An Image Sensor using Quad Tree for Selective Scanning with Adaptive Resolution

Junichi Akita Kunihiro Asada
Department of Electronic Engineering, University of Tokyo
(VLSI Design and Education Center, University of Tokyo)
7-3-1 Hongo, Bunkyo-ku Tokyo 113, Japan
Tel: +81-3-3812-2111(ex. 6771) / Fax: +81-3-5800-5797
e-mail: {akita,asada}@silicon.t.u-tokyo.ac.jp

Introduction

With the increasing needs for image information, such as multi-media, channel capacity between image sensors and image processing units in the conventional system is becoming one of critical factors that prevent complex and high speed image sensing[1].

In this study, we propose a new signal scan method with a kind of image compression capability by skipping scan of non-active elements using a tree structure of automata, which we call "tree scan." It is shown that on-chip data compression about 1/18 to 1/2 is possible by the tree scan against the conventional raster scan, that is used in CCDs. We also show that the tree scan is easily applied for the adaptive resolution scan like human eyesight. We designed the tree-structured node automata and photo detectors to be integrated on one chip using 1.5μm CMOS technology..

Scanning Strategy

A. Image scan using tree structure

In the conventional raster scan method as shown in Fig.1(a), all pixels in a sensor are always scanned, even if the number of active pixels is expected small enough, in such cases as inter-frame difference of movies. It implies that there are redundant cycles in the conventional scan method, resulting in useless consumption of power and time.

Fig.1(b) shows a new method of signal scan using a tree structure, which is composed of binary pixels at the lowest level and the node automata, which has a value as logical-OR of its lower levels' nodes and returns it to its upper level when given *transition* signal. Thus the value of 0 implies that the values of all its lower nodes are 0, so that the further scan is not needed. When the value is 1, the node starts scan of the lower level in order, and returns *completion* signal to the higher node when the scan of its lower level is completed. Assuming no correlations of pixels, the mean code length of tree scan, $\overline{L}$ is derived as follows[2].

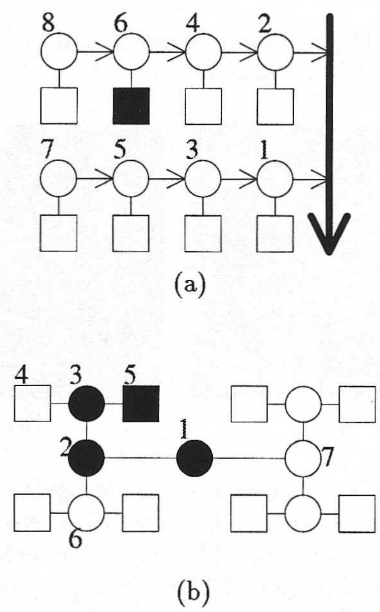


Figure 1: Two signal scan methods. (a)the conventional rater scan, (b)the scan using tree structure of automata. The circles are scan circuits and squares are pixels (active is black). The number in the figure indicates the sequence of scan.

$$\overline{L} = 1 + \sum_{l=2}^{N} b^{N-l} p_l b$$

$$= 1 + \sum_{l=2}^{N} b^{N-l+1}\{1 - (1-p_1)^{b^{l-1}}\}. \quad (1)$$

Here N is the number of depth of tree structure, b is the number of branches of one node, and p_1 is the active probability of each pixel, respectively. (Note that the number of pixels is equal to b^{N-1}.) The relation between $\overline{L}$ and p_1 is illustrated in Fig.2 for 2^{10} pixels, which indicates that $b = 4$ gives the smallest code length. It can also be shown smallest in all cases of the number of pixels. In this study we consider the tree structure with $b = 4$, i.e. so called "quad tree", which we call *1:4 tree* hereafter. The code encoded with this quad tree structure is called *1:4 tree code.*

B. Results of tree scan

Since the pixels may have spatial correlation each other in the actual images, the tree code length is expected to be shorter than in case of eq.(1), because it is derived assuming no correlation of the pixels, in

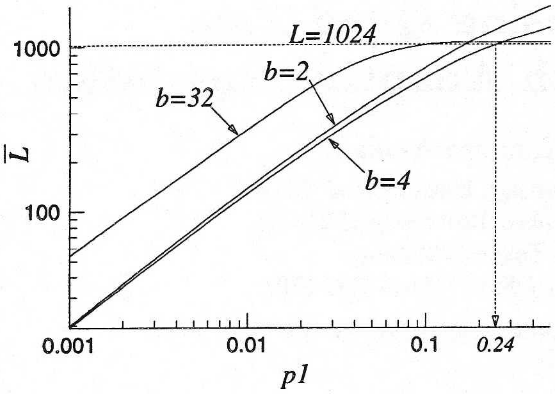


Figure 2: The relation between $\overline{L}$ and p_1.(the number of pixels is $2^{10} = 1,024$.)

