Pixel-parallel digital CMOS implementation of image segmentation by region growing

T. Morimoto, Y. Harada, T. Koide and H.J. Mattausch

Abstract: The paper proposes a real-time implementation architecture of image segmentation by region growing for grey-scale and colour video or still pictures. The proposed digital CMOS implementation realises pixel-based fully-parallel processing with a cell network. To verify the effectiveness of the proposed architecture, a full-custom test chip in 0.35 μm CMOS technology has been designed, containing a cell network for 10×10 pixels with an integration density of 19.6 pixel/mm^2. Measured image segmentation times and power dissipation are ≤ 9.5 μs and ≤ 36.4 mW at the low clock frequency of 10 MHz. From these results, it is estimated that a cell network for about 50 000 ~ 100 000 pixels can be integrated on a single chip in a 90 nm CMOS technology, realising very high-speed segmentation of about 300 μs at 10 MHz for QVGA-size grey-scale and colour images.

1 Introduction

Recently, a variety of applications for object-based visual information processing have been proposed as for example:

- Object-based image compression [1, 2].
- Object recognition and tracking for intelligent transport systems [3] or robot vision.
- Description of multimedia content as spatio-temporal components through segmentation into regions and region-motion tracking by MPEG-7 [4].

For partitioning of images into regions with different contents or objects, image segmentation is indispensable and has to be carried out in real-time when processing video pictures. Several video segmentation algorithms have already been proposed and can roughly be classified into following three groups; (a) temporal [5–8], (b) spatial [9–16], and (c) spatio-temporal [17] segmentations.

Temporal information is very useful, because it enables moving object extraction with simple calculations. Therefore, some popular segmentation algorithms exploit such temporal data as the frame difference between two consecutive frames [5], or the frame difference with a registered background image [6–8]. However, both moving and still objects must be extracted in the case of object recognition, so that spatial information must be included for still object extraction. In recent spatio-temporal segmentation algorithms [17], the input images of a video picture are divided into their constituting regions by spatial and temporal edge information (edgeflow method [13]). The regions that have similar motion information are merged. But the computational complexity of this method is high for the calculation of edge-motion parameters and for edge detection.

Spatial segmentation techniques can be further grouped into four major categories, namely (i) pixel classification [9, 10], (ii) edge based [9–14], (iii) region based [9–11], and (iv) model based [15, 16] approaches. Conventional spatial grey-scale or colour image segmentation algorithms are implemented in software with general-purpose or digital signal processors resulting in relatively large size, large power dissipation, and high cost for the application hardware. In addition, real-time processing is only possible for images with sufficiently small pixel numbers, because of algorithm complexity. For binary images, several types of fast labelling hardware have been proposed [18–20], but these hardware oriented algorithms cannot be extended easily to the segmentation of grey-scale and colour images.

Pixel classification methods have the disadvantage that sometimes isolated points or small noise-like pixel groups are classified as objects. The threshold for classification also depends strongly on the image contrast. Edge-based approaches such as sobel filtering [9–11, 14] need two steps for region extraction, edge detection and identification of regions surrounded by the edges. In addition the edge-based methods are very sensitive to noise. The recent snake model [15, 16] is one example for a model-based segmentation approach. A user defined initial deformed contour adjusts itself to the target object shape by exploiting so-called internal and external energies of the contour. The number of extractable objects is the same as the number of user defined contours, so that unspecific object extraction with respect to object number and structure is difficult.

Region growing [10, 11] is a well-known example for region-based methods suitable for coherent region extraction. A representative example of a cell network-based region-growing algorithm is called locally excitatory, globally inhibitory oscillator networks (LEGION) [21], and uses differential equations for describing the interaction between network cells. Such cell network-based algorithms are very attractive for VLSI implementation, because the highly parallel computation capability of a hardware cell network can be expected to easily reach real-time processing capability. Therefore, the LEGION algorithm has been

IEE Proceedings online no. 20045062
doi:10.1049/ip-cds:20045062
Paper first received 30th June 2004 and in final revised form 3rd March 2005

The authors are with Research Center for Nanodevices and Systems, Hiroshima University, 1-4-2 Kagamiyama, Higashi-Hiroshima 739-8527, Japan
E-mail: morimoto@sxsys.hiroshima-u.ac.jp

modified previously for analogue VLSI implementation [22, 23]. The analogue VLSI implementation of a cell network-based algorithm like LEGION was found to have following advantages: (a) toughness against noise; (b) short segmentation time; and (c) pixel-based fully parallel processing. However, analogue implementation cannot maintaining high reliability with scaled-down technology generations, so that a digital implementation becomes necessary. Therefore, the algorithm requirements for our target segmentation VLSI, which go beyond LEGION, are (i) compact digital integration possibility (low-complexity, highly reliable hardware), (ii) high segmentation quality with grey-scale and colour images, and (iii) suitability for real-time processing of high resolution images (possibility of highly parallel computation). To satisfy these requirements, we have developed a new digital cell network-based segmentation algorithm [24, 25]. The developed algorithm, in particular, enables grey-scale or colour image segmentation by only changing a preprocessing step for calculating connection weights between the network cells.

In this paper, we mainly present the cell network-based VLSI architecture of our algorithm and the verification of its effectiveness by a test-chip design, including a 10×10 cell network, in a 0.35 μm 3-metal layer CMOS technology.

2 Region growing image segmentation algorithm for easy digital implementation

To facilitate the understanding of the developed VLSI implementation architecture, an overview of the applied segmentation algorithm by region growing [24, 25] is given first. An important point of this algorithm is the small number of only four allowed states for each cell (corresponding to a pixel) of the segmentation network, namely self-excitable (leader pixel), not excited but excitable (not labelled), excited, and inhibited (labelled). The flowchart of our algorithm is shown in Fig. 1. It uses six functional steps: (a) initialisation, (b) detection of a self-excitable pixel (leader pixel), (c) self-excitation of the leader pixel, (d) detection of excitable dependent pixels, (e) excitation of dependent pixels (region-growing), and (f) inhibition of all excited pixels (segment pixels) after finishing of each region-growing process. Steps (d), (e) and (b)–(f) are carried out in inner and outer loops, respectively.

In the following, Fig. 2 is used to explain the segmentation process with this algorithm for a simple image example having 3×3 pixels and two regions. In the initialisation phase, connection weights $W_{ij;kl}$ between pixels are

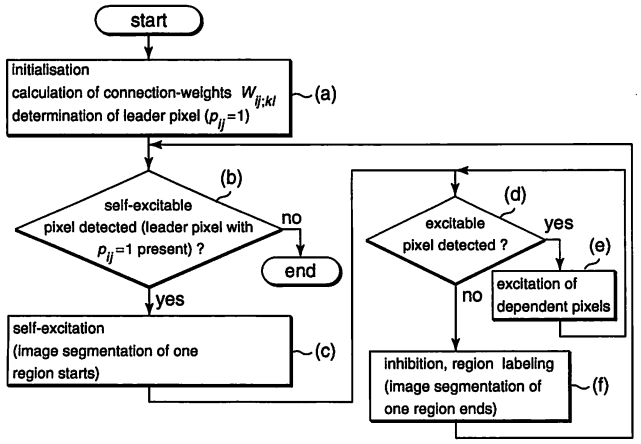

Fig. 1 *Flowchart of the proposed image segmentation algorithm by region growing*

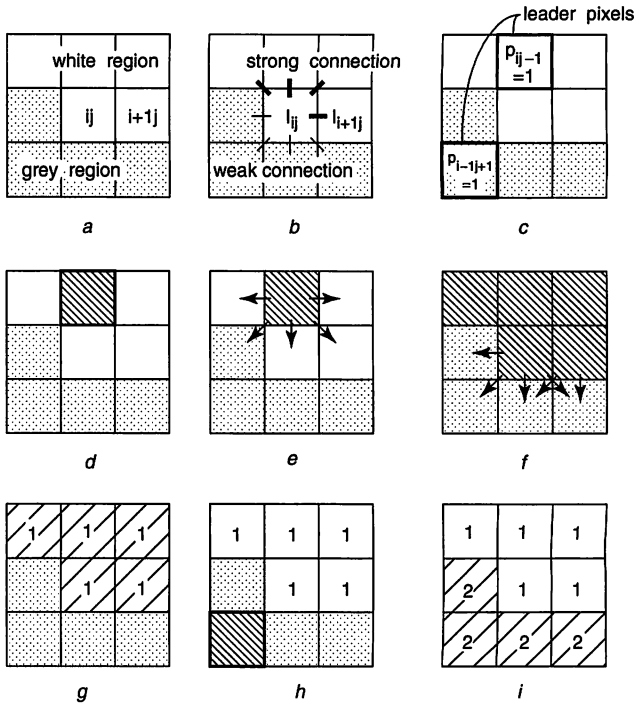

Fig. 2 *Explanation of our segmentation algorithm with a 3×3 grey-scale example image*
a Input image
b Calculation of connection-weight $W_{ij;kl}$
c Determinations of leader pixels
d Self-excitation of leader pixel
e Excitation of dependent pixels
f Excitation of dependent pixels (continued)
g Inhibition (segmentation of white region ends)
h Self-excitation next leader pixel
i Inhibition (segmentation of grey region ends)

calculated from the luminance data (RGB data for colour images) differences between neighbouring pixels (Fig. 2*b*).

