Extended Abstracts of the 2002 International Conference on Solid State Devices and Materials, Nagoya, 2002, pp. 242-243

D-3-4

Low-Complexity, Highly-Parallel Color Motion-Picture Segmentation Architecture for Compact Digital CMOS Implementation

Takashi Morimoto, Youmei Harada, Tetsushi Koide, and Hans Jürgen Mattausch

Research Center for Nanodevices and Systems, Hiroshima University, 1-4-2 Kagamiyama, Higashi-Hiroshima,739-8527, Japan
Phone: +81-824-24-6265 Fax: +81-824-22-7185 email: koide@sxsys.hiroshima-u.ac.jp

1. Introduction

Image segmentation is the process by which the original natural image is partitioned into meaningful regions and is an important initial task for higher-level image processing such as object recognition or object tracking. Several image segmentation algorithms have been proposed [1,2]. However, due to their complexity, compact digital VLSI implementation is impossible. For previously proposed analog VLSI approaches [3,4] the scalability to future sub-100nm, low-voltage CMOS technologies is questionable. In this paper, we present a low-complexity digital algorithm, offering high-density VLSI-implementation, comparable for gray-scale and color motion-picture segmentation.

2. Color Motion-Picture Segmentation Architecture

The segmentation algorithm (Fig. 1) uses a region-growing approach, which can be viewed as a simplified digital version of a locally-excitatory globally-inhibitory oscillator network (LEGION) [5]. Color and gray-scale picture segmentation differ only in the expressions of Fig. 2 for the connection-weight calculation between the network cells. The proposed VLSI-implementation architecture (Fig. 3) consists of 4 functional stages for connection-weight calculation, leader-cell determination, image segmentation and segmentation-result restoring, respectively.

In the 1st stage the pixel data, i. e. luminance data ($I(L)_i$) for gray-scale and RGB-data ($I(R)_i$, $I(G)_i$, $I(B)_i$) for color pictures, are used to determine the connection weights W_{ij} between pixels according to the expressions of Fig. 2 for picture columns in parallel. The block diagram of the 1st stage (Fig. 4, color case) shows that always 2 neighboring columns are needed for the calculation process. The calculated connection weights are transferred to the 2nd stage for determining leader pixels ($p_i=1$) and ordinary pixels ($p_i=0$). Leader pixels represent the seed pixels in the subsequent region-growing mechanism and require that the sum of the connection weights with their 8 nearest neighbors is larger than a predetermined threshold. The stages for connection-weight (W_{ij}) and leader/ordinary-pixel (p_i) calculation perform the initialization step in the algorithm of Fig. 1 and transmit W_{ij}, p_i to the cell-network of the image-segmentation stage in a column-pipelined mode.

Each cell of the 3rd stage, the image-segmentation network, represents a pixel of the original picture. In this network, which consist of active cells and connection-weight registers, the self-excitation and excitation steps of the algorithm of Fig. 1 are carried out for all pixels of the picture in parallel. The structures of an active cell, a block of 4 connection-weight-registers and the layout-floorplan for the network are shown in Fig. 5a, 5b and 6, respectively. The active cell consists of decoder, adder/ subtractor, control unit and three 1-bit registers. Fig. 6 also displays how the connection weight registers are effectively shared among neighboring active cells. In each region-growing cycle for a segment the new cell-state is decided by the states of the neighboring cells and the connection-weight registers. The 1-bit register l_i in each active cell is used as a flag for indicating whether this cell is included in the presently grown segment or not. After the growth of a given segment is completed, the segment number is stored in the upper-left connection-weight registers of all cells belonging to this segment. The segmentation process in the cell network finishes, if new segments cannot be grown anymore, i. e. when all leader pixels have been used up. The segmentation result, i.e. pixel/segment-number pairs, is then read-out from the cell network in a column-parallel mode and transmitted to the image-segmentation memory by the final segmentation-restore stage.

3. Performance and CMOS Test-Chip Implementation

The performance of the motion-picture segmentation architecture has been evaluated by applying the software-implemented algorithm of Fig. 1 to many test images. Color/gray-scale segmentation examples for the same image are shown in Fig. 7. For 300 x 300 (90,000) pixels images very short average and worst-case image-segmentation times of 60 μsec and 300 μsec, respectively, were verified even with a low clock frequency of 50MHz. This is below motion-picture-segmentation requirements by a factor 100.

The test-chip of Fig. 8 for the cell-network core was designed in 0.35μm, 3 metal CMOS technology. Decoder and adder/ subtractor of the active-cells, which consume the largest area portion, were designed in full-custom. All other circuits of the cell-network core were generated with a standard-cell library from the high-level VERILOG design. An integration density of 16.1 pixels/mm^2 was thus achieved. We have also estimated the possible pixel density for full-custom high-speed (bit-parallel active cell) and high-density (bit-serial active cell) designs in scaled-down CMOS technologies (Table.1), assuming just 3-metal layers. From this data we expect a one-chip integration of the proposed architecture for 300 x 300 pixel pictures at the 100nm technology node and for 600 x 800 pixel pictures at the 50nm technology node.

