Chapter 3

Tree Structure of Image Signals for Image Sensors

This chapter will describe the idea of using tree structure of image data for a kind of data compression. The analytical model of tree structure and its evaluation will be discussed in section 3.2 and section 3.3. It will be also discussed the decoding algorithms and its implementations of encoded codes by tree structure in section 3.4.

3.1 Selective Scanning of Image Signals and Tree Structure

3.1.1 Image scanning for selected areas

The procedure of CCD image sensors to read out the image data is to read out the all images of photo detectors regardless of its value, which is called "raster scan." In the raster scan, all the pixels have be scanned, even in case of the distributed data with two-dimensional correlation, as shown in Figure 3.1(a), and it will result in redundant scan steps for the large white areas. It will be very remarkable in the extreme case of very few pixels are active, which is often caused by taking the effective pixels of inter-frame difference for movie compression.

On the other hand, when we see something by our eyes, we will make attentions

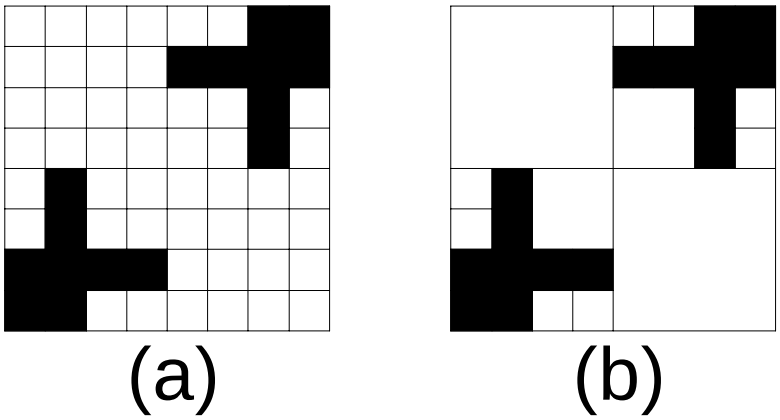


Figure 3.1: Sample of distributed image signals with two-dimensional correlation(a), and the area division for its distribution.

for the restricted area in the view and obtain the detail information from there, not always get the sharp images of whole area. The application of raster scan for such the scan for the restricted area will also result in redundant cycles for the non-interested areas.

3.1.2 Tree structure of image signals for selective scan

In such case of the image in Figure 3.1(a), one of the idea to reduce the redundant cycle is to divide the areas according to whether the divided areas contain the interesting data or not, for example, as shown in Figure 3.1(b).

Here we propose to apply the tree structure for image signals. The tree contains nodes, which have b nodes in the lower level and connected to the node in the higher level, and the nodes in the lowest level have b pixels, whose value is binary, in its lower level. The all connections between two nodes have three signal channels; the *start* signal to order the scan for its lower node, *completion* signal to inform the finish of its scan for its higher node, and *value* signal to inform its value for its higher node. (They are summerized in Figure 3.2.) We define the functions of each nodes as follows.

- It has the value of logical-OR of its lower b nodes. The node in the lowest level has the value of logical-OR of the b pixels.

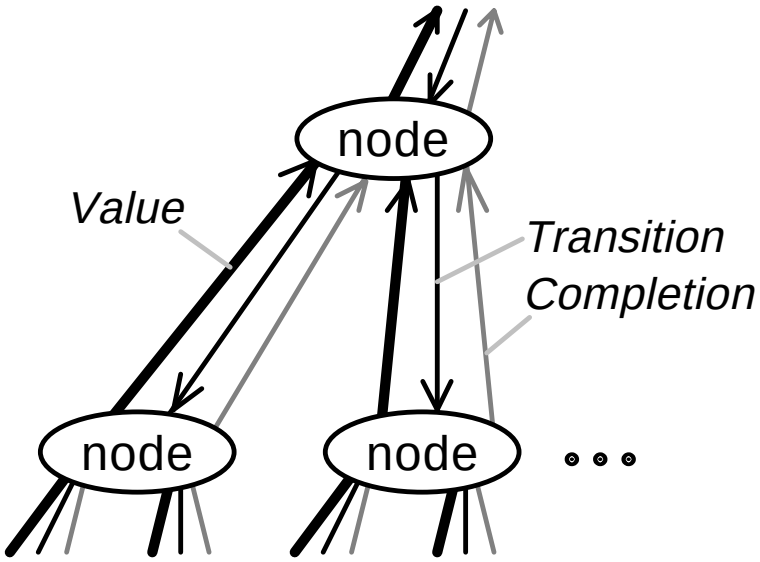
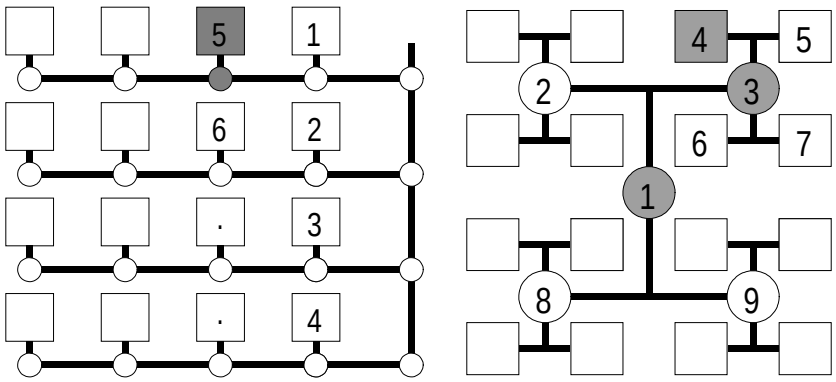