(a) (b)

Figure 3: The original of sample image(a)(42,953bits) and the image with making attention only to its subarea of girl's face(b)(1,581bits). (The other area are scanned by the unit of 8×8 pixels.). (Note that the pictures in Fig.2 are illustrated as gray scale images for readers' convenience. These images are processed after digitizing to binary images in this study.)

other words, assuming case of the image with a very high spatial frequency such as white noise.

We carried out the simulation of tree scan for still binary images and the inter-frame differences of moving pictures for the applications of moving picture compression. The results of image scan using the quad tree are shown in Table 1(a) and Table 1(b) which show that the tree scan gives the code length about 1/2 to 1/6 (for still images) or about 1/2 to 1/18 (for inter-frame difference of movies) against the raster scan. It is also usually smaller than the run-length entropy which determine the lower bound of the run-length coding.

C. Adaptive resolution scan

It is notable that the tree scan proceeds from the top level to the bottom pixels in order, and the lower level gives the higher resolution, while the higher level gives the summerized information of the lower level. We can dynamically select two scan methods; either the normal scan for the focused area, or a scan intensionally stopped on the way for the other area. Thus we can obtain an image with adaptive resolution, where resolution is high only in the focused area. For example, the focused image in Fig.3(b) gives just 1,581 bits, which is about 1/27 against the normal scan in Fig.3(a). This adaptive resolution scan, like human eyesight has, is implemented by adding in each automaton a circuit judging whether to continue the scan of lower nodes or to finish, controlled by a signal given from the outside.

Table 1: Quad tree code length (L_{t0}) and the run-length entropy (H_{r0}), with the ratio of black elements (p_1), for eight examples of ITU-T facsimile test images(a) and the inter-frame difference of example movies(b). (L_{t0} and H_{r0} are shown as the value per pixel, or the ratio of code length over the number of pixels.)

(a)

Name	p_1	L_{t0} [bit/pix]	H_{r0} [bit/pix]
CCITT-1	0.037	0.175	0.178
CCITT-2	0.045	0.159	0.240
CCITT-3	0.080	0.310	0.304
CCITT-4	0.141	0.573	0.452
CCITT-5	0.075	0.296	0.327
CCITT-6	0.049	0.188	0.269
CCITT-7	0.105	0.460	0.534
CCITT-8	0.450	0.753	0.313

(b)

Name	p_1	L_{t0} [bit/pix]	H_{r0} [bit/pix]
MissAmerica	0.011	0.055	0.106
TableTennis	0.114	0.454	0.584
FlowerGarden	0.172	0.601	0.724
Neck	0.017	0.067	0.125
Rail	0.036	0.122	0.149

Implementation of Tree Scan Image Sensor

A. Direct implementation of tree automata and sensors

The transition diagram and signals of the node automaton are shown in Table 2 and Fig.4, respectively.

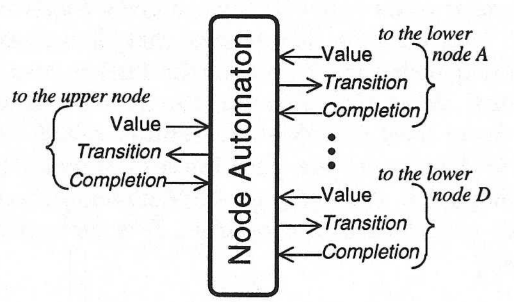


Figure 4: Signals of node automaton.

Table 2: Transition diagram of node automaton.

Inputs								Outputs					
T	V	CA	CB	CC	CD	S	S'	TA	TB	TC	TD	V	C
0	–	–	–	–	–	W	W	0	0	0	0	VO	0
↑	0	–	–	–	–	W	W	0	0	0	0	VO	1
↑	1	–	–	–	–	W	SA	↑	0	0	0	VA	0
↑	–	0	–	–	–	SA	SA	↑	0	0	0	VA	0
↑	–	1	–	–	–	SA	SB	0	↑	0	0	VB	0
↑	–	–	0	–	–	SB	SB	0	↑	0	0	VB	0
↑	–	–	1	–	–	SB	SC	0	0	↑	0	VC	0
↑	–	–	–	0	–	SC	SC	0	0	↑	0	VC	0
↑	–	–	–	1	–	SC	SD	0	0	0	↑	VD	0
↑	–	–	–	–	0	SD	SD	0	0	0	↑	VD	0
↑	–	–	–	–	1	SD	W	0	0	0	0	VO	1

