KMITL Sci. J. Vol. 5 No. 1 Feb. 2005

WFA Image Encoding with New Partitioning Method

Thommarat Lakhavijitlert[1] Korakot Prachumrak[2]

Department of Mathematics and Computer Science
Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang District,
Bangkok 10520, Thailand

ABSTRACT

Weight Finite Automata (WFA) image encoding is a method for encoding images which has been brought up by Culik and Kari [1]. They suggest the way to encode a digital image by applying quadtree partition to divide an image into subsquares and then construct subdividing images with linear combination as a weighted automaton. Nonatree is a new way to partition an image into a nine subsquare-tree. Instead of quadtree partition which is used by Culik and Kari, nonatree partition is applied with WFA to encode digital images.

KEYWORDS:
WFA, partition image, compression

1: E-mail: thommarat_tum@yahoo.com
2: E-mail: kpkorako@kmitl.ac.th

1. INTRODUCTION

The scope of this research is based on a weighted finite automaton (WFA). WFA is a tool for specifying digitized images. The image compression software based on this algorithm is competitive with other compression methods in the compression of typical grayscale images.

Culik and Kari suggested WFA method to encode an image by first partitioning an image into a set of level sub-images using quadtree-partition. Then, linear combination is applied to each sub-image in order to construct a labeled graph which is called a weighted finite automaton. This paper suggests a new method of encoding a digital image by combining WFA with a new method of image partitioning called nonatree partition.

This paper includes five more sections. First, the background part is the explanation of the general idea of the WFA encoding. Second part is the application of nonatree partition with WFA. Third, some examples of encoding and decoding are demonsrated here in this section. The last part of this paper concludes it and makes suggestion of the future work.

2. BACKGROUND

The WFA method was invented by K.Culik and J. Kari [2, 3, 5, 9, 10, 11] as a way of encode gray scale images. This method applies Weight Finite Automata (WFA) to compress images. The core idea of this method is to partition an image into quarters, then construct a weight finite automaton to suit them. Finally, the intensity of each pixel is kept in term of an automaton.

This background section explains some basic characteristics of WFA image encoding with the quad-tree partitioning used by Kurik and the other [4, 6, 7, 8, 12].

A finite-resolution image means a gray scale image which each pixel value is a real value between 0 and 2^n-1. Normally the value is between 0-255. **A multi-resolution image** defines 2^nx2^n resolution images, where n starts from 0, 1, 2, ... Each pixel at 2^nx2^n can be defined by a word of length n over the alphabet $\Sigma=\{0, 1, 2, 3\}$. Each letter of Σ is a quadrant of the partitioned image. Each word in Σ^* of length k is an address of a specific node of the quadtree at depth k [3].

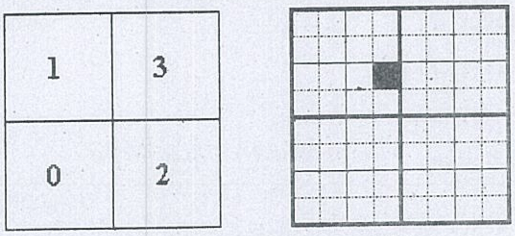
How to address the subsquares of Σ by words over Σ is shown here. Let $\Sigma=\{0, 1, 2, 3\}$, $w\in\Sigma^*$, $w=a_1a_2\ldots a_n$. w is the address of the subsquare of the original image where $a_i\in\Sigma$, $i = 1, 2, \ldots, n$. Since the typical way of partitioning images for WFA is a quadtree partition, the original image has the address ε and the address of the black square in fig. 1 (b) is 123.

3. WFA WITH NONATREE PARTITION

3.1 Nonatree Partition

Nanotree partition is a new way to partitioning method for WFA image encoding. That is, instead of partitioning the images into the basic quadtree partition the image is partitioned into 9 subsquares.

For Nonatree partition, **a finite-resolution image** is a gray scale image of the size



(a) (b)

Fig.1 (a) Quadtree partition and
(b) The black subsquare of the address 123

2	5	8
1	4	7
0	3	6

Fig.2 The addresses of the nonapartition sub-square.

