Pergamon

Comput. & Graphics, Vol. 20, No. 1, pp. 125–135, 1996

0097–8493/96 $15.00 + 0.00

0097–8493(95)00072–0

Technical Section

FINITE STATE TRANSFORMATION OF IMAGES

KAREL CULIK II[1†] and JARKKO KARI[2]

[1]Department of Computer Science, University of South Carolina, Columbia, SC 29208, U.S.A. and
[2]Mathematics Department, University of Turku, 20500 Turku, Finland

Abstract—Weighted finite automata (WFA) have been introduced as devices for computing real functions on $[0,1]^n$. The main motivation has been to generate functions on $[0,1]\times[0,1]$ interpreted as gray-tone images. Weighted finite transducers (WFT) are finite state devices that serve as a powerful tool for describing and implementing a large variety of image transformations and more generally linear operators on real functions. Here we show new results on WFT and demonstrate that WFT are indeed an excellent tool for image manipulation and more generally for function transformation. We note that every WFA transformation is a linear operator and show that most of the interesting linear operators on real functions (on $[0,1]^2$) can be easily implemented by WFT. We give a number of examples that include affine transformations, a low-pass filter, wavelet transform, (partial) derivatives, simple and multiple integrals. Since the family of WFA-functions is constructively closed under WFT, each of our examples is actually a proof of a theorem stating that for each WFA A there effectively exists another WFA B that computes the integral (or other transformations) of the function defined by A.

1. INTRODUCTION

Weighted finite automata (WFA) have been introduced in refs [1, 2] as devices for computing real functions on $[0,1]^n$. The main motivation has been to generate functions on $[0,1]\times[0,1]$ interpreted as gray-tone images. In refs [2, 3] (see also [4]) we developed inference algorithms for WFA. Using the algorithm from ref. [3] we can efficiently encode any image (digitalized photograph) by a WFA. One of the best image compression method has been developed on this basis. In ref. [5] the generalized k-tape WFA have been introduced and in particular, the 2tape WFA called *weighted finite transducers* (WFT) have been studied and shown to perform a number of simple operations on images. When considered as only defining mappings on finite words WFT are a special case of rational transducers [6]. In ref. [7] the iterative WFT have been studied and shown to be strictly more powerful image generators than the mutual recursive function systems (MRFS), see *e.g.* ref. [8].

Here we show new results on WFT and demonstrate that WFT are indeed an excellent tool for image manipulation and more generally for function transformation. We note that every WFA transformation is a linear operator and show that most of the interesting linear operators on real functions (on $[0,1]^2$) can be easily implemented by the WFT. We give a number of examples that include affine transformations, a low-pass filter, wavelet transform, (partial) derivatives, simple and multiple integrals. Since the family of WFA-functions is constructively closed under WFT [5], each of our examples is actually a proof of a theorem stating that for each WFA A there effectively exists another WFA B that computes the integral (or other transformations) of the function defined by A. In the case of integrals this has been shown in ref. [1] for average preserving WFA. We extend this result to arbitrary WFA and to multiple integrals.

In Section 2 we introduce an automata-theoretic notation for multiresolution images and weighted finite automata and transducers. In Section 3 we introduce average preserving WFT and show that average preserving WFA are closed under average preserving WFT. This is important since an average preserving WFA generates an average preserving function that is a well-defined multiresolution image. We show that average preserving WFT are closed under the operations of composition, addition and multiplication by a constant.

In Section 4 we demonstrate the surprising ability and flexibility of WFT to implement image transformations and linear function operators. We show that every piecewise affine transformation can be implemented by WFT. The crucial, somewhat tricky example, of a WFT is the one that shifts an image by one pixel for any finite resolution. It leads us to implementation of filters, (partial) derivatives and other linear operators.

2. MULTIRESOLUTION FUNCTIONS, WFA AND WFT

A *multiresolution function* f over alphabet Σ is a function $\Sigma^* \to \mathbb{R}$. It is *average preserving* if

$$\sum_{a\in\Sigma} f(ua) = p\cdot f(u), \quad \text{for all } u \in \Sigma^*, \qquad (1)$$

where $p = |\Sigma|$ is the cardinality of the alphabet Σ.

† Author for correspondence.

