Nona-Tree Weighted Finite Automata Compression of Simple Images

T. Lakhavijitlert, and K. Prachumrak

Abstract—We propose Nona-tree Weighted Finite Automata (NWFA) as a new approach to compress simple images. NWFA is used as a tool to the lossy compression of simple images which mean, in this case, bi-level and simple gray-scale images. An image is divided into multi-level of 9 partitions (nona-tree partition) and encoded into the form of an automaton. The NWFA compression ratios of most of the tested simple images are better than those of WFA with quad-tree partition.

Keywords—WFA, nona-tree partition, lossy compression, linear combination.

I. Introduction

THIS research proposes Nona-tree Weighted Finite Automata (NWFA) as a new way to compress simple images, bi-level[1,2] and simple gray-scale images[3], in term of lossy compression. Although based on weighted finite automata (WFA) compression [4], NWFA shows better compression ratio on most of the tested images.

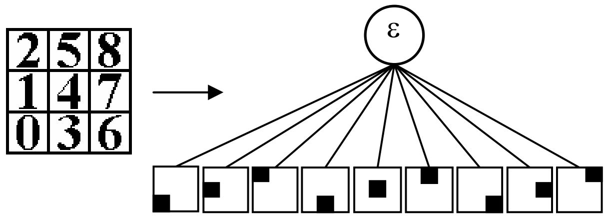
WFA is one of the techniques which have been used to compress digital images [5,6]. WFA represents an image in term of a weighted finite automaton with a very good compression ratio. WFA is based on the idea of fractal that an image has self-similarity in itself. In this case, the self-similarity is sought from the symmetry of an image, so the encoding algorithm divides an image into multi-levels of quad-tree segmentations and creates an automaton from the sub-images [7]. The idea of this research is that not all images are symmetry of the 4 quadrants. We then propose another way of partitioning images that should benefit images in general which are not always symmetric in four.

In next section, we describe the notations used in NWFA. The encoding and decoding algorithms are explained in section III. Section IV shows how to apply the NWFA algorithms to simple images. In the last section, we discuss the advantages and limitations of NWFA, and also implications for further research.

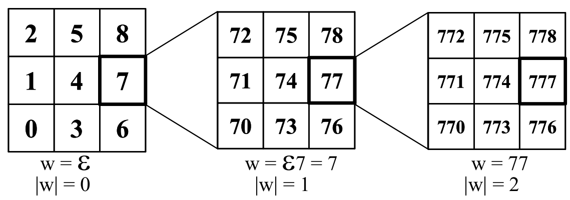
II. Nona-Tree Weighted Finite Automata

A. Preliminaries

A finite-resolution image is an image of the size 3^mx3^m pixels. A Multi-resolution image means a collection of 3^nx3^n resolution image for n= 0, 1, …, n, where each pixel can be defined by a word of the length n over an alphabet Σ={0, 1, 2, 3, 4, 5, 6, 7, 8}. The addresses of the subsquares of nona-tree partitions are shown in Fig. 1a.



a. Positions of the 9 subimages



b. Some example addresses of the sub-image

Fig. 1 The positions of the sub-images

For an image w, the addresses of the sub-images are w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, Fig 1b. A multi-resolution image is a real function that can be defined by $f : \Sigma^* \rightarrow \Re$ to be an average preserving function (ap-function), if

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

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

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

An ap-function f can be represented by a finite labeled nona-tree where the multi-resolution image size is 3^nx3^n. The maximum address length is n when the sub-image becomes the size of 1x1 pixel.

Authors are with King Mongkut's Institute of Technology, Ladkrabang, Bangkok 10520, Thailand

The intensity of the root (labeled ε) is $f(\varepsilon)$, its children intensities are $f(w_0)$, $f(w_1)$, $f(w_2)$, $f(w_3)$, $f(w_4)$, $f(w_5)$, $f(w_6)$, $f(w_7)$, $f(w_8)$ etc.

The formal definition of a nona-tree weighted finite automaton (NWFA) consists of a 5-tuple A=(Q, Σ, W_a, I, F) where:

1. Q is a finite set of *state*,
2. Σ={0, 1, 2, 3, 4, 5, 6, 7, 8} is a finite alphabet,
3. W_a:QxQ→ℜ 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→ℜ is the *initial distribution*,
5. F:Q→ℜ is the *final distribution*.