In the case of a colour image, the partial connection weight $W_{ij;kl}(R)$ for red (R) colour between pixel (i, j) and pixel (k, l) is defined by

$$W_{ij;kl}(R) = \frac{I_{max}(R)}{1 + |I_{ij}(R) - I_{kl}(R)|} \quad (1)$$

where $I_{max}(R)$ is the maximum value of the component data for red colour. The first indices i and k in each pair ij and kl refer to the pixel column and the second indices j and l refer to the pixel row. Similarly, partial connection weights $W_{ij;kl}(G)$, $W_{ij;kl}(B)$ for green (G) and blue (B) are defined. With these partial connection weights the connection weights $W_{ij;kl}$ between pixel (i, j) and neighbouring pixels (k, l) for colour images are defined by

$$W_{ij;kl} = \min\{W_{ij;kl}(R), W_{ij;kl}(G), W_{ij;kl}(B)\} \quad (2)$$

In the case of a grey-scale image, only (1) with the luminance data is used to determine $W_{ij;kl}$.

The determination of the connection weights is the only difference between the handling of colour and grey-scale images. The initialisation phase then finishes with the determination of the leader pixels (leader pixel state variable $p_{ij} = 1$, otherwise $p_{ij} = 0$), which are the seeds of the subsequent region growing processes. A pixel is defined to be a leader pixel, if the sum of its connection weights with neighbours is larger than a threshold value ϕ_p (Fig. 2*c*). An important advantage of the leader pixel concept is noise suppression, because it prevents noise-like pixels from becoming separate segments. Another advantage in

comparison to pixel classification methods is the robustness against contrast variations, because the connection weights require only the difference information of neighbouring pixels.

In the main segmentation phase, one of the leader pixels is first selected with a token-passing search, scanning the image in a systematic way, e.g. row by row. Then the selected leader pixel (the first leader pixel found) is self-excited (Fig. 2*d*) and the excitation state of the pixel is denoted by setting its excitation state variable $x_{ij} = 1$. The subsequent region growing process is performed in growing steps. In each step, excitable pixels are determined with a threshold condition f_z for the sum of connection weights with already excited neighbours ($x_{kl} = 1$). These excitable pixels are then automatically excited in parallel (Fig. 2*e*), which leads to a growth step of the present region. The growing steps are repeated as long as excitable pixels exist (Fig. 2*f*). If there is no further excitable pixel left, the region of excited pixels ($x_{ij} = 1$) is labelled as one segment and inhibited ($x_{ij} = 0$) (Fig. 2*g*). The labelling status of each pixel is recorded in a flag l_{ij} (0 = not labelled, 1 = labelled). The above operations of leader pixel search (continuing from the leader pixel of the previous segment), region growth by dependent pixel excitation and inhibition of all region pixels after finishing of the growth process are repeated until all leader pixels are inhibited (Fig. 2*h*, *i*). With a systematic token-passing search this means that all pixels of the image have been scanned. The final remaining unlabelled pixels are non-leader pixels corresponding to noise pixels or very small segments without leader pixels. A possible treatment of these unlabelled pixels is to merge them in a post-processing step with the most similar neighbouring segment with respect to luminance or colour.

We have written a software simulator of the proposed algorithm in C language, and tested the segmentation of many natural image samples on a Pentium 4 (1.3 GHz) processor. The average processing time of our relatively simple algorithm for 300 × 300 pixel images was still about 10 s. Since real-time applications require about three orders of magnitude shorter segmentation times, a specialised cell network-based hardware implementation becomes necessary. In the following, we use the word 'cell' to refer to a pixel, in addition to the corresponding hardware cell of the segmentation network.

3 Cell network-based pixel-parallel architecture for VLSI implementation

The proposed implementation of our segmentation algorithm as a digital circuit is shown in Fig. 3 and consists of four functional parts: (1) connection-weight calculation circuit, (2) leader-cell selection circuit, (3) image segmentation cell network, and (4) segmentation restore circuit. In the connection-weight calculation circuit, the connection-weights $W_{ij;kl}$ are calculated from the pixel data (luminance or RGB) of the input image memory. Then the leader-cell selection circuit determines the set of leader cells, namely cells with $p_{ij} = 1$, based on the calculated connection-weights. These first two circuits are realised as a column-processing pipeline, which calculates $W_{ij;kl}$ and p_{ij} for pixel columns in parallel and transmits these data to the cell network also for columns in parallel. The cell network, which is the core of the proposed architecture, performs the image segmentation by parallel execution of the algorithm described in Section 2 for all pixels. Finally, segmentation results are written into an image segmentation memory through the segmentation restore circuit, again in column-parallel mode.

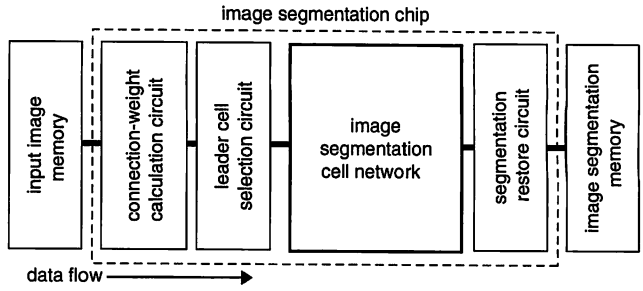

Fig. 3 *Block diagram of the cell network-based VLSI architecture for image segmentation*

3.1 Image segmentation cell network

We explain the image segmentation cell network first, because it is the core of our VLSI architecture and because its structure determines also the construction details of the connection-weight calculation circuit, the leader-cell selection circuit and the segmentation-restore circuit.

3.1.1 Overview of cell network structure and operation: The cell network performs a pixel-parallel processing of the segmentation algorithm for the complete image and consists of cells P_{ij}, which represent corresponding pixels (ith column, jth row), and of vertical (V) and horizontal (H) connection-weight-register blocks $WR_{ij}(V)$, $WR_{i+1,j}(H)$, which are placed between cells alternately. Each weight-register block contains four registers for 3 bit-encoded connection weights and stores connection weights between the adjacent pixels in diagonal and horizontal or vertical directions, respectively. Figure 4*a* illustrates the structure of the cell network with a 3 × 3 pixel example. Each cell is a processing element and determines its state from the sum of connection-weights with excited neighbouring cells. The connection-weight-register blocks are used for weight storage and weight supply to neighbouring cells. The application of horizontal and vertical weight-register blocks has the three VLSI-implementation advantages of area efficiency, minimum wire length, and high layout regularity. Furthermore, the eight connection-weights of each cell with neighbour cells can be shared completely between adjacent cells and redundant storage can be avoided, as illustrated in Fig. 5. Therefore, the horizontal and vertical weight register-block concept is effective to reduce the circuit area for weight storage to about 1/2 in comparison to a straight-forward implementation, and the arrangement in the centre between adjacent cells enables minimum interconnection-wire length. The detailed interconnection of vertical and horizontal weight-register blocks with neighbouring cells is depicted in Fig. 4*b* and Fig. 5*a*, respectively.