Figure 3.2: Signals of the nodes in the tree structure.

- It returns its value when it receives *start* signal from the node in the higher level.
- When its value is "1", it implies that there are at least one "1" in its lower level, and executes the scan by activating *start* signals of its lower b nodes in order. If the scanned lower node node has returned "1", it will wait for the finish of the lower node's scan until it returns *completion* signal. When the all scans of the lower b nodes have finished, it returns *completion* signal to its higher nodes.
- When its value is "0", it implies that there are no "1"s in its lower level and all the descending nodes to the pixels, and finishes the scan by returning *completion* signal.
- It has $(b+1)$ internal states; `W` for waiting state, and `Si`$(i = 1, 2, ...b)$ for the state of scanning the i-th nodes in its lower level.

For example, the raster scan of 4×4 pixels as shown in Figure 3.3(a) will proceed in order as shown in figure, and it will make a 16 bits code, which is equal to the number of pixels. (Note that white pixels or nodes represent "0" and gray pixels or



(a) (b)

Figure 3.3: The two procedures of image scan; (a)conventional raster scan, (b)scan using tree structure.(The square and the circle represents the pixels and scanning circuits, respectively. The numbers in the figure represents the steps of scan.)

nodes represent "1".) On the other hand, the scan using the tree structure of the nodes defined above, as shown in Figure 3.3(b) will proceeds as the following steps. (Note that $b = 4$ is assumed in this case.)

1. First the highest nodes 1 is scanned, and it returns "1".
2. Next, the scan proceeds to the second level, the node 2 at first. The node of 2 returns "0", and no more scan for its lower level are needed.
3. The scan proceeds to the node 3, and it returns "1". According to this value, the following four pixels are scanned in order, 4, 5, 6, and 7, and then the scan for node 3 has finished.
4. Next, the node 8 and 9 are scanned in order, and both of them return "0", and then the all scans have finished.

In the above procedure of scan using tree structure, we obtain 9 bits code, which is shorter than that of the raster scan. This scan procedure using tree structure (we call "tree scan" afterward), will have the following advantages.

- The scan step for the area containing all 0s are drastically skipped, which is a kind of data compression.

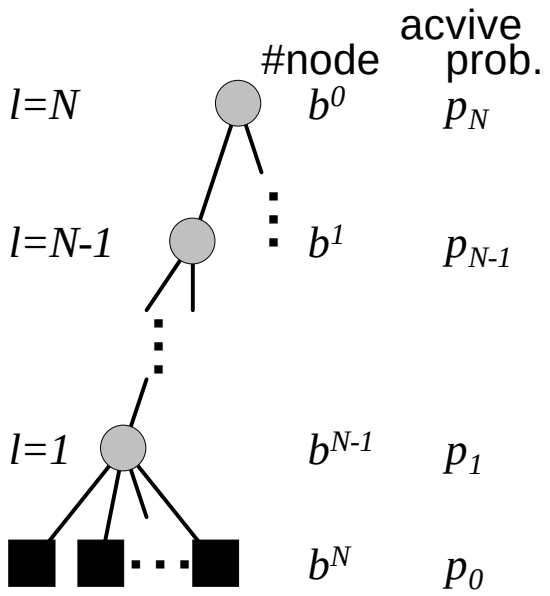


Figure 3.4: Analytical model of tree structure

- Only the signal path from the nodes or pixels scanned to the outside of image sensor is activated, and most of signal pathes rest are not activated, which will be effective for power reduction.