3^nx3^n pixels. **A Multi-resolution image,** in this case, means a collection of 3^nx3^n resolution images for $n = 0, 1, \ldots, n$, where each pixel can be defined by a word of the length n over an alphabet $\Sigma=\{0,1,2,3,4,5,6,7,8\}$. The addresses of subsquare of nonapartition are shown in fig.2.

For an image w, its subimages have addresses w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8. Therefore, a multi-resolution image is a real function, that can be defined by $f: \Sigma^* \rightarrow \Re$ to be **an average preserving function** (ap-funciotn)

A function $f: \Sigma^* \rightarrow \Re$ is **average preserving function** (ap-function) if

$$f(w) = \frac{1}{|\Sigma|}\sum_{a\in\Sigma} f(wa)$$

for each $w\in\Sigma^*$, $\Sigma=\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ then

$$f(w) = \frac{1}{9}[f(w_0) + f(w_1) + f(w_2) + f(w_3) + f(w_4) + f(w_5) + f(w_6) + f(w_7) + f(w_8)] \quad (1)$$

An ap-function f can be represented by an finite labeled quadtree when the multi-resolution image size is 3^nx3^n resolution. Then the maximum value of labeled quadtree length is n (when subimage size is 1x1 pixel). The grayness value of the root (labeled ε) is $f(\varepsilon)$, its children grayness value is $f(w_0)$, $f(w_1)$, $f(w_2)$, $f(w_3)$, $f(w_4)$, $f(w_5)$, $f(w_6)$, $f(w_7)$, $f(w_8)$ etc.

The formal definition of a weighted finite automaton (WFA) consist of a 5-tuple $A = (Q, \Sigma\ f, I, F)$ where

1. Q is a finite set of *state*,
2. $\Sigma=\{0,1,2,3,4,5,6,7,8\}$ is a finite alphabet,
3. W_a: Qx$Q \rightarrow \Re$ is the *weight function*, for $W_a(p,q)$ the weights at edges labeled by a going out of node p to node q, for each $p,q\in Q$, and each $a\in\Sigma$,
4. $I: Q \rightarrow \Re$ is the *initial distribution*,
5. $F: Q \rightarrow \Re$ is the *final distribution.*

For $(p, a, q) \in Q$xΣxQ is a transition a from node p to node q of A iff $W_a(p, q) \neq 0$.

3.2 Encoding Algorithm

For the original image M which can be defined by an ap-function : $f: \Sigma^* \rightarrow R$

M_a define Subimage for address a

One has to construct the following vectors, in order to construct WFA:

$F(q)$ = the final distribution of state q

$I(q)$ = the initial distribution of state q

N = the index of the last state created

i = the index of the next processed state

$\gamma: Q \rightarrow \Sigma^*$ = a mapping of states to subsquares

Then follow the encoding flowchart shown in fig. 3 to construct a WFA.

3.3 Decoding Algorithm

The flowchart in fig. 4 shows how to decode a WFA for an image resolution 3^nx3^n

Input: A WFA specified by W_a, a=0, 1, 2, 3, 4, 5, 6, 7, 8, 9, I and F, and a non-negative integer n for image resolution 3^nx3^n

Output: The value $f(w)$ for all $w\in\Sigma^n$.