3.1.2 Seed-cell determination of segments and segment growth: All cells are linked together with the ports pre_{ij} and $next_{ij}$ for realising a sequential leader cell (self-excitable cell acting as seed of the region growing) search as indicated in Fig. 4*c*. In a synchronous implementation, this search would take as many clock cycles as there are cells to traverse until a new leader cell is reached. To improve this situation, a clock-asynchronous token-passing approach is used. The starting point of the implemented sequential search is at the upper-left cell of the network, where the input signal $start = 1$ is applied. If cell P_{ij} is not a leader cell, that is if it has the leader-cell state variable $p_{ij} = 0$, then the corresponding cell passes the token through to the next cell by setting its corresponding token-flag

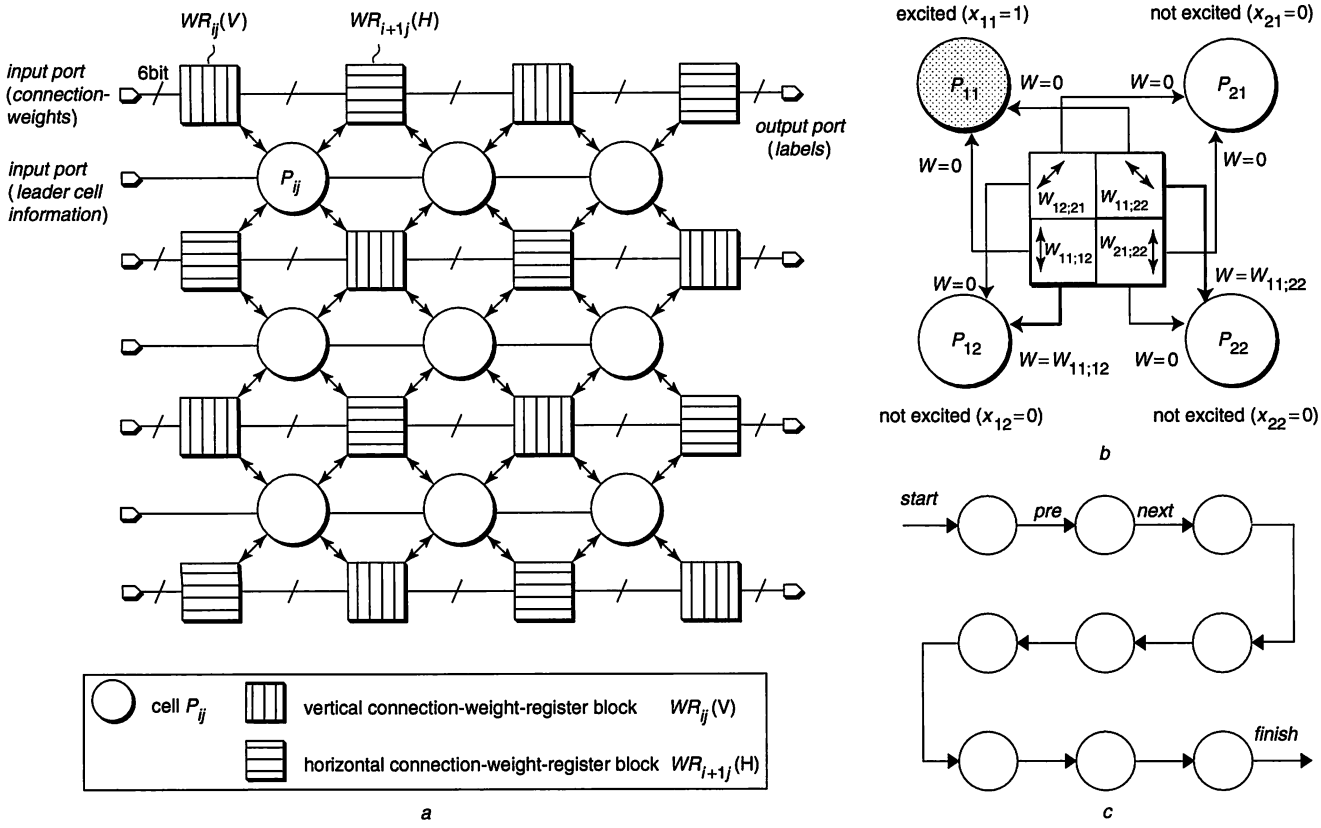

Fig. 4 *Structure diagram of the cell network*
a Block diagram of the cell network construction for 3 × 3 pixels
b Block diagram of four cells with a vertical connection-weight-register block for explanation of cell-state transition
c Sequential cell connection for leader cell (seed of the region growing) search

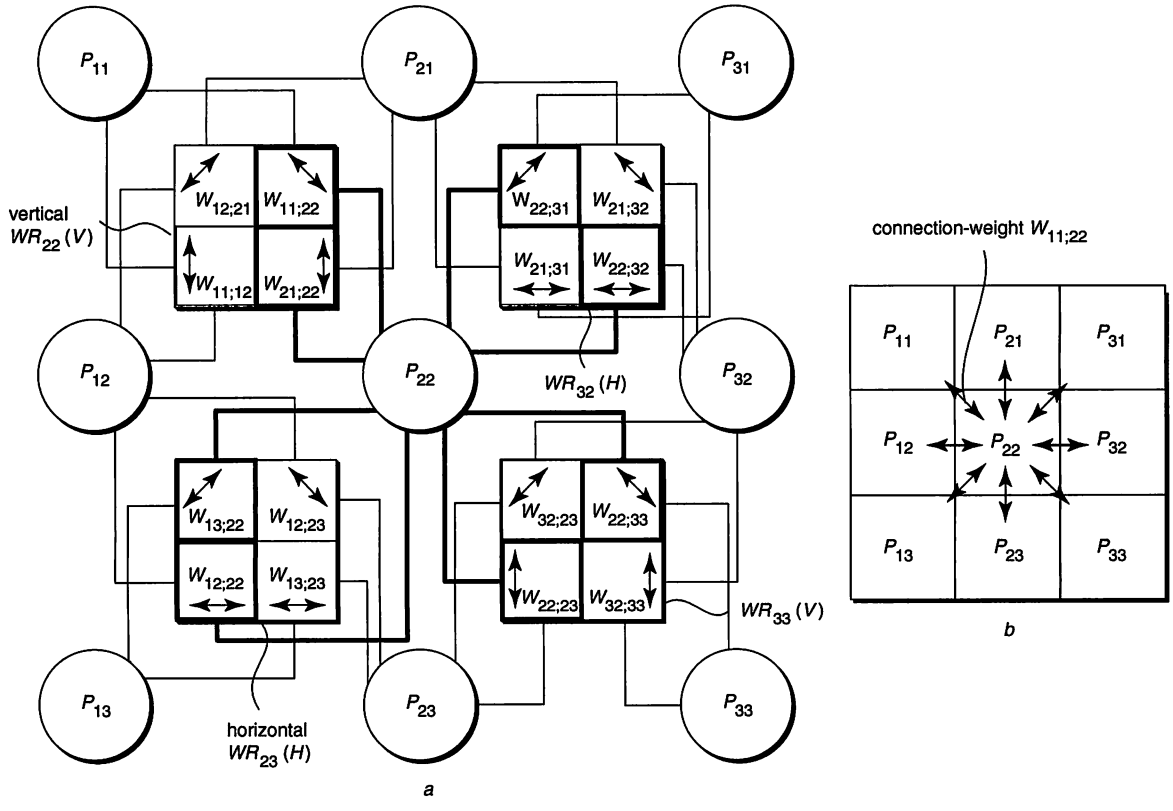

Fig. 5 *An example of connections between cell P_{22} and its four neighbouring connection-weight-register blocks*
a Cell network structure with interconnections to weight-register blocks around P_{22}
b Schematic representation of the connection structure

register $n_{ij} = 0$ (see also the cell structure diagram of Fig. 6*a*) and its token output signal $next_{ij} = 1$. If the cell P_{ij}, which receives the token (change of $pre_{ij} = 0$ to $pre_{ij} = 1$), is a leader cell and thus satisfies the condition $p_{ij} = 1 \& pre_{ij} = 1$, then this cell becomes the seed cell of the new segment-growing process by setting $n_{ij} = 1$ and $next_{ij} = 0$. Afterwards this seed cell is synchronised with the clock and segment growing starts. When the segment-growing process has finished, the inhibition process of the corresponding segment is carried out, during which n_{ij} and $next_{ij}$ of the segment's seed cell are reset to $n_{ij} = 0$ and $next_{ij} = 1$. The sequential search for a new leader cell is then started again at the seed cell of the inhibited segment and carried out as described above. Segmentation of the whole image is completed, when the last cell of the sequential search path is reached and when this cell sets its next-output signal to 1.