In image processing applications a four letter alphabet $\Sigma=\{0, 1, 2, 3\}$ will be used. Words of length k over Σ will be interpreted as addresses of subsquares in the division of the unit square into $2^k \times 2^k$ subsquares as follows: each letter refers to one quadrant as shown in Fig. 1. Word wa addresses the quadrant a of the subsquare addressed by w. A multiresolution function $f:\Sigma^* \to [0,1]$ defines a sequence of gray-tone images with increasing resolutions: its restriction to Σ^k defines an image in resolution $2^k \times 2^k$. The gray-tone intensity of a point in the subsquare addressed by $w \in \Sigma^k$ is $f(w)$. The images at different resolutions are compatible if the multiresolution function is average preserving. In this case one can easily move from a higher resolution to a lower one by simply computing the averages of the intensities inside each subsquare.

A multiresolution function f over $\Sigma=\{0, 1, 2, 3\}$ defines an *infinite resolution image* $\hat{f}:[0,1]^2 \to [0,1]$ if the sequence $f_{|\Sigma k}$, $k=0, 1, 2, \ldots$ of finite resolution images converges point-wise to $\hat{f}$. In a similar fashion, WFA over 2^m letter alphabet are used to define functions $[0,1]^m \to [0,1]$ for all $m \geqslant 1$.

An m-state *weighted finite automaton (WFA) A* over alphabet Σ is defined by a row vector $I^A \in \mathbb{R}^{1\times m}$ (called the initial distribution), a column vector $F^A \in \mathbb{R}^{m\times 1}$ (the final distribution), and weight matrices $W_a^A \in \mathbb{R}^{m\times m}$ for all $a \in \Sigma$. The WFA A defines a multiresolution function f_A over Σ by:

$$f_A(a_1 a_2 \ldots a_k) = I^A \cdot W_{a_1}^A \cdot W_{a_2}^A \cdot \ldots \cdot W_{a_k}^A \cdot F^A.$$

The WFA A is called average preserving if:

$$\sum_{a\in\Sigma} W_a^A \cdot F^A = p \cdot F^a, \qquad (2)$$

where $p=|\Sigma|$ is the cardinality of the alphabet Σ. In other words, a WFA is average preserving if its final distribution is an eigenvector of $\Sigma_{a\in\Sigma} W_a^A$ corresponding to its eigenvalue $|\Sigma|$. It is known (see ref. [2]) that the multiresolution function computed by an average preserving WFA is average preserving, and that every average preserving multiresolution function computable by a WFA can be computed by an average preserving WFA.

Analogously to WFA, an n-state *weighted finite transducer* (WFT) M from alphabet Σ_1 into alphabet Σ_2 is specified by:

1	3
0	2

Fig. 1. The addresses of quadrants.

1. Weight matrices $W_{a,b} \in \mathbb{R}^{n\times n}$ for all $a \in \Sigma_1 \cup \{\varepsilon\}$ and $b \in \Sigma_2 \cup \{\varepsilon\}$.
2. A row vector $I \in \mathbb{R}^{1\times n}$, called the initial distribution.
3. A column vector $F \in \mathbb{R}^{n\times 1}$, called the final distribution.

The WFT M is called ε-free if weight matrices $W_{\varepsilon,\varepsilon}$, $W_{a,\varepsilon}$ and $W_{\varepsilon,b}$ are zero matrices for all $a \in \Sigma_1$ and $b \in \Sigma_2$.

The WFT M defines function $f_M:\Sigma_1^* \times \Sigma_2^* \to \mathbb{R}$, called weighted relation between Σ_1^* and Σ_2^*, by:

$$f_M(u, v) = I \cdot W_{u,v} \cdot F, \quad \text{for all } u \in \Sigma_1^*,\ v \in \Sigma_2^*,$$

where

$$W_{u,v} = \sum_{\substack{a_1\ldots a_k=u \\ b_1\ldots b_k=v}} W_{a_1,b_1} \cdot W_{a_2,b_2} \cdot \ldots \cdot W_{a_k,b_k}, \qquad (3)$$

if the sum converges. [If the sum does not converge, $f_M(u, v)$ remains undefined.] In (3) the sum is taken over all decompositions of u and v into symbols $a_i \in \Sigma_1 \cup \{\varepsilon\}$ and $b_i \in \Sigma_2 \cup \{\varepsilon\}$, resepectively.

In the special case of ε-free transducers:

$$f_M(a_1 a_2 \ldots a_k,\ b_1 b_2 \ldots b_k) = I \cdot W_{a_1,b_1} \cdot W_{a_2,b_2} \cdot \ldots \cdot W_{a_k,b_k} \cdot F,$$

for $a_1 a_2 \ldots a_k \in \Sigma_1^k$, $b_1 b_2 \ldots b_k \in \Sigma_2^k$ and $f_M(u, v)=0$, if $|u| \neq |v|$.