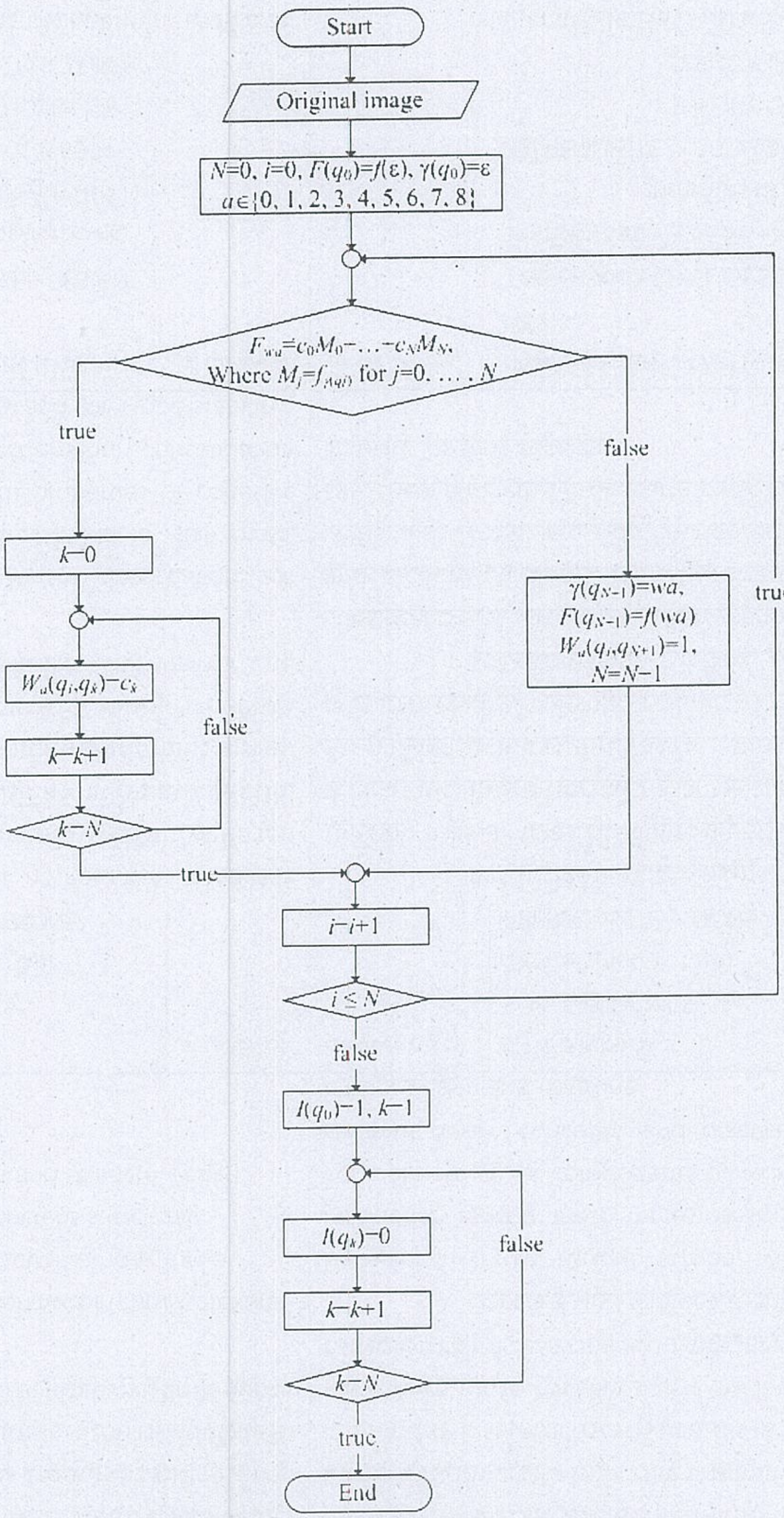


Fig. 3 Encoding Algorithm

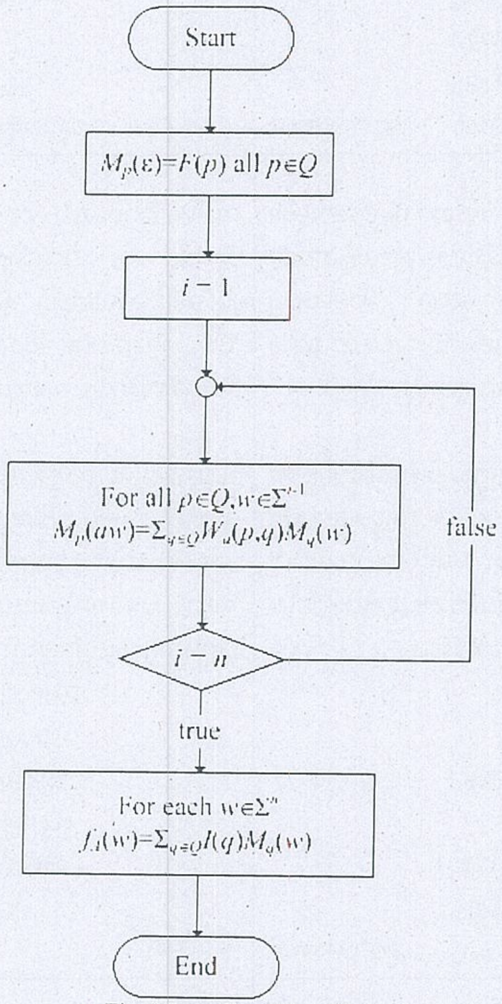


Fig.4 Decoding Algorithm

Variable:

- $I(p)$ = the initial distribution value of state p
- $M_p(w)$ = the multiresolution of subimage address w of state p
- $F(p)$ = the final distribution value of state p
- w = the address of the subimage
- $f_A(w)$ = the grayness value of the subimage address w
- $W_a(i,j)$ = the weight of label a from node i to node j
- Q = set of state of WFA
- Σ = {0, 1, 2, 3, 4, 5, 6, 7, 8}

4. WFA IMAGE ENCODING AND ENCODING

4.1.1 Encoding Example 1

This section shows how to encode an image by creating a weight finite automaton from an original image shown in fig. 5. To encode an image into an automata, one needs to follow these steps. First, set Initial distribution for each node. Then, set all the transitions for the automaton. Finally, set the final distribution for all the nodes.

First, state q_0 is assigned to the original image square, f_ε, whose its final distribution, $F(q_0)=f_\varepsilon$, is 1/9.