The basic function of the cell-state transition during the growth process is explained with Fig. 4*b*, consisting of one excited cell (P_{11}), three not-excited cells (P_{21}, P_{12}, P_{22}) and a vertical connection-weight-register block WR(V). The state signal ($x_{11} = 1$) of the excited cell P_{11} is transfered to its four neighbouring WRs, of which only one is displayed in Fig. 4*b*. In response to the state signal, the shown WR(V) transfers the stored weights $W_{11;12}$, $W_{11;22}$ to the not-excited neighbouring cells P_{12}, P_{22} of the excited cell P_{11}. The not-excited cell P_{21} receives the weight $W_{11;21}$ from the horizontal connection-weight-register block WR(H) in the row above (not shown). Each cell calculates the sum of

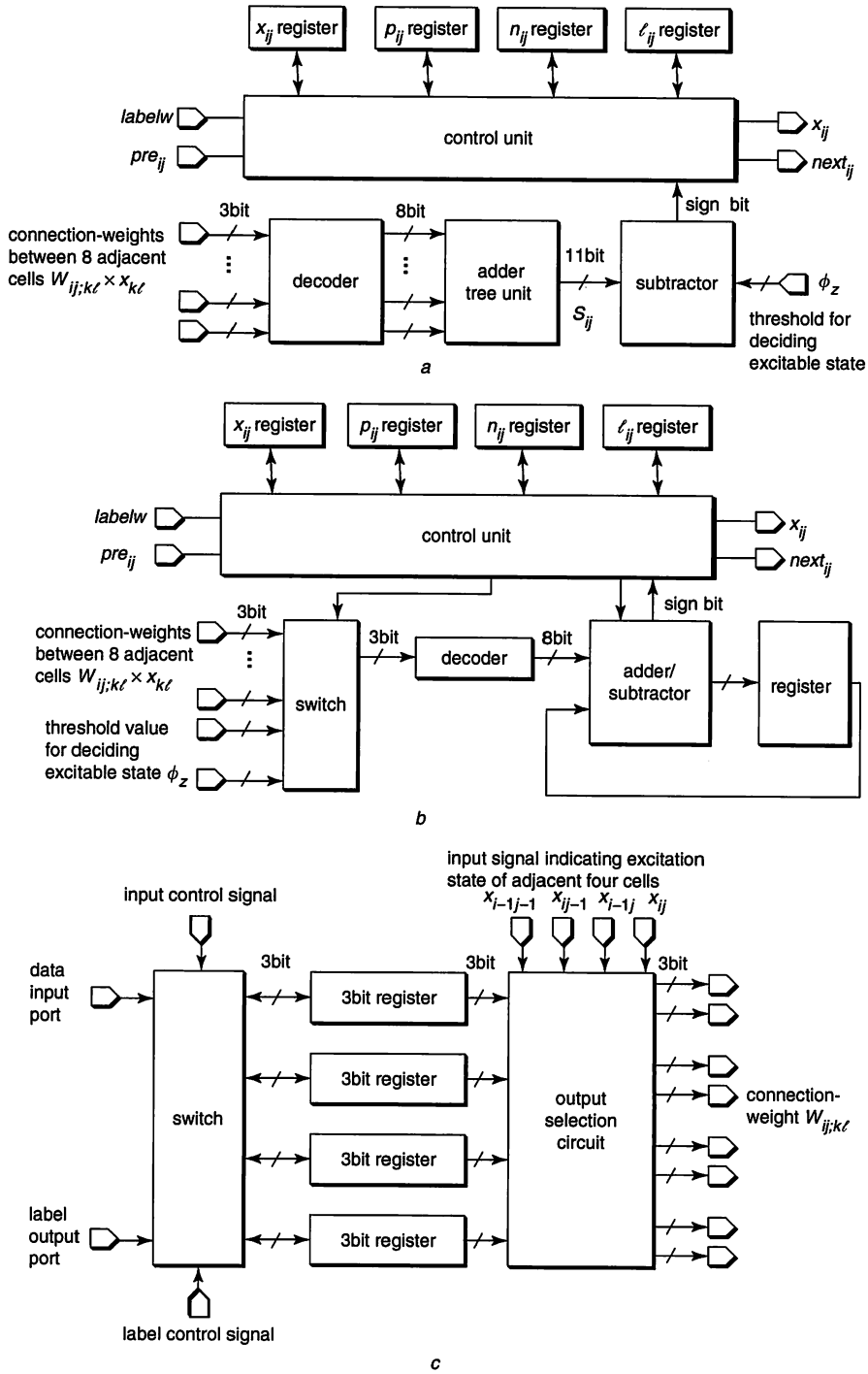

Fig. 6 *Two structure-diagram possibilities for a network cell*
a Weight-parallel architecture
b Weight-serial architecture
c Connection-weight-register block

the transferred connection-weights from up to eight excited neighbours in each clock-cycle, and determines its state based on the calculated sum.

3.1.3 Construction of cells and connection-weight-register blocks:

The summation of connection weights with neighbouring cells can be carried out with a parallel or serial (reduced adder hardware) approach. The two realisation methods have been investigated, and possible cell-implementation examples for weight-parallel and weight-serial processing are shown in Fig. 6. In both examples each network cell consists of decoders, adders/subtractors, a control unit, and four 1 bit registers. The 3 bit-encoded connection-weights are decoded into 8 bit data, and the cell state is determined by the sum of the connection-weights with the excited neighbours. The binary registers x_{ij}, p_{ij}, l_{ij} store the corresponding variables described in Section 2, and register n_{ij} is used for sequential leader-cell search. The signal x_{ij} is connected to the adjacent connection-weight-register blocks $WR_{ij}(V)$, $WR_{i+1j}(H)$, $WR_{ij+1}(H)$, $WR_{i+1j+1}(V)$ and the label number, described in Section 3.4, is controlled by the *labelw* signal. The label (segment) number is stored in the upper-left connection-weight-register block ($WR_{ij}(V)$ or $WR_{ij}(H)$) adjacent to cell P_{ij} to reduce the cell area and to reuse the connection-weight-register blocks. Figure 6*a* is the structure diagram of the weight-parallel architecture. This architecture calculates the sum of connection-weights of neighbouring cells with eight decoders and a three stage adder tree unit in one clock cycle. Alternatively, the weight-serial architecture of Fig. 6*b* calculates this sum in nine clock cycles, but the circuit area is smaller than for the weight-parallel architecture because of sequential processing with only one adder, one decoder and one register. Since the cell structure is simple and compact, high speed and high density implementation can be achieved in both cases.

A connection-weight-register block consists of four 3 bit registers, an output selection circuit, and a switch, as shown in Fig. 6*c*. The basic structure of horizontal and vertical register block is the same. Only the interconnections to the adjacent cells are different. The output selection circuit transfers each connection-weight to the corresponding neighbouring cells according to the status of these neighbouring cells, as indicated in the example of Fig. 4*b*.

3.2 Connection-weight calculation circuit

Figure 7 shows the block diagram of the connection-weight calculation circuit. The concept enabling the pipeline processing is depicted in Fig. 7*a* for four consecutive rows. The input-data flow of luminance (or RGB) values is indicated on the left side of Fig. 7*a* and is realised with delay registers and selectors (not shown) from pixel-column data in parallel. The corresponding output-data flow of connection weights is indicated on the right side. Four connection-weights $W_{ij;kl}$ per column pixel are actually calculated in each input-clock cycle so that selectors and weight-calculation circuits have to be operated at twice the input-clock frequency. The block diagrams of the weight calculation circuits for grey-scale and colour images (three times increased bit number) are shown in Figs. 7*b* and *c*, respectively. The absolute difference calculation circuit determines $|I_{ij} - I_{kl}|$. To avoid the division necessary in