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

III. NWFA Encoding Algorithm

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

M_a defines sub-image of address a

To construct NWFA, we have to construct the following vectors [8]:

$F(q)$ = 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
$\gamma : Q\to\Sigma^*$

input : Image M has the size of $3^k\times3^k$
output : NWFA M which represents an image M

1. Set $N = 0$, $i = 0$, $F(q_0) = f(\varepsilon)$, $\gamma(q_0) = \varepsilon$
2. process q_i, that is, for $w=\gamma(q_i)$ and each $a\in\{0,1,2,3,4,5,6,7,8\}$
   a) if there are $c_0, c_1, \dots, c_N$ such that

   $$f_{wa}=c_0M_0 + c_1M_1 + \dots + c_NM_N$$

   where $M_j = f_{\gamma(qj)}$ for $j = 0,1,\dots, N$ then set $W_a(q_i,q_j)=c_j$ for $j=0,1,\dots,N$
   b) otherwise set $\gamma(q_{N+1}) = wa$, $F(q_{N+1})= f_{avg}(wa)$ and $W_a(q_i, q_{N+1}) = 1$, $N = N + 1$
3. set $i = i + 1$, if $i \le N$, then go to 2.
4. set $I(q_0) = 1$, $I(q_i) = 0$ for $j = 1, \dots, N$ where I is the initial distribution of M

This encoding algorithm uses sub-images of the image M to create the linear combination in step 2. Each sub-image represents a node of the weighted finite automata. Give ϕ_i as the brightness value of the sub-images where i = 0, 1, … , *8* then

$$\phi_i= \frac{1}{9}\,[\phi_i(0) + \phi_i(1) + \phi_i(2) + \phi_i(3) + \phi_i(4) + \phi_i(5) + \phi_i(6) + \phi_i(7) + \phi_i(8)] \tag{2}$$

Equation (3) can be expressed in term of linear combination [2,9] as follows:

$$f_{wa} = c_0\phi_0 + c_1\phi_1 + \dots + c_N\phi_N \tag{3}$$

from (2), we can distribute it to be (4)

$$\begin{aligned}
f(w0) &= c_0\phi_0(0)+c_1\phi_1(0)+\dots+c_N\phi_N(0)\\
f(w1) &= c_0\phi_0(1)+c_1\phi_1(1)+\dots\dots+c_N\phi_N(1)\\
f(w2) &= c_0\phi_0(2)+c_1\phi_1(2)+\dots+c_N\phi_N(2)\\
&\cdots\\
f(w6) &= c_0\phi_0(6)+c_1\phi_1(6)+\dots+c_N\phi_N(6)\\
f(w7) &= c_0\phi_0(7)+c_1\phi_1(7)+\dots+c_N\phi_N(7)\\
f(w8) &= c_0\phi_0(8)+c_1\phi_1(8)+\dots+c_N\phi_N(8)
\end{aligned} \tag{4}$$

Equation (4) can be written in the matrix form [5,10]:

$$[f]=[\phi][C] \tag{5}$$

where [ϕ] is the matrix size $9^k\times(n+1)$
[C] is the matrix size $(n+1)$x1 of coefficient
[f] is the matrix size $9^k\times1$ of sub-image brightness value of image f_{wa}.

IV. NWFA Decoding Algorithm

This algorithm shows how to decode an NWFA for an image resolution $3^n\times3^n$.

Input : An NWFA specified by W_a, a=0, 1, 2, 3, 4, 5, 6, 7, 8, I and F, and a non-negative integer n for an image resolution $3^n\times3^n$.