4. Conclusions

We proposed a digital algorithm for gray-scale/color image segmentation of real-time video signals and a cell-network-based implementation architecture in conventional CMOS technology. Segmentation of natural gray-scale or color images requires only a small change in the preprocessing circuit for weight calculation. Practical application in fully-integrated motion-picture-segmentation chips is estimated to become possible at the 100nm-technology node.

Acknowledgment

The test-chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo with the collaboration by Rohm Corporation and Toppan Printing Corporation.

References

[1] L. Lucchese et al, Proc. of IEEE 10th Tyrrhenian Workshop on Digital Communication, pp. 110-119 (1998).
[2] J. C. Russ, The Image Processing Handbook, CRC PRESS (1999).
[3] H. Ando, et al., IEICE Trans. Fundamentals, Vol. E83-A, No. 2, pp. 329-336 (2000).
[4] F. Perez, et al., International J. of Comput. Vision, Vol. 12, pp 17-42 (1994).
[5] D. L.Wang, et al., Neural Computation, Vol.9, No.4, pp.805-836 (1997)

[Image Segmentation Algorithm]

1. **Initialization**
   (a) Initialization of global inhibitor. $z(0) = 0$;
   (b) Calculation of the connection weights (Table 1(a)).
   (c) Detection of leader cells
   if $(\sum_{j \in N(i)} W_{ij} > \phi_p)$ then $p_i = 1$; otherwise $p_i = 0$;
   (d) Set all cell to non-excitation. $x_i(0) = 0$;
2. **Self-excitation**
   if (excitable cells $= \emptyset$) then stop; // terminate
   else if ($find\_leader() == i \wedge p_i == 1$) then
   $x_i(t+1) = 1, z(t+1) = 1$; //self-excitation
   go to (3.Excitation)
   else go to (2.Self-excitation) ;
3. **Excitation**
   Setting of global inhibitor.
   $z(t) = \bigvee_{\forall i} z_i(t)$; // logical OR of z_i
   if $(z(t) == 0)$ then
   if $(x_i(t) == 1)$ then
   $x_i(t+1) = 0; z_i(t+1) = 0; p_i = 0$; // inhibition
   go to (2.Self-excitation);
   else if $(x_i(t) == 0 \wedge z_i(t) == 0)$ then
   $S_i(t) = \sum_{j \in N(i)} (W_{ij} \times x_j(t))$;
   if $(S_i(t) > \phi_x)$ then
   $x_i(t+1) = 1; z_i(t+1) = 1$; // excitation
   else $x_i(t+1) = 0; z_i(t+1) = 0$;//non-excitation
   else if $(x_i(t) == 1 \wedge z_i(t) == 1)$ then
   $x_i(t+1) = 1; z_i(t+1) = 0$;
   go to (3.Excitation);

Fig. 1: Detail description of the proposed image segmentation algorithm.

(a) **gray-scale images**

$$W_{ij} = \frac{I(L)_{max}}{1+|I(L)_i - I(L)_j|}, j \in N(i)$$

I_i : luminance of pixel i.

(b) **color images**

$$W(R)_{ij} = \frac{I(R)_{max}}{1+|I(R)_i - I(R)_j|} \text{ (for red)},$$

$$W(G)_{ij} = \frac{I(G)_{max}}{1+|I(G)_i - I(G)_j|} \text{ (for green)},$$

$$W(B)_{ij} = \frac{I(B)_{max}}{1+|I(B)_i - I(B)_j|} \text{ (for blue)},$$

$$W_{ij} = \min\{W(R)_{ij}, W(G)_{ij}, W(B)_{ij}\}.$$

Fig. 2: Connection weight calculation expressions.

Table. 1: Estimated pixel density by full custom design.

Technology node(nm)	integration density (pixel/mm^2)	
	High Speed Architecture (bit parallel)	Compact Architecture (bit serial)
350	22	43
180	81	163
100	263	527
50	1054	2107
35	2150	4300

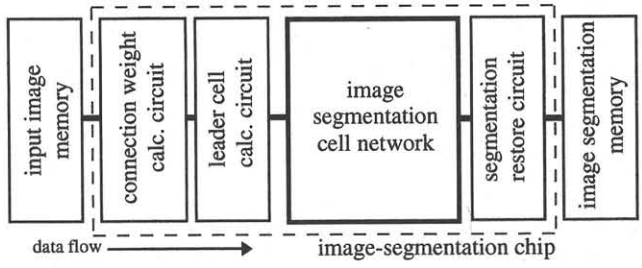


Fig. 3: Block diagram of the cell-network-based image segmentation architecture.