Let $\rho:\Sigma_1^* \times \Sigma_2^* : \to \mathbb{R}$ be a weighted relation and $f:\Sigma_1^* \to \mathbb{R}$ a multiresolution function. The application of ρ to f is the multiresolution function $g=\rho(f):\Sigma_2^* \to \mathbb{R}$ over Σ_2 defined by:

$$g(v) = \sum_{u\in\Sigma_1^*} f(u)\rho(u, v), \quad \text{for all } v \in \Sigma_2^*,$$

provided the sum converges. The application $M(f)$ of WFT M to f is defined as the application of the weighted relation f_M to f, *i.e.* $M(f)=f_M(f)$.

If $\Sigma_1=\Sigma_2=\{0, 1, 2, 3\}$ the weighted relation ρ can be applied also on (integrable) infinite resolution images $\alpha:[0, 1)^2 \to [0, 1]$. Assume there exists an (unique) average preserving multiresolution function f such that $\hat{f}=\alpha$. We define $\rho(\alpha) = \widehat{\rho(f)}$, provided $\rho(f)$ exists and converges to an infinite resolution image $\widehat{\rho(f)}$. The application of a WFT M to α is defined as the application of f_M to α.

Lemma 1—WFT M is a linear operator $\mathbb{R}^{\Sigma_1^*} \to \mathbb{R}^{\Sigma_2^*}$. In other words:

$$M(r_1 f_1 + r_2 f_2) = r_1 M(f_1) + r_2 M(f_2),$$

for all $r_1, r_2 \in \mathbb{R}$ and $f_1, f_2:\Sigma_1^* \to \mathbb{R}$. More generally, any weighted relation acts as a linear operator. □

It follows naturally from Lemma 1 that weighted relations over four letter alphabet act as linear operators of infinite resolution images.

In ref. [5] the application of WFT M to WFA A was defined. For simplicity we define it here only for ε-free WFT. The application of an ε-free WFT M to an m-state WFA A over alphabet Σ_1 defined by initial distribution I^A, final distribution F^A and weight matrices W_a^A, $a\in\Sigma_1$, is the mn-state WFA $B=M(A)$ over alphabet Σ_2 with initial distribution $I^B=I\otimes I^A$, final distribution $F^B=F\otimes F^A$ and weight matrices:

$$W_b^B=\sum_{a\in\Sigma_1} W_{a,b}\otimes W_a^A \quad \text{for all } b\in\Sigma_2 .$$

Here, $\otimes$ denotes the ordinary tensor product of matrices (called also Kronecker product or direct product), defined as follows: Let T and Q be matrices of sizes $s\times t$ and $p\times q$, respectively. Then their tensor product is the matrix:

$$T\otimes Q=\begin{pmatrix} T_{11}Q & \dots & T_{1t}Q \\ \vdots & & \vdots \\ T_{s1}Q & \dots & T_{st}Q \end{pmatrix},$$

of size $st\times pq$.

Clearly, $f_B=M(f_A)$, *i.e.* the multiresolution function defined by B is the same as the application of the WFT M to the multiresolution function computed by WFA A.

3. AVERAGE PRESERVING WFT

Let the cardinalities of the alphabets be $p=|\Sigma_1|$ and $q=|\Sigma_2|$. We call the ε-free WFT *average preserving* if for all $a\in\Sigma_1$ holds:

$$\sum_{b\in\Sigma_2} W_{a,b}\cdot F+\frac{q}{p}F. \tag{4}$$

In other words, the final distribution F is an eigenvector of matrices $\Sigma_{b\in\Sigma_2}$, $W_{a,b}$ corresponding to eigenvalue $\frac{q}{p}$, for all $a\in\Sigma_1$.

According to the next theorem the application of an average preserving WFT to an average preserving WFA is an average preserving WFA. Moreover, application to any average preserving multiresolution function is average preserving.

Theorem 1—Let M be an ε-free WFT. $M(A)$ is average preserving for every average preserving WFA A if and only if M is average preserving. Moreover, if M is average preserving, $M(f)$ is average preserving for every average preserving multiresolution function f.

Proof—Let $p=|\Sigma_1|$ and $q=|\Sigma_2|$ be the cardinalities of the alphabets.