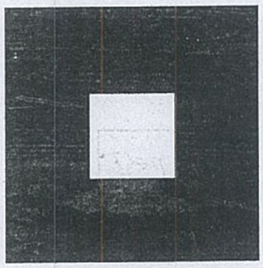


Fig.5 An example of a gray scale image.

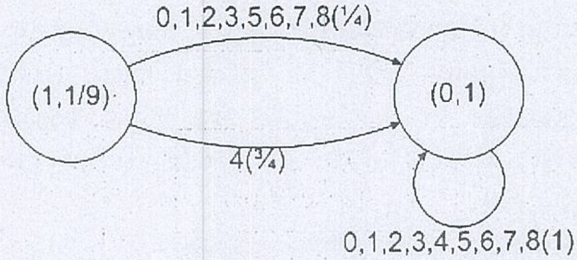


Fig.6 A WFA represents the multiresolution image in fig. 5.

Then, divide the original image into 9 subsquares (as in fig.2), 0, 1, 2, 3, 4, 5, 6, 7, 8.

The image in the first subsquare 0 can be expressed as $\frac{1}{4}\cdot f_\varepsilon$. This can be obtained by decreasing the contrast by a quarter from the original image f_ε, so that $W_0(q_0,q_1)=\frac{1}{4}$ and $F(q_1)=1$.

The image of the subsquare 1, 2, 3, 5, 6, 7, 8 is the same as the subsquare 0, so that $W_1(q_0,q_1) = \frac{1}{4}$, $W_2(q_0,q_1) = \frac{1}{4}$, $W_3(q_0,q_1) = \frac{1}{4}$, $W_5(q_0,q_1) = \frac{1}{4}$, $W_6(q_0,q_1) = \frac{1}{4}$, $W_7(q_0,q_1) = \frac{1}{4}$, $W_8(q_0,q_1) = \frac{1}{4}$.

The image of the subsquare 4 can be represented by state q_1 with a linear combination $\frac{3}{4}\cdot f_\varepsilon$, so that $W_4(q_0,q_1)= \frac{3}{4}$.