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

1. Set $\psi_p(\varepsilon) = F(p)$ for all $p\in Q$
2. Do the following step 3 for all $i = 1, 2, \dots, n$
3. For all $p\in Q$, $w\in\Sigma^{i-1}$ and $a\in\Sigma$ compute
$$\psi_p(aw)=\sum_{q\in Q}W_a(p,q)\psi_q(w)$$
4. For each $w\in\Sigma^n$ compute
$$f_M(w)=\sum_{q\in Q}I(q)\psi_q(w)$$

Next we analyze the complexity of the decoding algorithm. Denote m as a number of nodes. The time complexities of step 1, 3 and 4 is $O(m)$, $O(mm\cdot|\Sigma|^{i-1}\cdot|\Sigma|)= O(m^2\cdot|\Sigma|^i)$ and $O(m\cdot|\Sigma|^n)$, respectively. Step 3 is repeated for i=1, 2, … , n so that the complexity of steps 2 and 3 is $O(m(|\Sigma|+|\Sigma|^2+|\Sigma|^3+\dots+|\Sigma|^n)) = O(m^2\cdot|\Sigma|^n)$ then we can compute the time complexity from $O(m+m^2\cdot|\Sigma|\cdot n+m\cdot|\Sigma|\cdot n)=O(m^2\cdot|\Sigma|^n)$

V. NWFA Image Encoding and Decoding

A. Encoding Example I

This section shows intuitively how to build an NWFA from an image "*ribbon*" as in Fig. 2. The encoding algorithm is already described in section III. From this image, we explain how to construct an NWFA S over Σ ={0, 1, 2, 3, 4, 5, 6, 7, 8}. NWFA S is drawn as a directed graph on which each path shows the label of the addresses and the weight of the intensity. The node of the graph represents the states of

NWFA *S*. The edges represent non-zero elements of the transition. If $W_a(i,j) = r \neq 0$, there is an edge in the graph from node *i* to node *j* with label *a* and weight *r*. Weights are shown in parentheses. To simplify the graph, multiple edges with the same weight but several labels are drawn as a single edge.

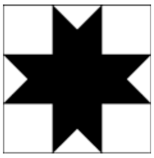


Fig. 2 The image "Ribbon"

The image to encode has the size of 243x243 pixels. This bi-level image has the values of 0 for black and 1 for white.

In the first step, we define the image Ribbon in Fig. 2. to encode to be q_0 and $F(q_0)$=1275/9. Then, consider its sub-images the nine sub-squares 0 to 8 which have the address length 1. The sub-image f_0 and f_1 can not be expressed as a linear combination of f_ε, where f_w is the brightness of sub-image address $w \in \Sigma^*$.

Next, we apply step 2b of the encoding algorithm. Define $W_0(q_0,q_1)=1$ and $F(q_1)=255$; $W_1(q_0,q_2)=1$ and $F(q_2)=576/9$ as; the sub-image of address 2 is the same as the node q_1, it can be expressed as $f_2=1\cdot f_1$, and define $W_2(q_0,q_1)=1$. Then, consider that the sub-image of address 3 can not be expressed as a linear combination of f_ε so, we define a new node which $W_3(q_0,q_3)=1$ and $F(q_3)=576/9$. The brightness value of sub-image f_4 is 0, and from (4) when the coefficients are 0, the new paths are not created. For sub-image f_5, this can not be expressed as a linear combination from any node, so we create a new node and define $W_5(q_0,q_4)$ and $F(q_4)=576/9$. Sub-image 6 is the same as sub-image f_0 , we define $W_6(q_0,q_1)=1$. Sub-image 7 can not be expressed as a linear combination of any node, then define $W_7(q_0,q_5)=1$ and $F(q_5)=576/9$.

For the sub-images of the address length 2, consider node q_1, all the sub-images are the same as q_1 itself. We can create 8 paths for q_1 to q_1 with label 0, 1, 2, … , 8 while each weight is 1. The next node is q_2, we consider all sub-images of q_2, for sub-image address 0, it can not be expressed as a linear combination of any node, so we create a new node which has $W_0(q_2,q_6)=1$ and $F(q_6)=128$; sub-image 1 can be expressed as a linear combination $f_{11}=f_0$, then define $W_1(q_2,q_1)=1$. The sub-image 2 of q_2 can not be expressed as a linear combination of any node, then define $W_2(q_2,q_7)=1$ and $F(q_7)=128$. The sub-image 4 of q_2 is the same as q_2 itself, thus we can define $W_4(q_2,q_2)=1$. Sub-image 3 and the rest of sub-images of q_2 are black squares, thus the brightness is 0 and also the coefficients of the linear combination are 0s. In this case, the paths are not created. Nodes q_3, q_4 and q_5, are the same as q_2. Then, consider each of their sub-images. If it can not be expressed as a linear combination then create it as a new node.