($\Rightarrow$): Let $a\in\Sigma_1$ be arbitrary. Consider the average preserving one-state WFA A, with initial and final distributions equal to 1, and with weight matrices $W_a^A=p$ and $W_a^A=0$ for $a'\neq a$. Then $B=M(A)$ is the WFA with initial and final distributions I and F of M, respectively, and with weight matrices $W_b^B=p\cdot W_{a,b}$ for all $b\in\Sigma_2$. The average preserveness condition (2) for WFA B is clearly equivalent to (4) for letter a. Since we assume the WFA B is average preserving regardless of the choice of a, (4) has to be satisfied for every $a\in\Sigma_1$. In other words, WFT M is average preserving.

($\Leftarrow$): Assume M satisfies (4) for every $a\in\Sigma_1$. Let A be an average preserving WFA with initial and final distributions I^A and F^A, and with weight matrices W_a^A for $a\in\Sigma_1$. Let $B=M(A)$ be the application of M to A. Then:

$$\begin{aligned}\sum_{b\in\Sigma_2} W_b^B F^B &= \sum_{\substack{a\in\Sigma_1\\ b\in\Sigma_2}} (W_{a,b}\otimes W_a^A)(F\otimes F^A)\\ &=\sum_{a\in\Sigma_1}\left(\sum_{b\in\Sigma_2} W_{a,b}F\right)\otimes(W_a^A F^A)\\ &=\sum_{a\in\Sigma_1}\frac{q}{p}F\otimes(W_a^A F^A)\\ &=\frac{q}{p}F\otimes\sum_{a\in\Sigma_1} W_a^A F^A\\ &=\frac{q}{p}F\otimes pF^A=qF^B.\end{aligned}$$

In other words, $B=M(A)$ is average preserving.

If M is average preserving and $f:\Sigma_1^*\to\mathbb{R}$ is an average preserving multiresolution function, then $g=M(f)$ satisfies for every $v\in\Sigma_2^*$:

$$\begin{aligned}\sum_{b\in\Sigma_2} g(vb) &= \sum_{b\in\Sigma_2}\sum_{\substack{u\in\Sigma_1^*\\ a\in\Sigma_1}} f(ua)\,(IW_{u,v}W_{a,b}F),\\ &=\sum_{\substack{u\in\Sigma_1^*\\ a\in\Sigma_1}} f(ua)IW_{u,v}\left(\sum_{b\in\Sigma_2} W_{a,b}F\right)\\ &=\sum_{u\in\Sigma_1^*}\left(\sum_{a\in\Sigma_1} f(ua)\right)IW_{u,v}\,\frac{q}{p}\,F\\ &=q\sum_{u\in\Sigma_1^*} f(u)IW_{u,v}F=qg(v),\end{aligned}$$

that is, g is average preserving. □

The composition of weighted relations was defined in ref. [5]. Let us recall its definition, as well as definitions of some other operations. Let $\sigma:(\Sigma_1^\sigma)^*\times(\Sigma_2^\sigma)^*\to\mathbb{R}$ and $\rho:(\Sigma_1^\rho)^*\times(\Sigma_2^\rho)^*\to\mathbb{R}$ be weighted relations. Define $\sigma\circ\rho$ (composition), $\sigma+\rho$(sum), $r\rho$ (product with scalar $r\in\mathbb{R}$), $\sigma\cdot\rho$ (concatenation) and ρ^+ (catenation closure) as follows:

$$(\sigma \circ \rho)(u, v) = \sum_{w \in \Sigma_2^*} \sigma(u, w)\rho(w, v),$$
$$(\sigma + \rho)(u, v) = \sigma(u, v) + \rho(u, v),$$
$$(r\rho)(u, v) = r\rho(u, v),$$
$$(\sigma \cdot \rho)(u, v) = \sum_{\substack{u=u_1u_2 \\ v=v_1v_2}} \sigma(u_1, v_1)\rho(u_2, v_2) \text{ and}$$
$$(\rho^+)(u, v) = \rho(u, v) + (\rho \cdot \rho)(u, v) + (\rho \cdot \rho \cdot \rho)(u, v) + \dots = \sum_{\substack{u=u_1\dots u_k \\ v=v_1\dots v_k}} \rho(u_1, v_1)\rho(u_2, v_2)\dots\rho(u_k, v_k).$$

In the case of composition it is assumed that $\Sigma_2^\sigma = \Sigma_1^\rho$, and in the cases of addition and concatenation that $\Sigma_1^\sigma = \Sigma_1^\rho$ and $\Sigma_2^\sigma = \Sigma_2^\rho$. The composition is a weighted relation between $(\Sigma_1^\sigma)^*$ and $(\Sigma_2^\rho)^*$ while all others are between $(\Sigma_1^\sigma)^*$ and $(\Sigma_2^\sigma)^*$. In the definition of the catenation closure and sum is taken over all decompositions of words u and v into subwords u_i and v_i including empty words. It is possible that the sum does not converge, in which case the catenation closure is not defined. The sum in the definition of composition does not always converge either. However, if the weighted relations are defined by ε-free WFT the composition is always defined.