It is also notable that the more the tree scan step proceeds to the lower level, the smaller divided areas, or precise details, are scanned. This characteristic of tree scan gives the idea of executing the scan of the needed resolution for the needed sub-areas, or "adaptive resolution scan", which will be discussed in section 4.2.

3.2 Model and Analysis of the Scan Procedure of Tree-Structured Image Signals

3.2.1 Model of tree structure of image signals

We consider the model of tree structure discussed in the previous section as shown in Figure 3.4. Here b is the number of nodes connected under one node, and N is the number of levels, and p_l is the probability that the node in l-th level has the value of "1", and the number of pixels is expressed as b^N. Note that p_0 is the probability

that the pixel has the value of "1", or it is active. The node in l-th level has the value of "1" when at least one of b nodes in its lower level has the value of "1", and then p_l should satisfy the following equation.

$$p_l = 1 - (1 - p_{l-1})^b \tag{3.1}$$

Solving the Equation (3.1), p_l is expressed as follows.

$$p_l = 1 - (1 - p_0)^{b^l} \tag{3.2}$$

3.2.2 Mean code length of tree-scan

Here we can calculate the expectation of the number of steps in the procedure of the tree scan, or the expectation of code length $\overline{L}$ as the following procedures.

1. First, the node at the highest level, $l = N$ is scanned, and $\overline{L} = 1$.
2. If the node at the highest level has the value of "1", the b nodes in $l = N - 1$ are scanned, and $\overline{L} = \overline{L} + b$.
3. The following scan are executed just for the nodes whose value is "1" in b nodes at $l = N - 1$, which are $b \cdot p_{N-1}$ nodes, and $\overline{L} = \overline{L} + b \cdot b \cdot p_{N-1}$.
4. The following scan are executed just for the nodes whose value is "1" in b^2 nodes at $l = N - 2$, which are $b^2 \cdot p_{N-2}$ nodes, and $\overline{L} = \overline{L} + b \cdot b^2 \cdot p_{N-2}$.

Iterating the procedures above, the expectation of code length, $\overline{L}$ is derived as follows.

$$\begin{aligned} \overline{L} &= 1 + \sum_{l=1}^{N} b^{N-l+1} p_l \\ &= 1 + \sum_{l=1}^{N} b^{N-l+1} \{1 - (1 - p_0)^{b^l}\} \end{aligned} \tag{3.3}$$

The relations of $\overline{L}$ and the active probability of pixel, p_0 are shown in Figure 3.5 for various b, in case of the number of pixels, b^N is 1,024. The results in Figure 3.5 indicates that the tree structure with the number of branches, b is 4 gives the shortest code length for all p_0, and thus we treat the tree structure with $b = 4$, we

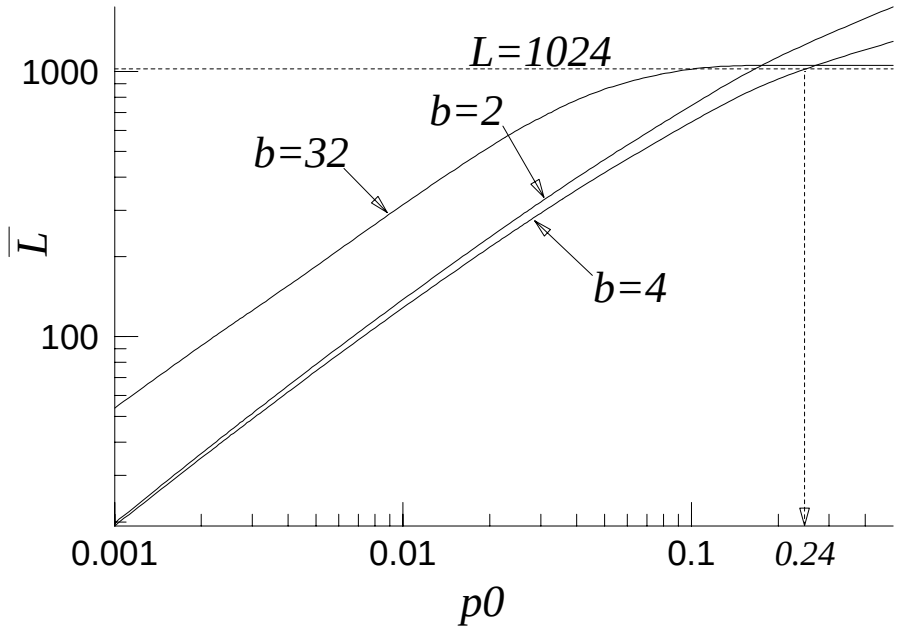


Figure 3.5: The relations of mean code length,$\overline{L}$ and the active probability of pixel, p_0 for various b.