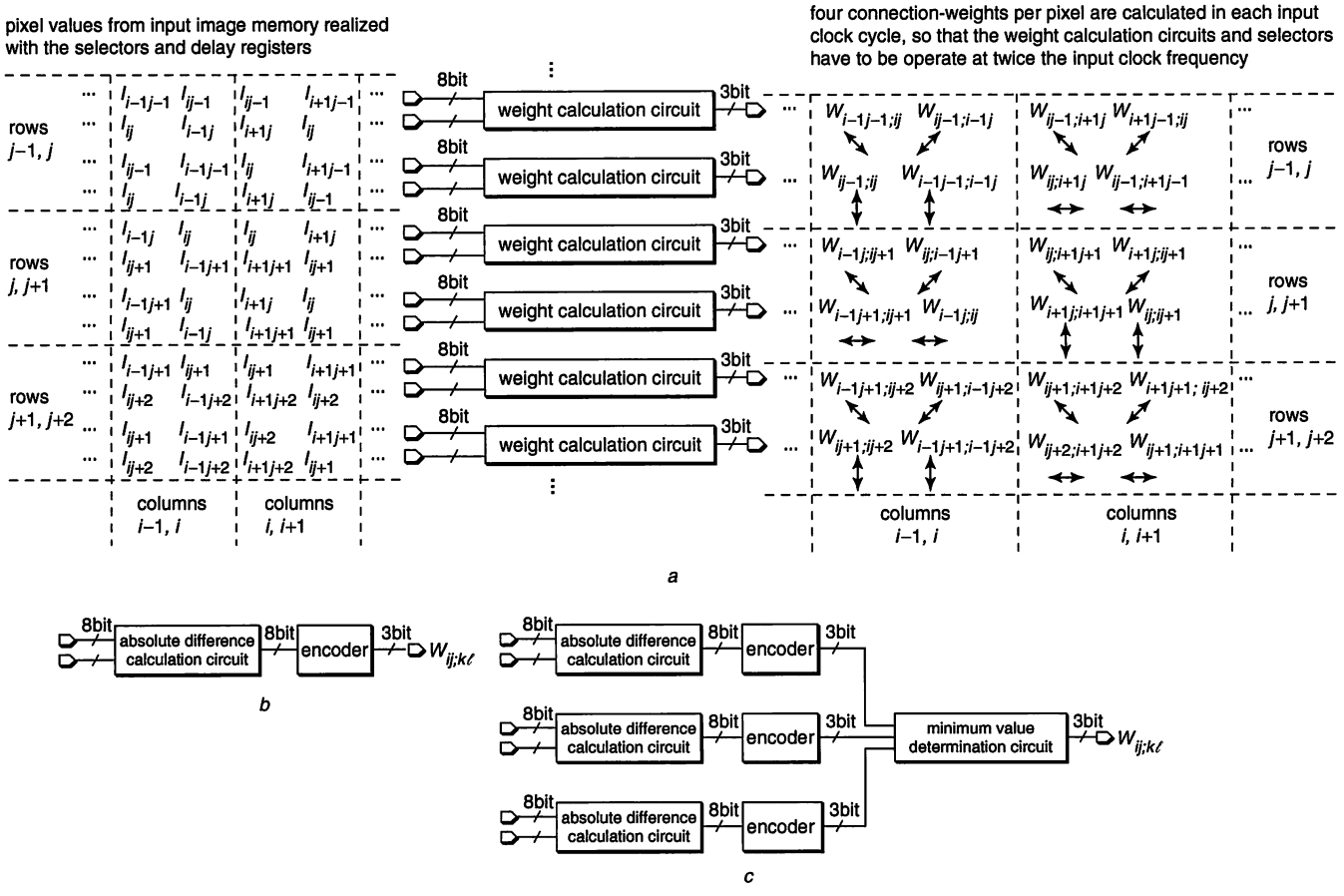

Fig. 7 *Block diagram of the connection-weight calculation circuit*
a Data flow diagram
b Weight calculation unit for grey-scale image
c Weight calculation unit for colour image

the direct calculation of $W_{ij;kl} = \frac{I_{max}}{1+|I_{ij}-I_{kl}|}$, a hardware encoder is used. The encoder applies a look-up table for determining $W_{ij;kl}$ from the 8-bit data of $|I_{ij} - I_{kl}|$ in a 3 bit representation. The error of the 3 bit representation was verified by simulations to be sufficiently small and to have only negligible influence on the segmentation results. For colour images, RGB components are processed according to the same scheme in parallel. In this case a final minimum value determination circuit for calculation of $W_{ij;kl} = \min\{W_{ij;kl}(R), W_{ij;kl}(G), W_{ij;kl}(B)\}$ is used.

3.3 Leader cell selection circuit

Figure 8*a* shows the input data flow of connection weights to the leader cell selection circuit for three consecutive rows. To generate this flow, connection-weights, calculated in the connection-weight calculation circuit, are intermediately stored in registers (not shown), and the corresponding connection-weights for column-wise leader-cell determination are selected with selector circuits (not shown). The leader-cell state variables p_{ij} for each pixel are determined from the connection weights to its eight nearest neighbours. These connection weights are supplied in groups of four at twice the output-clock frequency of the leader cell calculation units, which have the detailed structure shown in Fig. 8*b*. All leader-cell state variables p_{ij} of one column are determined in every output-clock cycle by these leader cell calculation units. If the sum of the connection-weights with neighbouring pixels is bigger than the pre-defined threshold f_p, the leader-cell-calculation unit defines this state variable

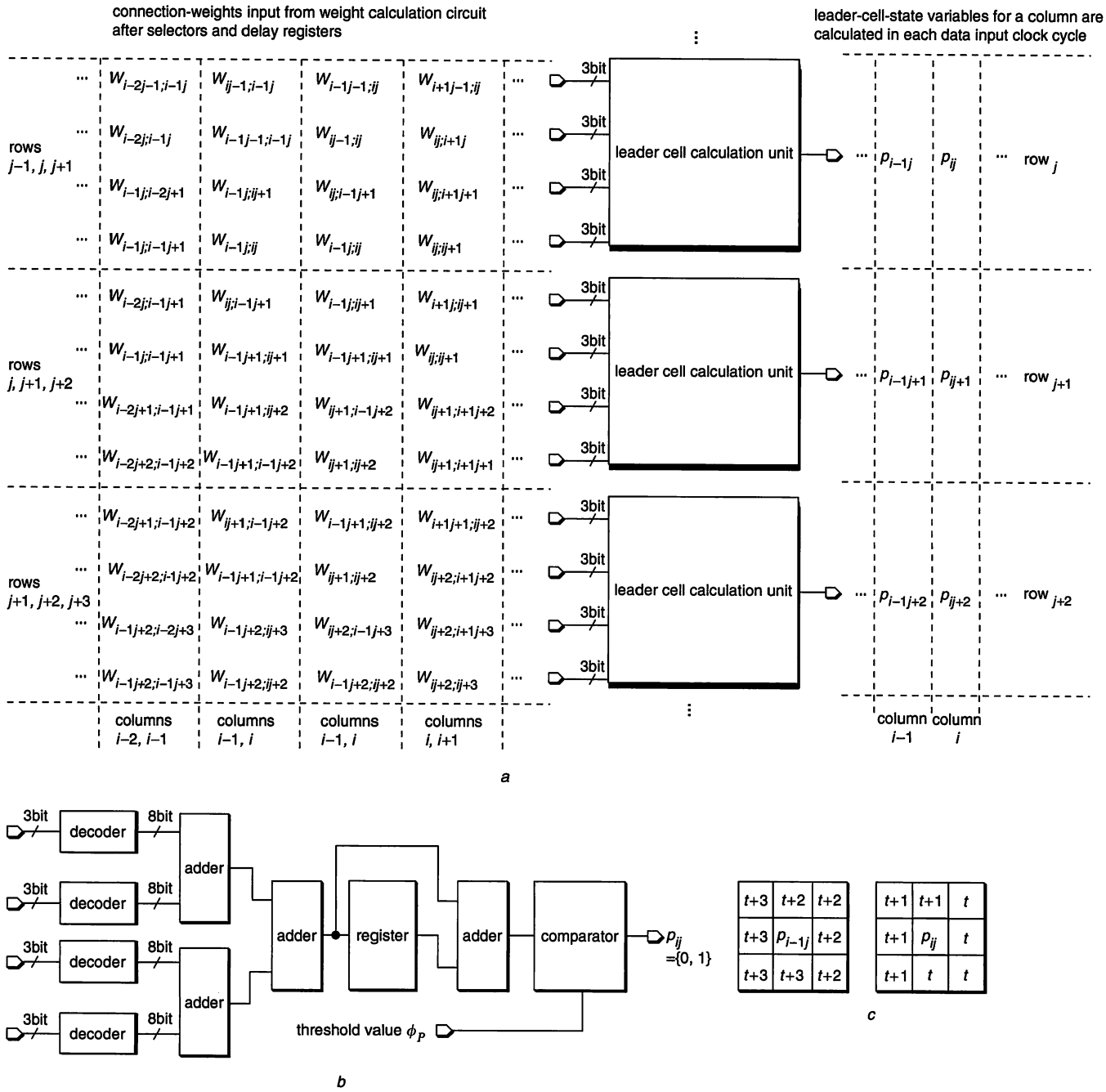

Fig. 8 *Block diagram of the leader cell selection circuit*
a Data flow diagram
b Block diagram of the leader cell calculation unit
c Order in which the connection-weights are transferred for P_{ij} decision, t denotes the clock cycle
– in case of P_{ij} the upper-left *WR* is vertical
– in case of P_{i-1j} the upper-left *WR* is horizontal

as $p_{ij}=1$ and otherwise $p_{ij}=0$. The supply sequence of the eight neighbouring connection-weights changes, according to the usage of horizontal and vertical weight-register blocks for even and odd columns (as well as rows) according to the scheme shown in Fig. 8*c*.