For the next step, all nodes with the depth of 3, the sub-images of the address length 2. For the first node is q_6, the sub-images 0, 4 and 8 of q_6, are the same as q_6, then create $W_0(q_6,q_6)=1$, $W_4(q_6,q_6)=1$ and $W_8(q_6,q_6)=1$. The sub-image 1, 2 and 5 are the same as q_1 which creates $W_1(q_6,q_1)=1$, $W_2(q_6,q_1)=1$ and $W_5(q_6,q_1)=1$. Another node is a black square that does not need to create a path to any node. The nodes q_7, q_8 and q_9 have the same sub-images as q_6. The rest sub-images create the new nodes and paths the same way as done in q_6.

Finally, set the initial value for the node q_0 as $I(q_0)=1$. For the other nodes, the initial values are set to be all 0s. The completed NWFA which represents the image in Fig. 2 is shown in Fig. 3.

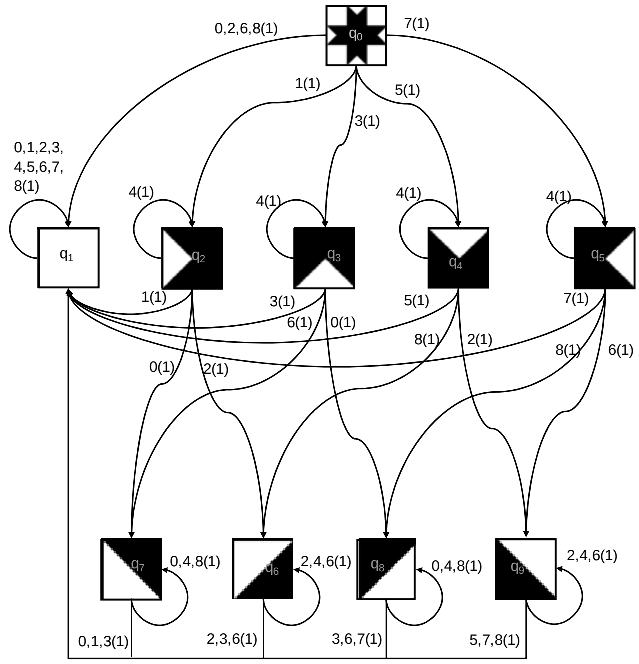


Fig. 3 The NWFA represents the image "Ribbon" in Fig. 2

B. *Decoding Example I*

This section shows how to decode the NWFA in Fig. 3 into the original image. An NWFA can be specified as a diagram with *n* nodes $\{1,\ldots,n\}$. There is an edge from node *i* to node *j* with label $a \in \Sigma$ and weight $r \neq 0$ iff $W_a(q_i,q_j)=r$.

The multi-resolution image can be read from a diagram as follows: for the corresponding pixels, find all paths of the corresponding addresses, and multiply the weights with the initial and final distribution. Thus, $f_A(w)$ is the sum of the weights of all paths whose labels form the word *w*.

For example, the brightness of the sub-image address 142 represented by $f(142)$ can be computed as follows:

$f(142) = I(q_0) \cdot W_1(q_0,q_2) \cdot W_4(q_2,q_2) \cdot W_2(q_2,q_7) \cdot F(q_7)$

$= 1 \cdot 1 \cdot 1 \cdot 128 = 128.$

If in any node there is no path links to the others, then the weight becomes 0 and the final distribution value is 0, for example, $f(50) = 1 \cdot 1 \cdot 0 \cdot 0 = 0$.

C. *Encoding Example II*

This section shows how to encode an image "snow" from Fig. 4. Start by creating the state q_0 (the parent node) that represents the image f_ε . We can process q_0 by expressing all nona-tree of f_ε as the linear combination of the existing states or as a new state. The sub-image of the address length 1 in Fig. 4 can be expressed as $0 \cdot f_\varepsilon$ for the addresses 0, 2, 6 and 8, and the new states q_1, q_2, q_3, q_4 and q_5 represent sub-images f_1, f_3, f_4, f_5 and f_7, respectively.