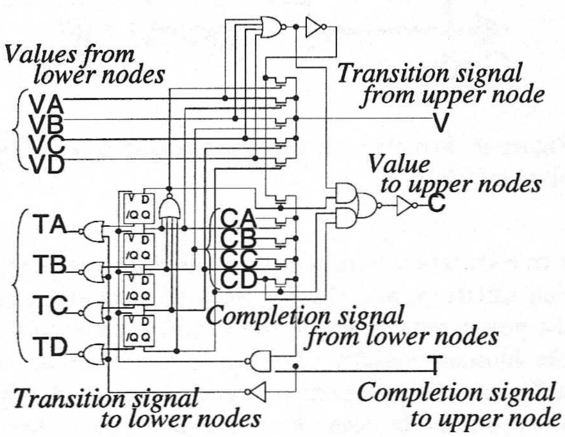


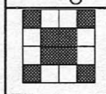
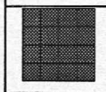
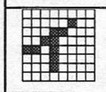
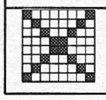
Figure 5: Circuit of node automaton.

A designed circuit of the node automaton is shown in Fig.5, where the clock signal is gated so as to propagate just through the scan path, which we call *selective activation*, which is expected to be effective for the application of high resolution image sensors .

We carried out `spice` simulation of tree structures composed of this circuit to estimate the power consumption for various images. Results of scanning 4×4 and 8×8 pixels are summerized in Table 3. In cases of 8×8 pixels, the 1:4 tree has one top level node, 2×2 middle level nodes, 4×4 lower level nodes, and 8×8 pixels (these were given as voltage sources in this simulation) in order. Table 3 shows the mean energy consumption in the whole trees per one bit in the code. The total energy consumption scanning the whole image is the product of the energy per bit and the code length. This result shows that the energy consumption per bit keeps almost constant independent to the image pattern. The constant energy is not sensitive to the number of pixels, since the length of the signal path activated from the top to a pixel is proportional not to the number of pixels but to the levels of the tree.

A possible layout of the quad tree is shown in Fig.6, that can place photo detectors uniformly with node automata full-filling the surface of chip, which can

Table 3: The power consumption U of 1:4 tree structure in the scan of 4×4 or 8×8 pixels.

Image	# of pixels	Code length	U[pJ/bit]
	4×4	21bit	0.760
	4×4	21bit	0.765
	8×8	41bit	0.879
	8×8	53bit	0.850

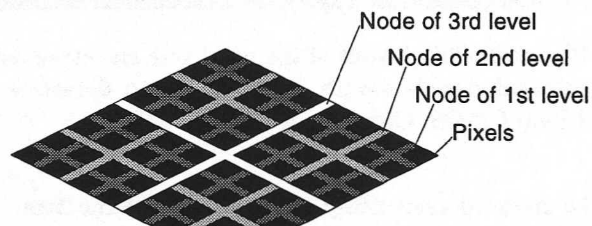


Figure 6: Possible layout of photo detector and node automaton of 1:4 tree structure.

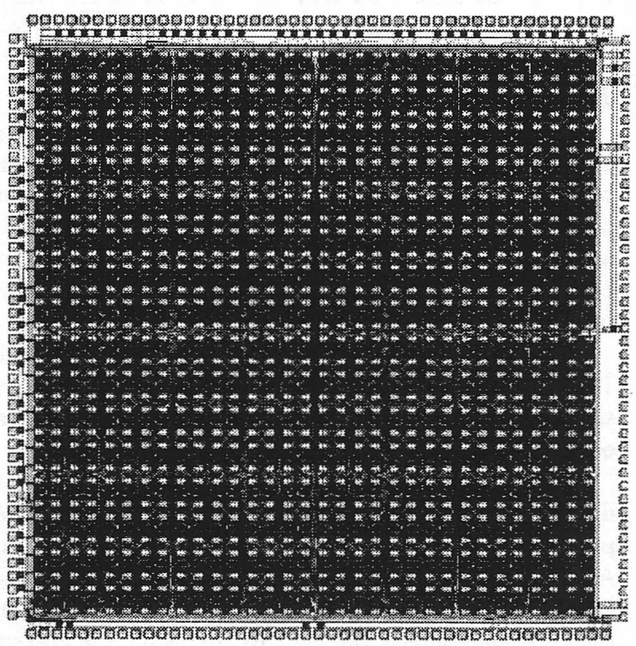


Figure 7: Chip layout of the quad tree structure with 32×32 photo detectors by 1.5μm CMOS. Chip size is 7.2mm×7.2mm.