call "1:4 tree structure" afterward. It is also indicated that $\overline{L}$ of 1:4 tree scan is shorter than the length of raster scan, 1024, in case of smaller p_0 than 0.24. (These results were derived not only for 1,024 pixels, but also for the more pixels.)

The number of branches, b and the number of levels, N are correlated through the number of pixels, n, as $n = b^N$, or $N = \log_b n$. Using this equation, Equation (3.3) can be easily extended to the case of the various b as follows.

$$\overline{L} = 1 + \sum_{l=1}^{\lfloor \log_b n \rfloor} b^{\log_b n - l + 1} \{1 - (1 - p_0)^{b^l}\} \tag{3.4}$$

Figure 3.6 shows the optimum b which gives the shortest $\overline{L}$ for given p_0. For large p_0, the scan steps will be executed for almost all nodes, and the case which the total number of nodes is small, or b is large, gives the shortest $\overline{L}$. For small p_0, the optimum b for various p_0 is between 3 and 4, and it indicates the properness to employ 1:4 tree structure in terms of minimizing $\overline{L}$.

In Equation (3.3), we assume no spatial correlations between pixels, but the pixels in usual images are expected to have distributions with the spatial correlations. The

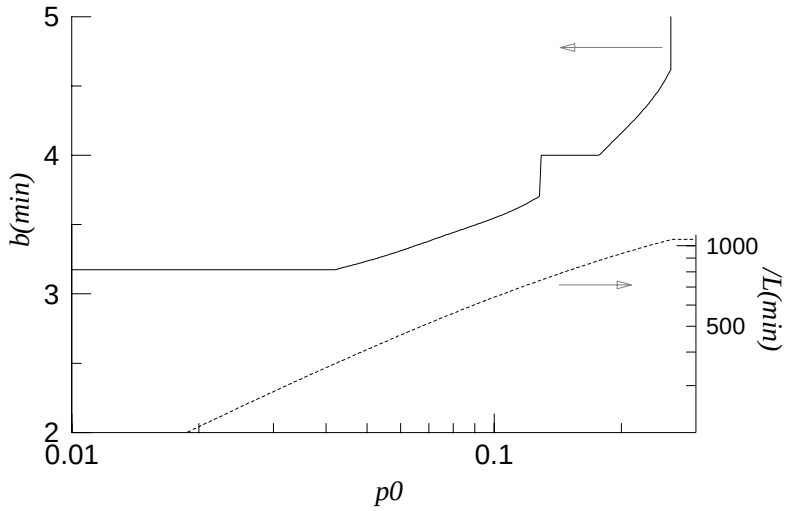


Figure 3.6: The optimum b which gives the shortest $\overline{L}$ for various p_0, and $\overline{L}$.

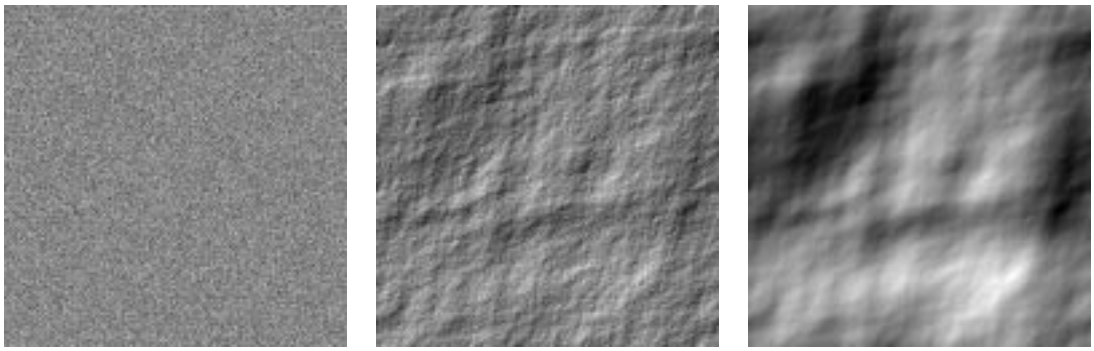
tree scan for two dimensional images is executed for the smaller divided areas as the step of scan proceeds to the lower level, and we can say that this procedure has a characteristic of gathering the pixels in the target sub-areas at each step. It is expected that $\overline{L}$ will be smaller for real images with spatially correlated pixels, than the case without spatial correlations as derived in Equation (3.3), or the white noise images containing very high spatial frequency.

