This reduction enables larger sized video segmentation with same scale of ISE array.

IV. FPGA-BASED VIDEO SEGMENTATION PROTOTYPE

For the verification of the proposed BSO architecture, we have developed an FPGA-based video segmentation evaluation prototype system with real-time video signal. Two purposes for development of this system are described as follows. (1) The segmentation quality and problems of the proposed architecture can be found by using this system more quickly and the visual inspection is more practical than circuit simulations. (2) Since FPGA device is reconfigurable, we can quickly apply and confirm improvements of the proposed architecture with this system.

Block diagram and picture of developed prototype system are shown in Figs. 5 and 6, respectively. In Fig. 5, blocks surrounded by a dashed line are implemented in three FPGAs.

Data stream of the input video signal is briefly explained in the following. A VGA-sized input image is resized to $80 \times 60$ pixels in the image resize block and stored in a clock-asynchronous external memory. Three external memories are used for storing 3 successive frame data. As shown in Fig. 7, three processings, which are the output of the segmentation result to display, image segmentation, and image input from the video camera, can be pipelined. The functionality of each memory is changed in a cyclic way by the active frame selector: (1) the input frame memory, (2) the processed frame memory, and (3) the segmentation result output memory. Finally, pixel and label data of each pixel are transferred to the image restore block for image resizing to the VGA via the labeled pixel extractor block. By using push-button switches, arbitrary segmented region can be displayed.

Due to the limitation of the number of data pins among FPGA boards, whole circuit blocks in Fig. 5 are partitioned to three FPGAs. Consequently, the image resize, the image restore, and labeled pixel extractor blocks are implemented in EPF10K250, and the push switch controller is realized on
TABLE I
SPECIFICATION OF THE DEVELOPED VIDEO SEGMENTATION PROTOTYPE SYSTEM.

<table>
<thead>
<tr>
<th>Input/Output Format</th>
<th>NTSC Y/C Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Pixels (Image Segmentation)</td>
<td>4,800 pixels</td>
</tr>
<tr>
<td>Number of Pixels (Extracted Region)</td>
<td>60 pixels</td>
</tr>
<tr>
<td>Number of Image Segmentation Elements</td>
<td>160 ISEs</td>
</tr>
<tr>
<td>Clock Frequency</td>
<td>12.27 MHz</td>
</tr>
<tr>
<td>FPGA Devices</td>
<td>EP1S60F1020C7, EPF10K250AGC599-3, EPF10K40RC208-4</td>
</tr>
</tbody>
</table>

TABLE II
FPGA RESOURCE USAGE.

<table>
<thead>
<tr>
<th>FPGA Device</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP1S60F1020C7 (Logic Element, LE)</td>
<td>28,052 / 57,120 (48%)</td>
</tr>
<tr>
<td>EP1S60F1020C7 (On-Chip Memory, bit)</td>
<td>144,256 / 5,215,104 (2%)</td>
</tr>
<tr>
<td>EPF10K250AGC599-3 (Logic Cell, LC)</td>
<td>157 / 12,160 (1%)</td>
</tr>
<tr>
<td>EPF10K40RC208-4 (Logic Cell, LC)</td>
<td>162 / 2,304 (7%)</td>
</tr>
</tbody>
</table>

EPF10K40. All the other blocks are implemented in EP1S60. The specification and the FPGA-resources usage of the prototype system are summarized in Tables I and II, respectively. We can see the usage of logic elements (LEs) and on-chip internal memory of the main FPGA (EP1S60) is about 48% and 2%, respectively. From these implementation results, BSO architecture (80×60 pixels) can be realized with only 0.5% logic increase from 27,929 (non-BSO) to 28,052 LEs. Figure 8 shows segmentation results (30 fps) with this prototype system. From these pictures, we can confirm the correct segmentation behavior. From the estimation of the FPGA resource usage for larger size images, single-chip QVGA-size image segmentation can be possible with the latest generation FPGA device (e.g. EP2S180 [16]).

V. CONCLUSION
A boundary-scan-only (BSO) image-scan video segmentation architecture has been proposed. The realization of BSO is simple but very effective for reduction of processing time and power consumption. The effectiveness of the proposed architecture has been also demonstrated by implementing an FPGA-based prototype system. For QVGA or VGA size images, over 80% processing time reduction of the region-growing process (2 rows scan) can be achieved by applying the BSO scheme.

In principle, the proposed 1-dimensional row-scan approach can be extend to 2-dimensional scan approach. This extension enables more flexible and larger size image processing. The verification of this extension is among of our future research tasks.

ACKNOWLEDGMENTS
Part of this work was supported by the 21st Century COE program “Nanoelectronics for Tera-bit Information Processing,” a Grant-in-Aid for young Scientists (B) (No. 16700184), Ministry of Education, Culture, Sports, Science and Technology, Japanese Government and a Grant-in-Aid for JSPS Fellows, 1650741, 2005.

REFERENCES