3.4 Segmentation restore circuit

The segmentation results, i.e. pixel/segment-number (label) pairs, are generated in the segmentation restore circuit. This means, this circuit assigns the pixels that are included in the grown region to the same segment number each time after a growth process has finished and transfers pixel/segment-number pairs to the image-segmentation memory.

There are two implementation possibilities: (1) For small size cell networks, most functions of the segmentation restore circuit can be carried out within the connection-weight-register blocks of the cell network expect for an external label generation circuit. In this case each connection-weight-register block needs a few additional gates for label write/read control. The external label generation circuit generates corresponding label numbers for each segment, and broadcasts them to all excited cells for storage in their upper-left weight-register blocks. The register l_{ij} (see Fig. 6*a*) is used as a flag indicating whether the cell (pixel) is included in the segment or not. Finally, after the complete image is segmented, the segmentation results are read-out from the connection-weight-register blocks of the cell network in a column-parallel mode and are transmitted to the image segmentation memory. However, the total circuit area for label read/write control and intermediate label storage in the weight-register blocks increases in proportion to the pixel number with this type of implementation. (2) For large-size cell networks, the information of excited cells of each grown segment is read-out to an external segmentation restore circuit in a column-parallel mode and pixel/label pairs are not written back into the weight-register blocks of the cell network for intermediate storage but are directly transferred to the image segmentation memory. This completely external implementation leads to smaller circuit area in the cell network, because label write/read control for the weight-register blocks is not necessary. However, all cell-state data must be outputted to the external segmentation restore circuit after each segment-growth, so that the processing speed is slower than for the first implementation with fully-parallel label writing into the segment-related weight-register blocks. In the test-chip design, presented in Section 4, the function of the segmentation restore circuit is realised with the first implementation possibility.

4 Performance evaluation of the VLSI architecture by simulation and test-chip fabrication

4.1 Simulated performance of the proposed VLSI architecture

Since the proposed cell network-based segmentation architecture of Figs. 3 and 4*a* is realised as a digital CMOS circuit, the achievable segmentation quality can be evaluated easily by simulation before the design of the hardware. Therefore, we have written an architecture simulator in C language (data-output interface in Java) and tested the architecture for many different image samples to verify good segmentation quality. Segmentation examples for a grey-scale (taken from a standard image database [26]) and a colour images are shown in Fig. 9. The example pictures are completely segmented. However, groups of unlabelled pixels in small, not coherent regions or in case of noise pixels occur. As expected, these regions happen to have no

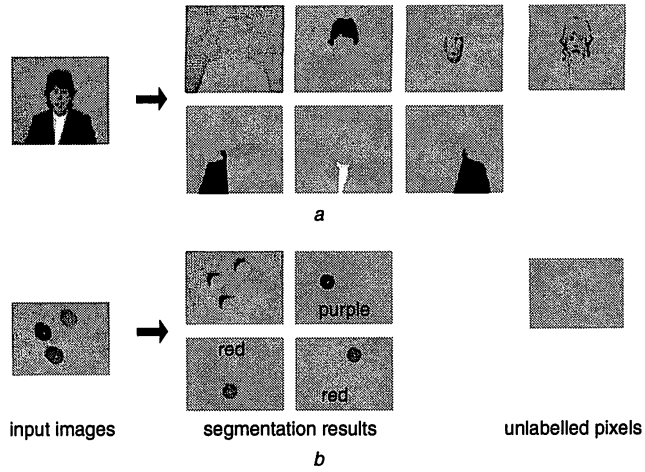

Fig. 9 *Examples of image segmentation with all extracted and unlabelled regions*
a Grey-scale image, size 311 × 279 pixels
b Colour image, size 320 × 240 pixels

leader pixel that can act as a seed for the growth process. Our method for handling these unlabelled pixels is to either ignore them or merge them with the neighbouring region that has the most similar luminance (or RGB data) in a post-processing step.

The theoretically estimated worst case for the segmentation process is a fine-grain checker-board image, which normally does not occur in nature. This results from the fact that the total segmentation time is proportional to the average pixel number that can be exited simultaneously in the growth steps of the image segments. The number of simultaneously excited pixels increases with the perimeter length of the grown region, which is smaller on the average for smaller segments. Therefore, the segmentation time increases in proportion to the number of image segments, which becomes largest for a fine-grain checker-board image.

The simulated worst case and in addition longest and shortest processing time for tested natural images are shown in Fig. 10. In the case of 320 × 240 (QVGA, 76 800 pixels) images, very high-speed image segmentation with the

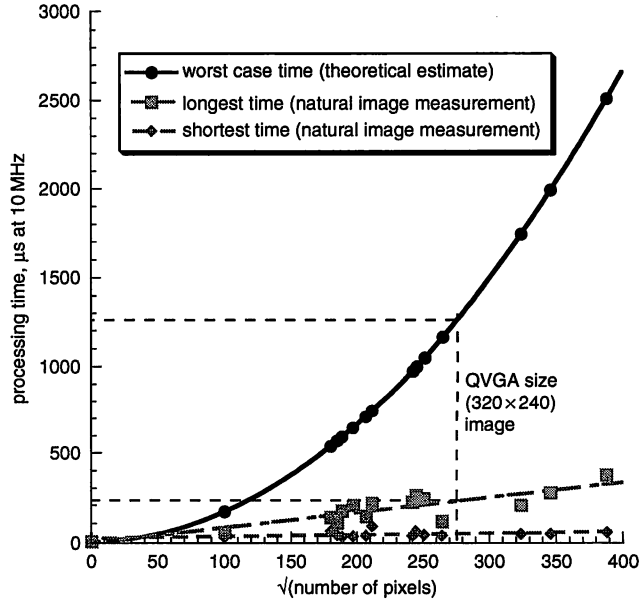

Fig. 10 *Processing time estimation of image-segmentation for larger image sizes at 10 MHz clock frequency (weight-parallel architecture)*

weight-parallel (respectively weight-serial) architecture in about r 300 μs (respectively r 2.7 ms) on the average and about r 1.3 ms (respectively r 11.7 ms), for the worst-case checker-board image are verified at 10 MHz clock frequency. Therefore, real-time image segmentation of good quality can be achieved even at such a low clock frequency for operation of the cell network.

4.2 Test-chip design in 0.35 mm CMOS

All circuits of the VLSI segmentation architecture, namely connection-weight calculation circuit (for grey-scale and colour images), leader cell selection circuit, image segmentation cell network, and segmentation restore circuit, have been designed and verified with the hardware description language Verilog-HDL. Layouts have been automatically generated with a standard cell library from this high-level Verilog design. In 0.35 μm CMOS technology with three metal layers, the resulting circuit areas per row of connection-weight calculation circuit and leader cell selection circuit are 0.078 mm^2 and 0.022 mm^2, respectively, which is small in comparison to the cell network.

Since the size of the complete cell network, which can be implemented by alternately laying a cell and a connection-weight-register block, determines the chip size and should not get too large, it is necessary to realise each cell and each register block as small as possible. Therefore, we have designed a full-custom cell network, including parts of the segmentation restore circuit, and a semi-custom external label generation circuit (standard-cell-library design) in the same 0.35 μm CMOS technology. The photomicrograph in Fig. 11 shows this full-custom image segmentation test chip after fabrication. The size of the cell network for segmentation with 10 × 10 cells is just 2.136 mm × 2.663 mm. The right part of Fig. 11 depicts a magnified layout image of a cell with its four adjacent register blocks. Table 1 summarises the characteristic data of the designed image segmentation test chip. The size of the cell network depends on the cells and weight-register blocks, so that we

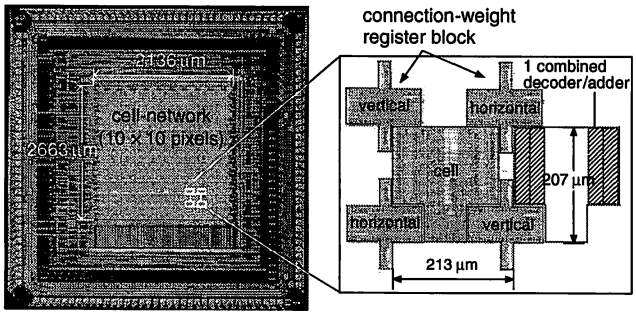

Fig. 11 *Photograph of the fabricated test chip for the cell network-based architecture in a 0.35 mm three-metal layer CMOS technology*
The hatched regions indicate the combined decoder/adder unit

Table 1: Characteristics of the test-chip

Implemented architecture	Weight-parallel architecture (10 × 10 pixels)
Technology	0.35 μm 2-poly 3-metal CMOS
Supply voltage	3.3 V
Measured max frequency	25 MHz
Worst-case power dissipation	36.4 mW at 10 MHz
Average power dissipation	24.4 mW at 10 MHz
Transistors	249, 810
Pixel integration density (PID)	19.6 pixel/mm^2

have optimised their layout implementation. The adder unit, which occupies the largest part of the cell area, is not realised with the conventional full-adder circuit with 24 transistors [27], but with a transmission gate full adder circuit requiring only 18 transistors per bit. Moreover, a significant simplification of each network cell was achieved by combining the decoder unit and the first stage of the adder as indicated in Fig. 11. These optimisations resulted in 47% circuit area reduction for each network cell as compared with a standard cell based implementation. For the connection-weight register block, 50% area reduction was realised by refining transistor-size design. Consequently, about 48% smaller area could be achieved by the full-custom design of the cell network.