We have generated the random images whose spatial power spectrums are proportional to $1/f^n$, where f is the spatial frequency, and carried out the procedure of 1:4 tree scan for them. Note that the images with larger n contains the less component of higher spatial frequency, as shown in Figure 3.7. Power spectrum $P(f)$ of two dimensional images $p(x, y)$ is defined by the following equation.

$$P(f) = P(\sqrt{k^2 + l^2}) \propto \left| \iint p(x, y) e^{j(kx+ly)} dxdy \right| \tag{3.5}$$

The random images whose power spectrum $P(f) = 1/f^n$ are given by invert Fourier transform of $e^{j\theta_f}/f$, where θ_f is the phase of each frequency components, and they should be determined randomly.

The results of calculating $\overline{L}$ for various images are shown in Figure 3.8. (The number of pixels is $2^{16} = 65,536$.) It is known that the spatial power spectrum of usual real images has a tendency to having between $1/f^{-1}$ and $1/f^{-2}$, and the



(a) (b) (c)

Figure 3.7: Sample of random images with various spatial power spectrum. (a)$1/f^0$, (b)$1/f^{-1}$, and (c)$1/f^{-2}$.

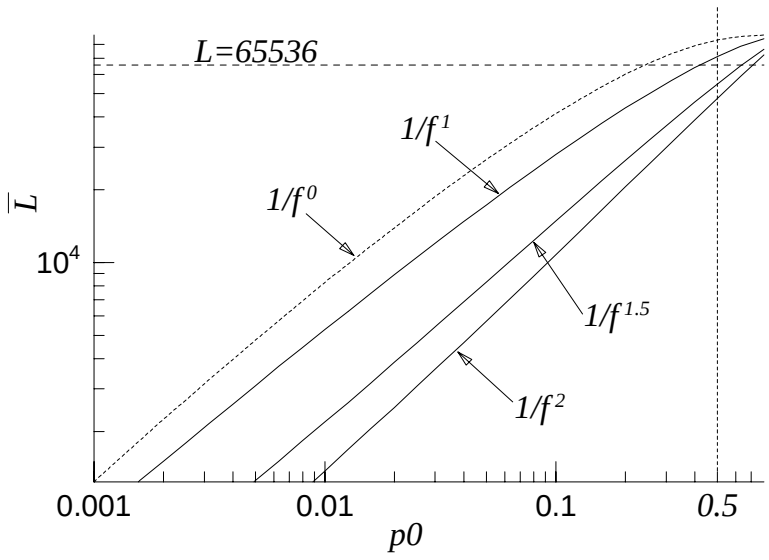


Figure 3.8: Calculated $\overline{L}$ for the images with various spatial power spectrum.