At state q_1, for all a = 0, 1, 2, 3, 4, 5, 6, 7, 8. $W_a(q_1,q_1)=1$ the grayness value is expressed by $1\cdot f(q_0)$ for all the subsquares, which results in the ap-WFA of fig. 6.

Finally, the initial distribution I is defined by $I(q_0)=1$ and $I(q_1)=0$. The weight finite automaton for fig.5 is shown in fig. 6.

4.1.2 Decoding Example 1

The multiresolution image can be constructed from a WFA as follows:

The grayness value or the weight can be gained by the multiplication of the initial distribution value of the first node, the final distribution value of the last node and all the transitions of the path. Then all the weight of the paths are added to all the corresponding pixels.

For example, find $f(w)$ if w is 04. This sub-image is the sum of 2 paths:

1. state q_0 to state q_1, the multiple product of the weight of this path is : $1\cdot\frac{1}{4}\cdot 1\cdot 1=\frac{1}{4}$.
2. state q_1 to state q_1, the multiple product of the weight of this path is : $0\cdot 1\cdot 1\cdot 1 = 0$

Then, the weight is found by the summation of 2 paths: $f(04)=\frac{1}{4}+0=\frac{1}{4}$.

This is the grayness value of subimage 04 for resolution 3^2x3^2.

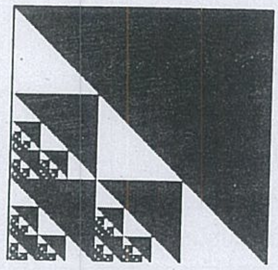


Fig.7 Another example of a digital image to be encoded.

4.2.1 Encoding Example 2

This section shows another sample of image encoding. From the original image in fig. 7, in order to encode the image into an automaton, the firststep is to set the initial distribution and the final distribution for all the nodes.

Then, divide the original image into 9 subsquare and set the addresses for them. The subsquares 0, 1, 3, have a linear combination $1 \cdot f_\varepsilon$, and are the same as the original image, so that $W_0(q_\varepsilon,q_\varepsilon)=1$, $W_1(q_\varepsilon,q_\varepsilon)=1$, $W_3(q_\varepsilon,q_\varepsilon)=1$.

The subsquare 2, 4, 6 can be expressed by the state q_0 with a linear combination $1 \cdot f_\varepsilon$ make a new state for state q_0 by define subsquare 2 to be a new state. Then, $W_2(q_\varepsilon,q_0)=1$, $W_4(q_\varepsilon,q_0)=1$, $W_6(q_\varepsilon,q_0)=1$.

Subsquares 5, 7, 8 have a grayness value of zero, which represents black, so the weight is zero. Its edge is not shown.

In state q_0, the grayness values of subsquare 2, 4, 6 are the same image as the state q_0, so that $W_2(q_0,q_0)=1$, $W_4(q_0,q_0)=1$, $W_6(q_0,q_0)=1$. The sub-square 0,1,3 can be represented by $1 \cdot f(q_0)$, so that $W_0(q_0,q_1)=1$, $W_1(q_0,q_1)=1$, $W_3(q_0,q_1)=1$.

At state q_1, for all a = 0, 1, 2, 3, 4, 5, 6, 7, 8. $W_a(q_1,q_1)=1$ as the grayness value is expressed by $1 \cdot f(q_1)$ for all the subsquares, so that this gives the ap-WFA of fig. 8.