The following formulas for the applications of the weighted relations to multiresolution function $f : (\Sigma_1^\sigma)^* \to \mathbb{R}$ follow from the definitions above:

$$(\sigma \circ \rho)(f) = \rho[\sigma(f)],$$
$$(\sigma + \rho)(f) = \sigma(f) + \rho(f),$$
$$(r\rho)(f) = r\rho(f),$$
$$[(\sigma \cdot \rho)(f)](w) = \sum_{w=w_1w_2} [\sigma(f)](w_1)[\rho(f)](w_2) \quad \text{and}$$
$$[(\rho^+)(f)](w) = [\rho(f)](w) + [(\rho \cdot \rho)(f)](w) + [(\rho \cdot \rho \cdot \rho)](f)(w) + \dots = \sum_{w=w_1\dots w_k} [\rho(f)](w_1)\dots[\rho(f)(w_k).$$

Again, the catenation closure is defined only if the sum converges (the sum is over all decompositions of w into subwords).

Let us define next corresponding operations on ε-free WFT. Let A (B) be an n_A-state (respectively n_B-state) ε-free WFT from alphabet $\Sigma_1^A(\Sigma_1^B$, respectively) to Σ_2^A (Σ_2^B) with initial distribution I^A (I^B), final distribution F^A (F^B) and weight matrices $W_{a,b}^A$ for $a \in \Sigma_1^A$ and $b \in \Sigma_2^A$ ($W_{a,b}^B$ for $a \in \Sigma_1^B$ and $b \in \Sigma_2^B$, respectively). Define new ε-free WFT $A \circ B$, $A + B$, rA for $r \in \mathbb{R}$, $A \cdot B$ and A^+ as follows:

Composition—Assume $\Sigma_2^A = \Sigma_1^B$. The composition $A \circ B$ is the n_An_B-state WFT from Σ_1^A to Σ_2^B with initial distribution $I^A \otimes I^B$, final distribution $F^A \otimes F^B$ and weight matrices $W_{a,b} = \Sigma_{c \in \Sigma_2^A} W_{a,c}^A \otimes W_{c,b}^B$ for all $a \in \Sigma_1^A$, $b \in \Sigma_2^B$.

Sum—Assume $\Sigma_i^A = \Sigma_i^B$ for $i = 1, 2$. The sum $A + B$ is the $(n_A + n_B)$-state WFT from Σ_1^A to Σ_2^A with initial distribution I^{A+B}, final distribution F^{A+B} and weight matrices $W_{a,b}^{A+B}$, $a \in \Sigma_1^A$ and $b \in \Sigma_2^A$ given by:

$$I^{A+B} = (I^A I^B), \quad F^{A+B} = \begin{pmatrix} F^A \\ F^B \end{pmatrix},$$
$$W_{a,b}^{A+B} = \begin{pmatrix} W_{a,b}^A & 0 \\ 0 & W_{a,b}^B \end{pmatrix}.$$

Multiplication by scalar $r \in \mathbb{R}$: the WFT rA is as A except that the initial distribution vector I^A is replaced by rI^A.

Concatenation—Assume $\Sigma_i^A = \Sigma_i^B$ for $i = 1, 2$. The concatenation of A and B can be defined as the $(n_A + n_B)$-state WFT with $(\varepsilon, \varepsilon)$-transitions given by:

$$I = (I^A \; 0), \quad F = \begin{pmatrix} 0 \\ F^B \end{pmatrix}$$

$$W_{a,b} = \begin{pmatrix} W_{a,b}^A & 0 \\ 0 & W_{a,b}^B \end{pmatrix}, \quad W_{\varepsilon,\varepsilon} = \begin{pmatrix} 0 & F^A I^B \\ 0 & 0 \end{pmatrix}.$$

This is equivalent to the ε-free EFT $A \cdot B$ with initial distribution $I^{A \cdot B} = I + IW_{\varepsilon,\varepsilon} = (I^A \; I^A F^A I^B)$, final distribution $F^{A \cdot B}F$ and weight matrices:

$$W_{a,b}^{A \cdot B} = W_{a,b} + W_{a,b} W_{\varepsilon,\varepsilon} = \begin{pmatrix} W_{a,b}^A & W_{a,b}^A F^A I^B \\ 0 & W_{a,b}^B \end{pmatrix},$$

for all $a \in \Sigma^{A_1}$ and $b \in \Sigma_2^A$.