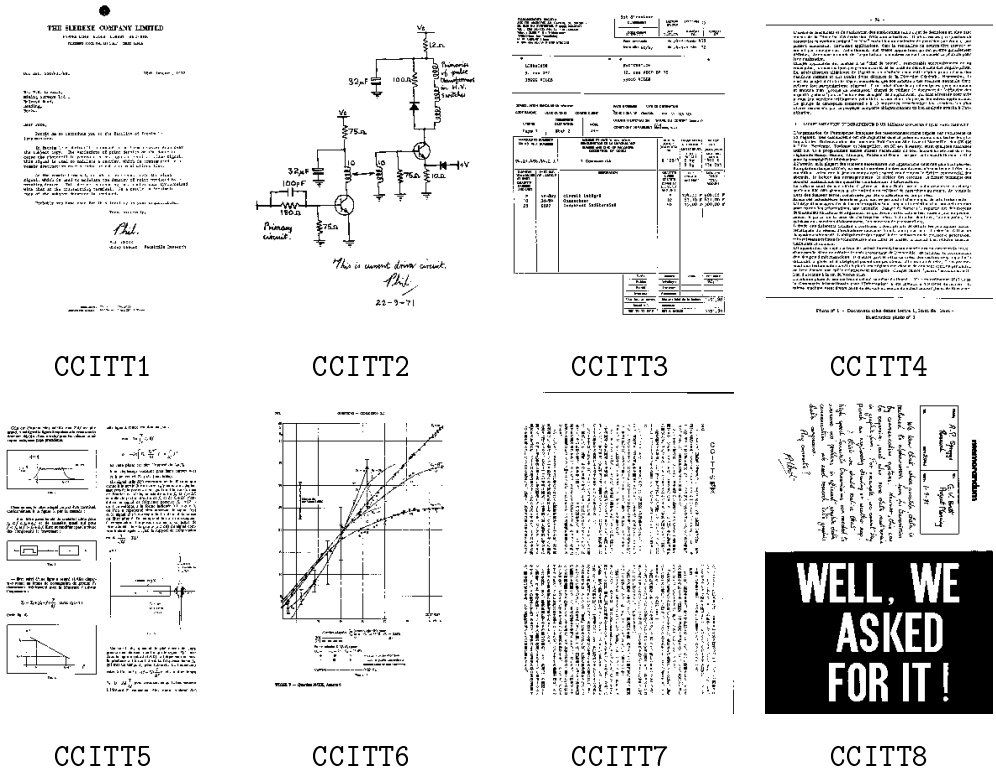


Figure 3.9: ITU-T facsimile standard images

possibility was indicated that $\overline{L}$ using 1:4 tree structure will be shorter than that of the raster scan, even in case of p_0 is larger than 0.5, or the more than half of pixels are black in usual real images.

The history of the idea applying quad tree structure for image signals is very old[18], but the efficiency applying quad tree for image signals was derived in terms of the image encoding efficiency discussed in above.

3.3 Experiments of Tree-Scanning for Images

In this section, the simulation results of 1:4 tree scanning for various real images are shown, with comparing the other scanning methods; the raster scan or run-length compression.

3.3.1 Tree scanning results for binary still images

Table 3.1: Scanning results for ITU-T facsimile standard binary images. (L_{t0} is the 1:4 tree code length per pixel, H_{r0} is the run-length entropy per pixel, and p_0 is the ratio of black pixels.)

Name	p_0	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

The 1:4 tree scan was carried out for the standard binary images for the evaluation of facsimile given by ITU-T (formerly CCITT) as shown in Figure 3.9.

The run-length compression is often used to scan binary images by one line, and to encode for the obtained binary bits. Some encoding code for run-length compression are known, for example MH (modified Huffman) code which is employed in G3 facsimile standard, and the lower bound of compression efficiency by one dimensional run-length compression is given by the following run-length entropy per pixel, H_{r0}.

$$H_{r0} = H_r/T = -\sum_{i=1}^{N} P_i \log_2 P_i \Big/ \sum_{i=1}^{N} iP_i \tag{3.6}$$

Here P_i is the relative frequency of run whose length is i, H_r is the run-length entropy, and T is the mean run length.

The scanning results are shown in Table 3.1 with the comparison with run-length compression. Table 3.1 shows that 1:4 tree scan is more effective especially in case of smaller p_0, which is just 1/2 to 1/7 against that of the raster scan. It is also notable that 1:4 tree scan is often more effective than run-length compression in most cases, especially in case of p_0 is small.

Table 3.2: Scanning results for inter-frame difference of movies. (L_{t0} is the 1:4 tree code length per pixel, H_{r0} is the run-length entropy per pixel, and $\overline{p_0^m}$ is the ratio of black pixels.)

name	$\overline{p_0^m}$	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

3.3.2 Tree scanning results for moving pictures

We carried out the 1:4 tree scan for inter-frame difference of some example movies. It is one of the most simple algorithms for movie compression to take inter-frame differences and scan just the pixels which have differences. Some example movie sequences are digitized to binary images, and the scan was carried out for the inter-frame differences, which is just exclusive OR of pixels in two succeeding frames. The scanning results are shown in Table 3.2 with the comparison of run-length compression. Here "MissAmerica", "TableTennis", and "FlowerGarden" are the standard of movies, and "Neck" and "Rail" are the sample movies captured by Hatori-Aizawa Laboratory, University of Tokyo, and the cuts of movies are shown in Figure 3.10. In "MissAmerica," the upper half of a lady is shown, and she talks with the motion of the body slowly. In "TableTennis," the view of table tennis game is shown, and it has two drastical scene change, with zoom out. In "FlowerGarden," the view of the garden is shown, and the whole scene moves slowly by the move of camera. In "Neck," the toy of moving rabbit is shown, and its head and hands are moving quickly. In "Rail", the toy of train is shown, and it moves slowly.