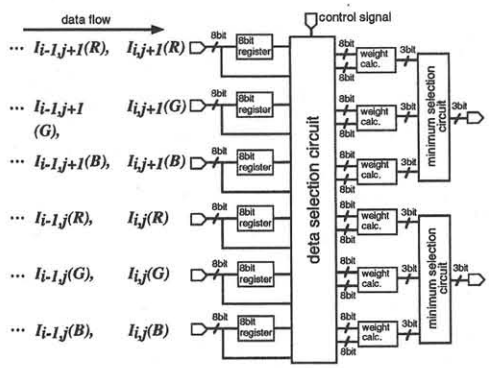
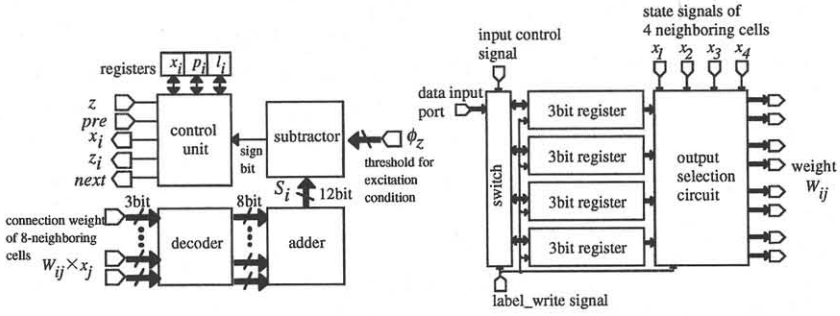


Fig. 4: Structure of connection weight calculation circuit for color image.



(a) active cell P_i (b) block of 4 connection-weight registers

Fig. 5: Structure of active cells and connection-weight-register blocks.

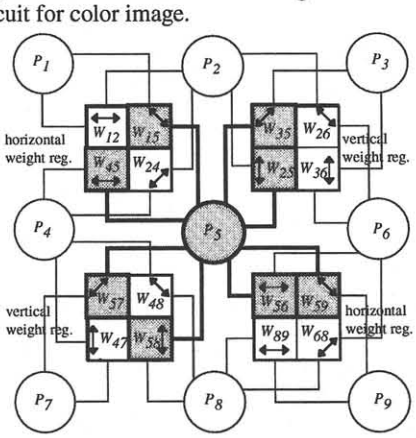


Fig. 6: An example of connections among cell P_5 and its four neighboring connection-weight-register blocks.

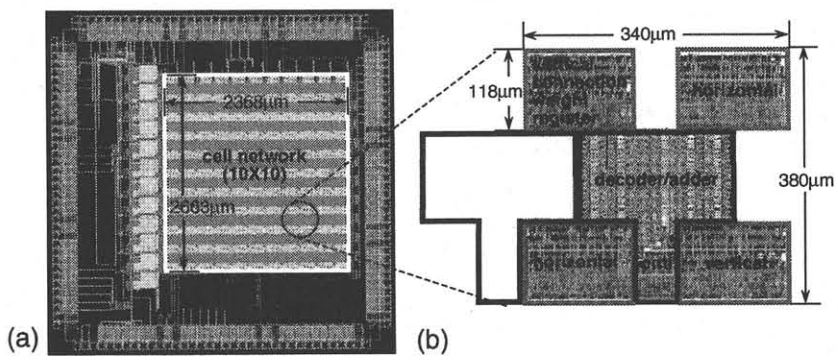
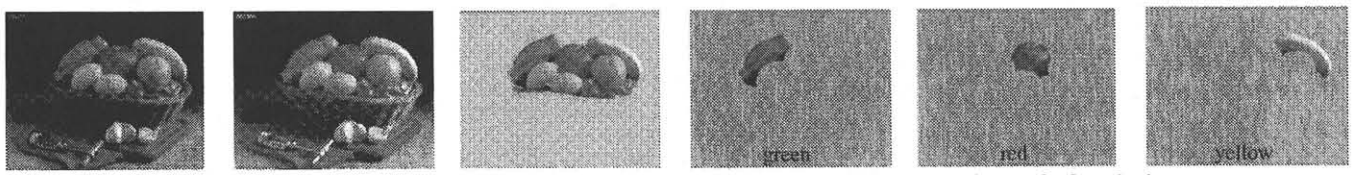


Fig. 8: The layout of the test-chip with a 0.35μm 3 metal layer CMOS technology. (a) cell network includes 10 x 10 cells. (b) The layout of a cell and four neighboring connection-weight-register blocks.



(a) input color image (b) gray-scale image of the input image (c) a segmentation result for gray-scale image (d) segmentation results for color image

Fig. 7: Example of image segmenttion for a color image.