4.3 Measurements with fabricated test chip

For the measurement of the CMOS test chip, various 10 × 10 pixel grey-scale test images were given as inputs. Since weight calculation and leader cell selection circuits could not be included in the test chip owing to the limitation of available space, corresponding connection-weights and leader-cell data were generated externally and loaded into the cell network before starting the segmentation.

Figure 12*a* shows the example of a checker-board like image with nine segments. The leader cell is located in the top-left corner of each segment and represents the seed of the region growing process. The numbers written into each of the 10 × 10 cells in Fig. 12*a* give the expected sequence of cell excitation during region growing. Figure 12*b* shows the input and output waveforms of the test chip as measured with a logic analyser for the checker-board image of Fig. 12*a*. The output signals OX0 ~ OX9 correspond to an OR-function of the x_{ij} register status of the cells in row0 ~ row9. '1' means that at least one cell in the corresponding row is in the excitation state. The power dissipation of this checker-board like example case was 17.8 mW at 10 MHz. Table 2 summarises segmentation time and power dissipation for five sample images. Part of the test chip's power dissipation is due to the inclusion of current-mode circuits. Because much faster processing speed than necessary for real-time application (30 frames/s) is achieved, it is possible to replace the current-mode circuitry by slower low-power circuitry to reduce power dissipation further.

4.4 Extrapolation to VLSI hardware for segmentation of higher resolution images

In this Section, we use the test chip results for estimating the VLSI-hardware requirements for segmentation of higher resolution images. The test chip design (Fig. 11) in 0.35 μm CMOS technology resulted in an integration density of about 5.10×10^{-2} mm^2/pixel for the cell network including the major part of the segmentation restore circuit. Since the present layout has still some unused areas, further layout compaction is possible. We have estimated an integration density of 4.55×10^{-2} mm^2/pixel when these unused areas are eliminated. The size of weight calculation circuit and leader cell selection circuit increases only in proportion to the row number of the cell network. From layout synthesis results with a standard-cell library, the integration density of weight calculation circuit (for colour image) and leader cell selection circuit is determined as 7.78×10^{-2} mm^2/row and 2.25×10^{-2} mm^2/row in 0.35 μm CMOS, respectively. To apply our VLSI architecture to the segmentation of images with higher pixel numbers, two possibilities are available. One straight-forward possibility is to use an advanced CMOS technology to increase the integration density, so that large cell networks can be integrated. The other

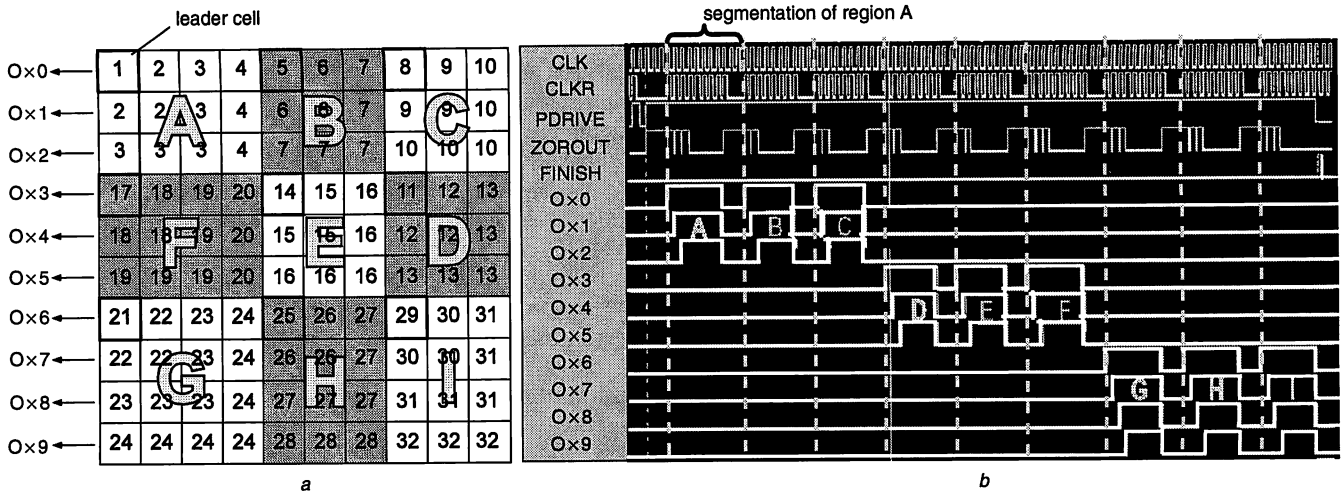

Fig. 12 *Image segmentation results with a checker-board image*
a Input image with checker-board pattern
b Input/output waveforms of the fabricated segmentation test chip

Table 2: Power dissipation and segmentation time for five test images

Input image pattern	Power dissipation at 10 MHz [mW]	at 20 MHz [mW]	Segmentation time at 10 MHz [μs]
Homogeneous	36.4	44.6	1.7
Stripe	19.8	28.4	5.2
S-shape	31.7	34.3	4.6
Triangle	16.8	28.0	1.3
Checker-board	17.8	31.0	9.5

possibility is to exploit the fast segmentation speed of the cell network for a sequential processing of tiles of the complete image with a correspondingly smaller cell network.

4.4.1 Large cell network in advanced CMOS technology: Figure 13 shows the extrapolation to a 90 nm CMOS technology resulting in integration density 1.50×10^{-3} mm^2/pixel and 6.62×10^{-3} mm^2/row for cell network and weight-calculation plus leader-cell selection circuits, respectively. Consequently, required chip size of the segmentation hardware is about 117 mm^2 for QVGA images. If more than five metal layers are available, even higher densities are possible by introducing dedicated routing layers for VDD and VSS. Since the typical chipsize for the cost–performance market in 2005 is about 140 mm^2 from the International Technology Roadmap for Semiconductors (ITRS2004 Update) [28], single chip implementation of our weight-parallel architecture for QVGA-size image segmentation is predicted possible with today's 90 nm CMOS technology. However, the power dissipation of a large size segmentation network is expected to become relatively large. One possibility for achieving low-power dissipation with such large-size cell networks is to keep only the boundary cells of the presently grown segment in active mode and all other cells of the cell network in low-power sleeping mode.

4.4.2 Processing of image tiles with small-size cell network: The fast segmentation speed of our proposed architecture can be further exploited for smaller

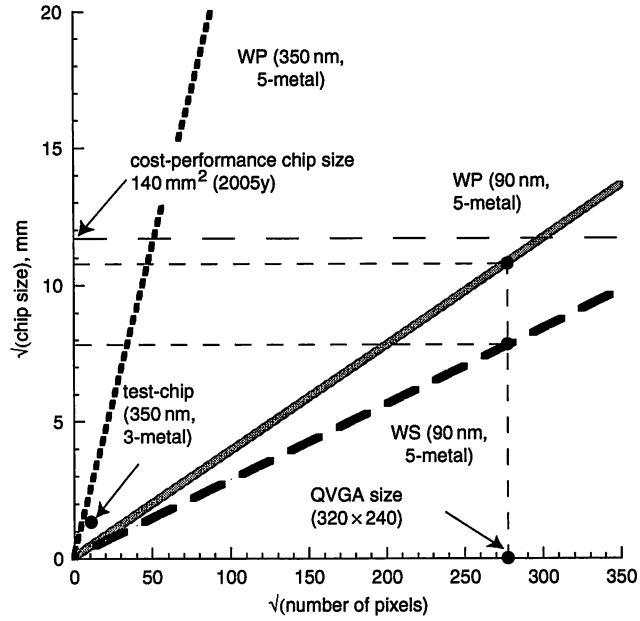

Fig. 13 *Chip size estimation for weight-parallel (WP) and weight-serial (WS) architectures for 90 nm CMOS technology with five metal layers against image size*

cell networks and thus reduce hardware cost by pipeline processing of tiled images. The basic concept of this approach is as follows. Input images are divided into tiles with an overlapped region of one row and one column. Then, each image tile is processed in sequential order with a correspondingly smaller cell network. The segment label numbers of the pixels in the overlap region are used for the segmentation of subsequent tiles to identify segments, which extend over several tiles. In consequence, this approach can avoid the necessity of a large-scale cell network, even for large input image sizes, and can enable compact integration. For VGA-size video pictures (640 × 480 pixels), we have roughly estimated the performance with a segmentation network of 41 × 33 cells. A segmentation time Γ 8 ms and a power dissipation Γ 100 mW are expected at 10 MHz. The necessary chip area for achieving this performance is estimated Γ 2.1 mm^2 in a 90 nm CMOS technology with five metal layers. These estimated data mean, furthermore, that real-time

segmentation of XGA (1024×768) and even SXGA (1280×1024) video images is within the reach of our proposed VLSI implementation architecture.