Table 3.2 indicates that only a few percents of pixels should be scanned, and 1:4 tree code length is about 1/14 to 1/8 over that of the raster scan. It is also indicated that the 1:4 tree code length is about 1/2 at maximum over that of the run-length

"MissAmerica"

"TableTennis"

"FlowerGarden"

"Neck"

"Rail"

Figure 3.10: Scene cuts of used sample movies

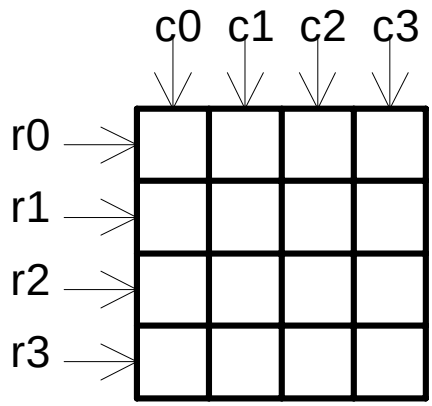


Figure 3.11: Example of 4×4 memory cells and the selection signal lines.

compression.

3.4 Decoding procedure of tree-scanned image signals

The encoded code by 1:4 tree scan can be decoded to binary two dimensional images, of course, since 1:4 tree scan is reversible procedure. Assuming that the decoded binary images are stored in two dimensional array of memory cells, and they are selected by the selection signal lines controlled by the external controller, as shown in Figure 3.11. The memory cells at (i, j) are selected when both column select line c_i and row select line r_j are activated, and the binary data is stored for all of the selected memory cells. This architecture of memory cells is similar to that of the conventional data memory, such as DRAM, but the difference is that more than two select signal lines can be activated so as to select the areas.

For example, the 1:4 tree code of `101101000` can be decoded by the following procedures, as shown in Figure 3.12.

1. First bit of 1:4 tree code, "1" represents the logical-OR of whole area, and all the memory cells are selected to be stored "1", as shown in Figure 3.12(a).

2. The following bit of code, "0" represents that the logical-OR of the upper-left 2×2 area is "0", and it is stored to this area with activating the corresponding select lines, as shown in Figure 3.12(b).