Then, for the address length 2, first, consider the node q_1; the sub-images 0, 1, 2, 3 and 5 are empty and the sub-images 4, 6, 7 and 8 are white which we create as a new state q_6. We can create an edge from q_1 to q_6 with the weight 1 and labels 4, 6, 7 and 8. The sub-images of the ap-functions f_3, f_4, f_5 and f_7 are the same as the sub-images f_1 that they are either black images

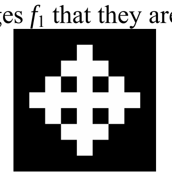


Fig. 4. "*snow*" image for example II

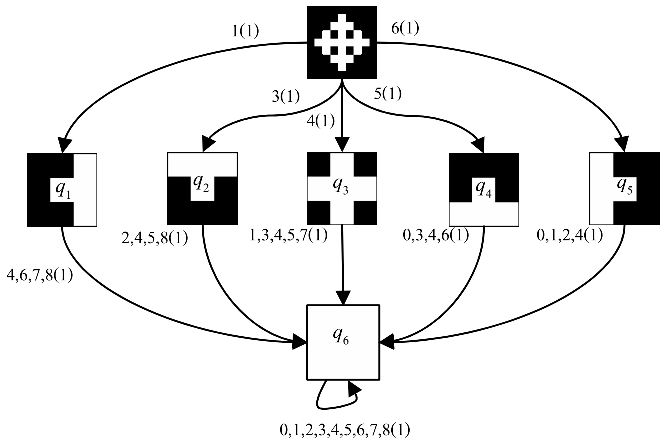


Fig. 5 The NWFA represents the image "*snow*"

or white images. For the sub-images of the white square, the paths are joined to the node q_6.

Following the same steps as shown in the last section, we can create an NWFA *S* to represent the image *"snow"* in Fig.4. The initial distribution at state q_ε is 1 and the other states are 0. The final distribution at each state is the average intensity of the image of the state.

D. Decoding Example II

In this section, the image to decode has the size of $3^k \times 3^k$ pixels and each pixel has the address of *k* length or its depth is *k*. The brightness of each sub-image of the depth *n* can be computed by the method in the decoding example I or by step 3 and 4 in decode algorithm which also can be written in the form of the following equation:

$$x_p(aw) = \sum_{q\in Q}\sum_{a\in\Sigma}\sum_{p\in Q} W_a(p,q)\,x_q(w) \quad for\,all\ p\in Q$$

$$\text{and} \qquad f(w) = \sum_{p\in Q} I(p)\,x_q(w) \qquad (5)$$

where $f(w)$ is the brightness value of the pixel address *w* and $|w| = n$ and $x_p(\varepsilon)=F(p)$.

VI. Conclusion

We have studied how to represent simple images by using NWFA. This method is a lossless compression. Fig.6 shows the WFA and NWFA compression ratio on some simple images. From these results, we can conclude that dividing the