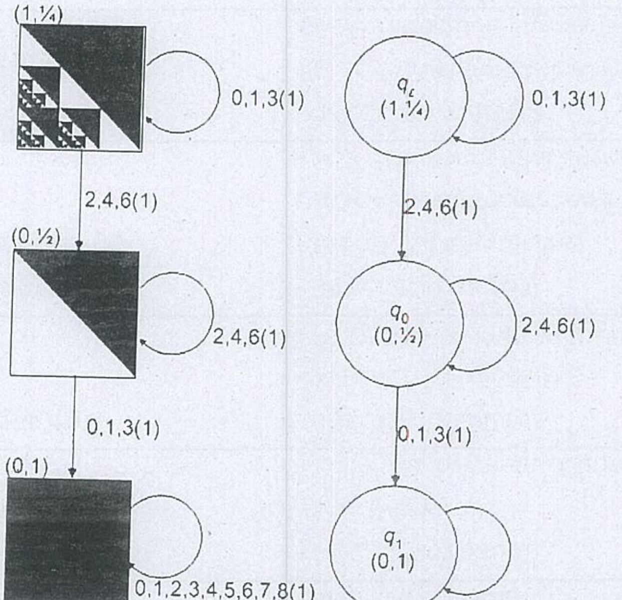


Fig.8. WFA of the original image fig. 7.

4.2.2 Decoding

The WFA over $\Sigma=\{0,1,2,3,4,5,6,7,8\}$ in fig.6 represents the original image in fig. 5. The straightforward way to decode the image of size 3^nx3^n from the WFA is to compute the multi-resolution $f(w)$ for all $w\in\Sigma^n$, (all paths of length n in the automaton). The decoding method is already shown in section 4.1.2.

5. CONCLUSIONS

A new way of encoding a digital image by applying nonatree partition to partitioning an original image into nine non-overlapped sub-images instead of quadtree partition is reported. By applying this alternative method of WFA, the size of the original image to be encoded is not limited to only 2^nx 2^n. Besides, in some images, this method needs fewer states than the quadtree partition which leads to smaller compression ratio, suggesting that future work can be done on the improvement of the entropy.

References

[1] Culik K. II and Kari J., **1933**, *Image Compre-ssion using Weighted Finite Automata*, Computer & Graphics Vol. 17, No. 3, pp. 305-313, Printed in Great Britain.

[2] Culik K. II and Kari J., **1995**, Finite state met-hods for compression and manipulation of images , *Data Compression Conference. DCC '95. Proceedings* , 28-30 March 1995 Pages:142 – 151.

[3] Culik K. II and Kari J.., **1995**, *Inferecce Algor-ithm for WFA and Image Compression*, Fractal Image Compression, Theory and Application, Fisher editor, Pages : 243-258, Springer-Verlag New York, Inc.

[4] U. Hafner, **1996**, Refining image Compression with weighted finite automata , *Data Co-mpression Conference. DCC '96. Procee-dings*, 31 March-3 April 1996 , Pages:359 – 368.

[5] Culik K. II and Kari J. and Valenta V., **1997**, Compres-sion of silhouette-like images based on WFA , *Data Compression Conference. DCC '97. Proceedings*, 25-27 March 1997 Pages:433.

[6] Z. Jiang and Litow B. and de Vel O. , **2001**, An inference implementation based on exten-ded weighted finite automata [for image compression], *Computer Science Confe-rence. ACSC 2001. Proceedings.* 24th Australasian , 29 Jan-4 Feb 2001, Pages:100 – 108.

[7] Y. sivasubramanyam and Kamala Krithivasan, **2001**, *Image Representation using distributed Weighted Finite Automata*, Published by Elsevier Science B.V..

[8] Yih-Kai lin and Hsu-Chun Yen, **2003**; "An ω-Automata Approach to the Representation of Bilevel Images", *IEEE.*

[9] Culik K. II and Peter von Rosenberg C., Generalized Weighted Finite automata Based Image Compression, *Department of Computer Science University of South Carolina Columbia*, S.C. 29208, U.S.A.

[10] Culik K., Valenta, V.; **1996**,Finite automata based compression of bi-level images,*Data Compression Conference, 1996. DCC '96. Proceedings , 31 March-3 April 1996 Pages:280 - 289*

[11] Culik K. II and Kari J., Finite State Transformations of Image, *Department of Computer Science University of South Carolina Columbia*, S.C. 29208, U.S.A.

[12] Katritzke, Techiques for WFA Construction, Pages:36-58.