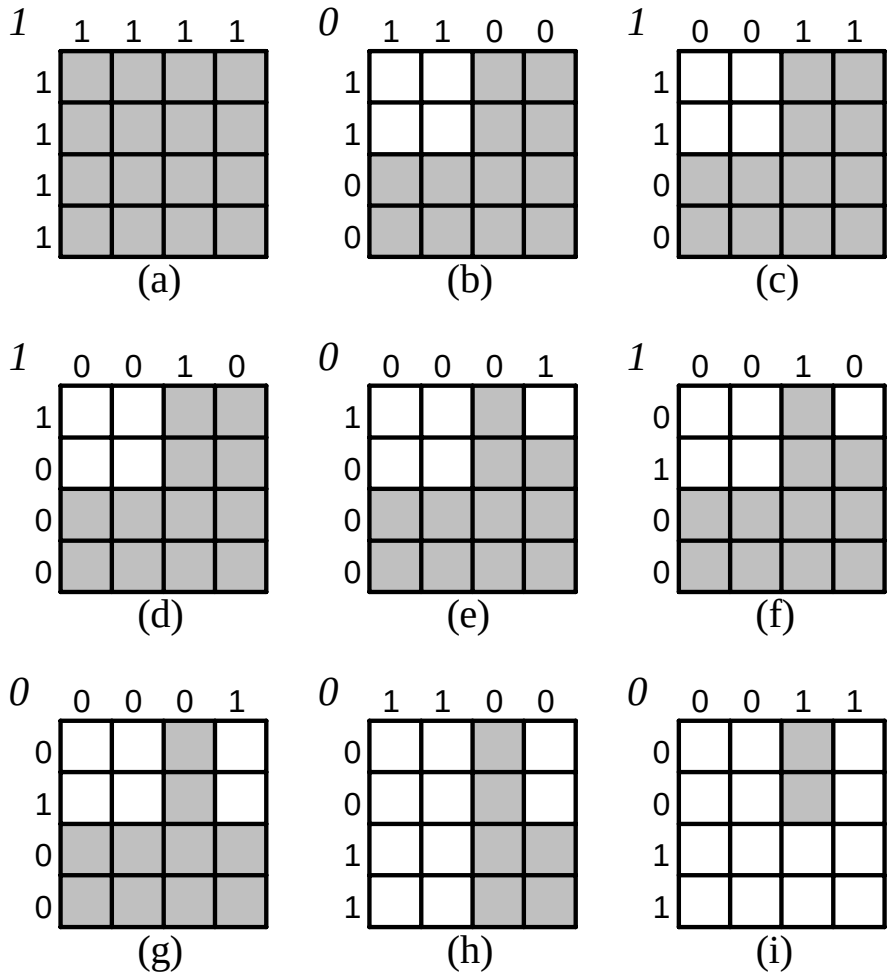


Figure 3.12: Example of decoding procedure of 1:4 tree code.

Table 3.3: States of automata in the decoding process shown in Figure 3.12.

1:4 tree code	1	0	1	1	0	1	0	0	0
upper(`s2`)	`W`	`A`	`B`	`B`	`B`	`B`	`B`	`C`	`D`
lower(`s1`)	`W`	`W`	`W`	`A`	`B`	`C`	`D`	`W`	`W`

3. Next bit of 1:4 tree code of "1" are stored to the upper-right 2×2 area, as shown in Figure 3.12(c). This bit means that there are at least one "1" bit in this area, and the decode procedure for 1×1 area in the upper-right 2×2 subarea are processed in order of upper-left, upper-right, lower-left, and lower-right, as shown in Figure 3.12(d), (e), (f), and (g).

4. The following two bits of "00" represent the lower-left and lower-right 2×2 area as all 0, and they are stored, respectively, as shown in Figure 3.12(h) and (i).

Note that in Figure 3.12, white squares and gray squares are the memory cells whose value is "0" and "1", respectively, and the values of "1" or "0" on the top side and the left side of the memory cell plain represents activated or negated state of column and row select lines, respectively. The value at the upper-left corner in each step is the 1:4 tree code according to each step.

The decoding process discussed above is implemented by making automata for each level, as shown Table 3.3. The upper automaton can select the area by the unit of 2×2 memory cells, and the lower automaton can by 1×1 memory cells. `W` represents the state of selecting all of its area, and `A`, `B`, `C` and `D` represents the states of selecting upper-left, upper-right, lower-left, and lower-right subarea, respectively. For example, the pair of two automata's states, (`s2`,`s1`)=(`B`,`A`) represents that the selected area is in upper-right 2×2 area by `s2`=`B`, and the precise position in this 2×2 area as the upper-left 1×1 memory area by `s1`=`A`, which is just corresponds to the state in Figure 3.12(f).

The selection lines of memory cells are implemented by using the `ra`, `rb`, `ca`, and `cb` signals defined in Table 3.4 as follows.

`r0` = `ra2` && `ra1`

Table 3.4: Truth table for select lines.

state	ra	rb	ca	cb
W	1	1	1	1
A	1	0	1	0
B	1	0	0	1
C	0	1	1	0
D	0	1	0	1

r1 = ra2 && rb1

r2 = rb2 && ra1

r3 = rb2 && rb1

c0 = ca2 && ca1

c1 = ca2 && cb1

c2 = cb2 && ca1

c3 = cb2 && cb1

Here the number following ca, and so on, represents the order of automata, for example ca2 represents the ca signal according to the state of automata s2, and && represents the logical-AND operator.

3.5 Summary and Conclusion

In this chapter, a concept of applying the tree structure for image signals is proposed. The image scan for tree structure including the nodes, which have logical-OR of lower nodes, can implement a kind of data compression by skipping the redundant scan step for white sub-areas. The size of sub-areas scanned can be changed dynamically according to the scanning step, the lower level of tree structure represents the higher spatial resolution. It is also derived that the quad tree structure is the most adequate to treat the image signals, which we call "1:4 tree structure." The 1:4 tree scan is more effective for some sample images than the raster scan, and than the run-length compression in most cases.

The decoding algorithm of 1:4 tree code is also discussed, ant it can be implemented by using the state automata and selection lines, with the array of memory cells, which can be selected simultaneously.