5 Conclusions

We have proposed a cell network-based digital image segmentation architecture for real-time applications with pixel parallel processing of grey-scale and colour images. A CMOS test chip for the cell network, which is the main functional stage, has been fabricated, in a 0.35 μm CMOS technology and verifies the effectiveness of our proposal. In the performance verification of the test chip, high-speed segmentation in Γ 9.5 μs and low-power dissipation of Γ 36.4 mW at 10 MHz are measured.

We have also proposed two extension possibilities for the segmentation of higher resolution images. The extrapolation results of the straight-forward method, which uses a large cell network in advanced CMOS technology, show that QVGA-size image segmentation will be possible within 300 μs at 10 MHz for a 90 nm CMOS technology. Another method for segmentation of higher resolution images is pipelined processing of image tiles. The performance with a segmentation network of 41×33 cells is Γ 8 ms segmentation time and Γ 100 mW power dissipation at 10 MHz in addition to Γ 2.1 mm^2 chip area for 90 nm CMOS technology with five metal layers. Consequently, the tiled-image-based approach offers a solution with a much better cost performance relation than the straight-forward implementation of large cell networks.

Future work includes test chip design of a cell network for pipeline processing with tiled images, implementing also peripheral circuits such as the circuits for connection-weight calculation and leader cell selection. The improvement of the architecture for very low-power dissipation and the development of a complete image segmentation system are further important topics.

6 Acknowledgments

The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo by the collaboration with Rohm Corporation and Toppan Printing Corporation. The authors gratefully acknowledge Professor T. Morie, Kyushu Institute of Technology, Dr. O. Kiriyama and Mr. H. Adachi, Hirosima University, and the anonymous reviewers for their valuable comments. Part of this work was supported by the 21st Century COE program, Ministry of Education, Science and Culture, Japanese Government, a Research Grand from Mazda Foundation and a Grant-in-Aid for JSPS Fellows, 1650741, 2004.

7 References

1 Danyali, H., and Mertinsand, A.: 'Object-based image compression with simultaneous spatial and SNR scalability support for multicasting over heterogeneous networks'. Proc. 7th Int. Symp. on DSP for Communication Sys. (DSPCS 2003), 2003, pp. 77–81
2 Strat, T.M.: 'Object-based encoding: next-generation video compression'. Proc. Workshop and Exhibition on MPEG-4, 2001, pp. 53–57
3 Japanese Ministry of Land, Infrastructure and Transport, Road Bureau ITS Homepage 2004 URL http://www.mlit.go.jp/road/ITS/index.html, 2004
4 Moving Picture Experts Group 2004 homepage URL http://www.chiariglione.org/mpeg/, 2004
5 Ranganathan, N., and Mehrotra, R.: 'A VLSI architecture for dynamic scene analysis', *CVGIP, Image Underst.*, 1991, **53**, (2), pp. 189–197
6 Chien, S.Y., Ma, S.Y., and Chen, L.G.: 'Efficient moving object segmentation algorithm using background registration technique', *IEEE Trans. Circuits Syst. Video Technol.*, 2002, **12**, (7), pp. 577–586
7 Chien, S.Y., Huang, Y.W., Hsieh, B.Y., and Chen, L.G.: 'Single chip video segmentation system with a programmable PE array'. Proc. IEEE Asia-Pacific Conf. on ASIC, 2002, pp. 233–236
8 Yang, R., and Welch, G.: 'Fast image segmentation and smoothing using commodity graphics hardware', *J. Graph. Tools*, 2002, **7**, (4), pp. 91–100
9 Jähne, B.: 'Digital image processing' (Springer-Verlag, Berlin, Heidelberg, New York, 2001, 5th edn.)
10 Russ, J.C.: 'The image processing handbook' (CRC Press, 1999), pp. 371–429
11 Bräunl, T., Feyrer, S., Rapf, W., and Reinhardt, M.: 'Parallel image processing' (Springer-Verlag, Berlin, Heidelberg, New York, 2000)
12 Sarkar, S., and Boyer, K.L.: 'Integration, inference, and management of spatial information using bayesian networks: perceptual organization', *IEICE Trans. Pattern Anal. Mach. Intell.*, 1993, **15**, (3), pp. 256–274
13 Ma, W.Y., and Manjunath, B.S.: 'Edgeflow: a technique for boundary detection and segmentation', *IEEE Trans. Image Process.*, 2004, **9**, (8), pp. 1375–1388
14 Shufeng, W., and Qun, Y.: 'The application of vision technology in AGVS,'. Proc. Int. Conf. on Agricultural Engineering, 1999, Vol. Number I, pp. 127–131
15 Kass, M., Witkin, A., and Terzopoulos, D.: 'Snakes: active contour models', *Int. J. Comput. Vis.*, 1987, **1**, (4), pp. 321–331
16 Gevers, T.: 'Robust segmentation and tracking of colored objects in video', *IEEE trans. Circuits Syst. Video Technol.*, 2004, **14**, (6), pp. 776–781
17 Papadimitriou, T., Diamantaras, K.I., Strintzis, M.G., and Roumeliotis, M.: 'Video scene segmentation using spatial contours and 3-D robust motion estimation', *IEEE Trans. Circuits Syst. Video Technol.*, 2004, **14**, (4), pp. 485–497
18 Ishiyama, Y., Kubo, F., Takahashi, H., and Tomita, F.: 'Labeling board based on boundary tracking', *Syst. Comput. Jpn.*, 1995, **26**, (14), pp. 67–76
19 Jean, S.D., Liu, C.M., Chang, C.C., and Chen, Z.: 'A new algorithm and its VLSI architecture design for connected component labeling', *Proc. Int. Symp. Cir. &Sys.*, 1994, **2**, (6), pp. 565–568
20 Mozef, E., Weber, S., Jaber, J., and Prieur, G.: 'Parallel architecture dedicated to image component labeling in *O(nlogn)*: FPGA implementation', *Proc. SPIE*, 1996, **2784**, pp. 120–125
21 Wang, D.L., and Terman, D.: 'Image segmentation based on oscillator correlation', *Neural Comput.*, 1997, **9**, (4), pp. 805–836
22 Ando, H., Morie, T., Miyake, M., Nagata, M., and Iwata, A.: 'Image segmentation/extraction using nonlinear cellular networks and their VLSI implementation using pulse-modulation techniques', *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 2002, **E85-A**, (2), pp. 381–388
23 Ando, H., Morie, T., Nagata, M., and Iwata, A.: 'A nonlinear oscillator network for gray-level image segmentation and PWM/PPM circuits for its VLSI implementation', *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 2000, **E83-A**, (2), pp. 329–336
24 Morimoto, T., Harada, Y., Koide, T., and Mattausch, H.J.: 'Low-complexity, highly-parallel color motion-picture segmentation architecture for compact digital CMOS implementation'. Ext. Abst. of 2002 Int. Conf. on Solid State Devices and Materials, 2002, pp. 242–243
25 Morimoto, T., Harada, Y., Koide, T., and Mattausch, H.J.: 'Real-time segmentation architecture of gray-scale/color motion pictures and digital test-chip implementation'. Proc. IEEE Asia-Pacific Conf. on ASIC, 2002, pp. 237–240
26 CityU-EE-CSL Image Database URL http://abacus.ee.cityu.edu.hk/benjiman/imagedb/, 2005
27 Weste, N.H.E., and Eshraghian, K.: 'Principles of CMOS VLSI design: a system perspective' (Addison-Wesley Publishing Company, 1993, 2nd edn.), pp. 513–526
28 ITRS 2004 Update: The International Technology Roadmap for Semiconductors 2004. URL http://public.itrs.net/