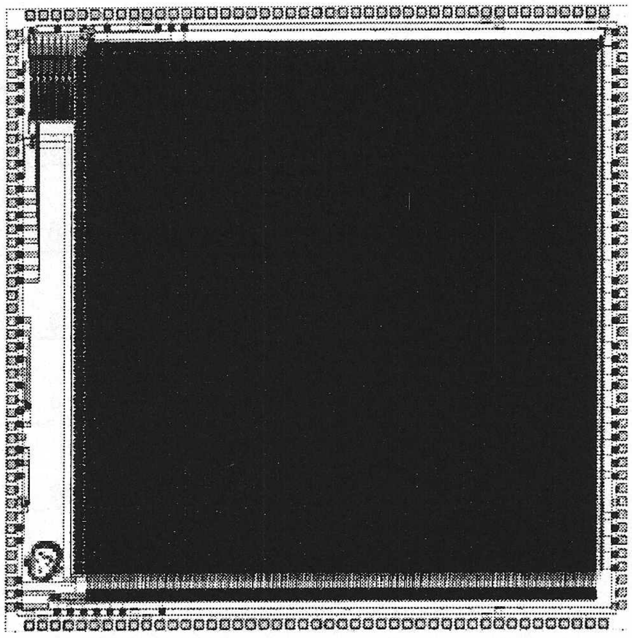


Figure 8: Chip layout of the quad tree structure using external decoders with 128×128 photo detectors by 1.5μm CMOS. Chip size is 7.2mm×7.2mm.

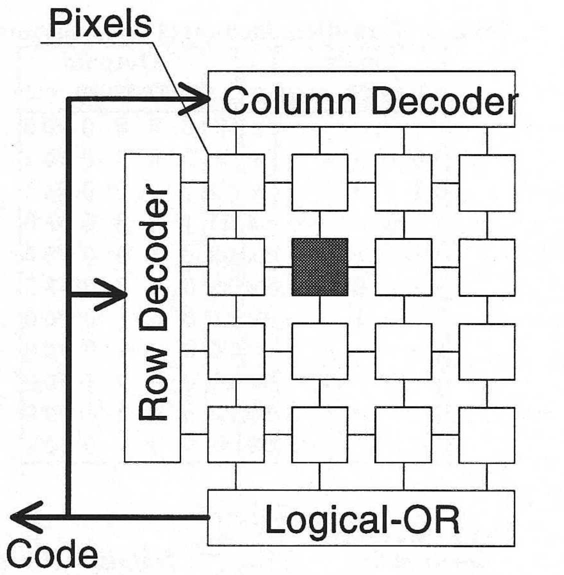


Figure 9: Structure of tree scan sensor using external pixel selectors.

be iterated recursively for any level of the tree. In this layout the higher levels' automata can occupy the larger area, and this is reasonable in order to minimize delays, since the higher levels' node circuits have to drive longer signal wires[3]. We designed this two-dimensional quad tree structure including node automata and 32 × 32 photo detectors for 1.5μm CMOS with double metal layers, as shown in Fig.7, with the fill-factor of 30%. This chip is now under fabrication, and measurement results will be reported at the conference.

B. Alternative implementation

Although the implementation described above is a straight forward method realizing the 1:4 tree scan with the selective activation, the fill-factors of the photo-diodes is relatively low because of the extra-area occupied by the automata, that resulted in a low resolution sensor of 32x32 pixels of a 1.5 um CMOS technology.

Figure 8 shows an alternative implementation of the 1:4 tree sensor with 128×128 pixels designed using the same technology and the same chip area, the function of which is identical to the 32x32 sensor.

This sensor has a memory-like architecture of photo-detectors with a new access modes not only to each photo-detector but also to a block of photo-detectors. Along with control circuits placed around the periphery of the photo-detector array, as described in Fig.9, the chip can emulate the 1:4 tree scan with the almost same clock rate as the direct implementation, though the power consumption is higher.

Conclusion

We have proposed a new image scan method using a tree structure with a kind of on-chip data compression efficiency, and the tree scan is adequate for both the power reduction and the adaptive resolution scan like human eyesight. For the increased image data in future, the tree scan is expected to be effective to overcome the bottleneck of channel capacity between image sensors and image processing units; the image data can be compressed directly with the capability of adaptive resolution to cope with such cases as abrupt switching of scenes in the inter-frame difference, where lowered resolution is inevitable not only in intensity but also in space to keep the data rate constant.

The future development of VLSI technology will enable us to fabricate the tree-structured image sensor with the more pixels. For example 480×480 pixels can be integrated with the tree structure by the direct implementation, and 1920×1920 pixels by the alternative implementation if using 0.1μm CMOS technology.

References

[1]W.Lawler *et al.*, "Performance of High-frame-rate, back-illuminated CCD imagers," *Proc. SPIE*, Vol.2172, p.90, 1994.

[2]K.Asada *et al.*, "A Tree Structure of Automata for Selective Image Scanning and Its Implementation," *IIZUKA '96*, p.113, 1996.

[3]N.H.E.Weste *et al.*, "Principles of CMOS VLSI design: a systems perspective," Addison-Wesley, 1988.