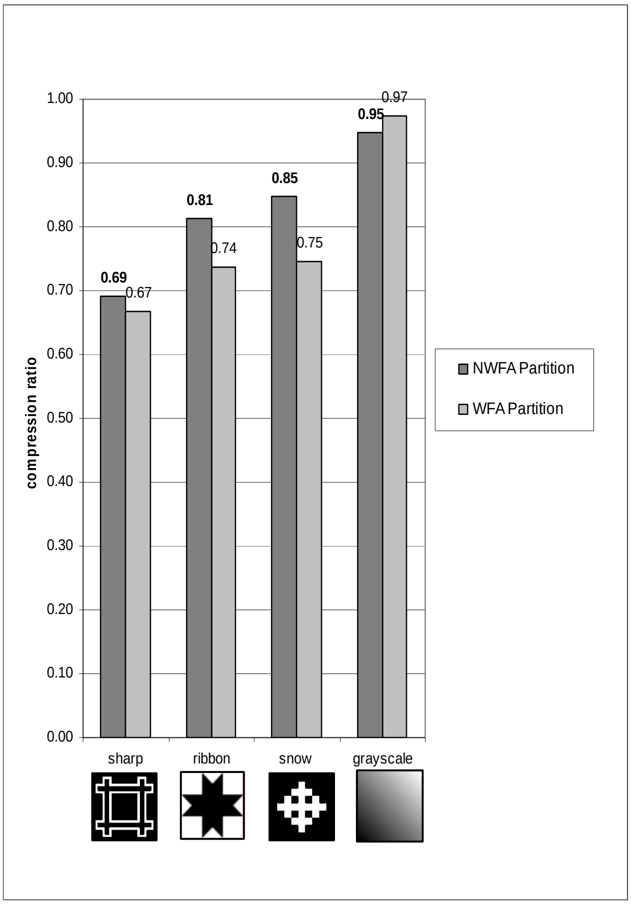


Fig. 6 The compression ratios of the NWFA and WFA of some images

original image into more nodes (nona-tree) from the beginning can reduce the levels of division in the automata, because it has higher chance of finding the self-similarity in the sub-images of the same level. This leads to the reduction of the number of nodes as a whole. From this conclusion, our future work will be on to study many ways of partitioning an image for the better compression ratio.

References

[1] K. Culik, J. Kari, and V. Valenta, "Compression of Silhouette-like images based on WFA", Data Compression Conference, 1997.

[2] Yih-Kai Lin and Hsu-Chun Yen, "An ω-Automata Approach to the Representation of Bilevel Images", IEEE Transaction on Systems, Man, and Cybermetics-Part B, vol 33, 2003, June 2003, pp. 524-531.

[3] Kamala Krithivasan, "Weighted Finite Automata and Representation of Image", Theoretical Computer Science Lab Department of Computer Science and Engineering Indian Institute of Technology, Madras. Lecture Note.

[4] K. Culik II and J. Kari, “Image Compression Using Weighted Finite Automata”. Computers and Graphics, vol. 17, no. 3, May/June 1993, pp. 305-313.

[5] U. Hafner, Lehrstuhl fur Inf., Wurzburg Univ., Germany "Refining Image Compression with Weighted Finite Automata",presented at the IEEE Data compression Conference, March 1996 , pp. 359-368.

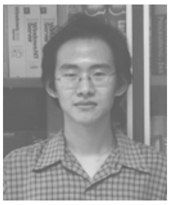
[6] F. Katritzke, "Refinements of Data Compression Using Weighted Finite Automata", Ph.D. dissertation, graph. Darst. - Siegen, Univ., Diss., 2001.

[7] F.Katritzke, W. Merzenich, M. Thomas, "Enhancements of partitioning techniques for image compression using weighted fintie automata", *Elsevier*, 2003.

[8] Z. Jiang, B. Litow, Olivier de Vel, "An Inference Implementation Based on Extended Weighted Finite Automata", *Proceeding of Australian Computer Science Conference,* vol. 23, 2003, pp. 100-108.

[9] Mathieu Giraud and Dominique Lavenier, “Linear Encoding Scheme for Weighted Finite Automata”, M. Domaratzki et al. (Eds.): CIAA 2004, LNCS 3317, pp. 146–155, 2005.

[10] S. Mallat and Z. Zhang, “Matching Pursuit With Time-Frequency dictionaries” , IEEE Transactions on Signal Processing, Vol. 49, No. 3, March 2001 , pp. 507-510..



Thommarat Lakhavijitlert was born on 8 December 1979 in Thailand. He received his B.Sc. in Applied Mathematics at King Mongkut’s Institute of Technology North Bangkok, Thailand, in 2002. He is currently a master degree student in the department of Mathematics and Computer Science, King Mongkut’s Institute of Technology Ladkrabang, Thailand. His research interests include image compression and computer graphics.

Korakot Prachumrak received a B.Eng in Computer Engineering from Kasetsart University, Thailand, 1995 an M.Sc. in Computer Graphics and Virtual Environments from The University of Hull. U.K., 1997 and a Ph.D in Engineering from Kagoshima University, Japan, 2003.

She is currently an Assistant Professor in the Department of Mathematics and Computer Science, King Mongkut’s Institute of Technology, Ladkrabang, Thailand. Her research interests include computer graphics, fractal theory and image processing.