Catenation closure—the weighted relation f_A^+ is defined only if $|f_A(\varepsilon, \varepsilon)| = |I^A F^A| < 1$. Assume this is the case, and denote $t = 1/(1 - I^A F^A)$.

The catenation closure of the WFT A can be defined as the n_A-state WFT that differs from A only in the weight matrix for $(\varepsilon, \varepsilon)$-transitions: $W_{\varepsilon,\varepsilon} = F^A I^A$. An equivalent ε-free WFT A^+ is specified by initial distribution $I^{A+} = tI^A$, final distribution $F^{A+} = F^A$ and weight matrices $W_{a,b}^{A+} = W_{a,b}^A + tW_{a,b}^A F^A I^A$.

According to the following theorem the operations defined on WFT and weighted relations are compatible. Its proof is straightforward.

Theorem 2—For ε-free WFT A and B holds $f_{A \circ B} = f_A \circ f_B$, $f_{A+B} = f_A + f_B$, $f_{rA} = rf_A$ for all $r \in \mathbb{R}$, $f_{AB} = f_A \cdot f_B$ and $f_{A^+} = f_A^+$. In the case of catenation closure we naturally assume that f_A^+ is defined. □

Our next theorem states that the class of average preserving WFT is closed under the operations of composition, addition and multiplication by a scalar.

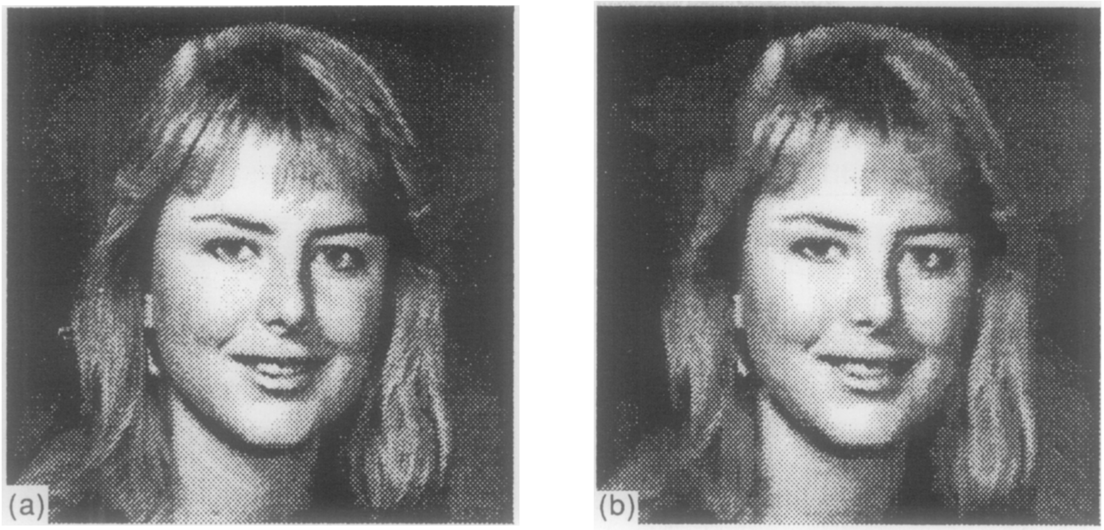

Fig. 2. Image *Carol:* (a) original; and (b) regenerated by a WFA.

Theorem 3—Let A and B be average preserving ε-free WFT. Then $A \circ B$, $A + B$ and rA are average preserving as well.

Proof—Straightforward computations verify (4) in all cases. □

4. EXAMPLES

Now, we will demonstrate that WFT can implement almost every useful linear operator. It has been shown in ref. [5] that every affine transformation on $\mathbb{R}^2$ is realized by a WFT. We give few examples of (piecewise) affine transformations restricted to $[0, 1]^2$. We will illustrate most our WFT by mapping image *Carol* shown in Fig. 2. The original resolution 512×512 8 bits per pixel, is in Fig. 2(a). The image in Fig. 2(b) is regenerated by WFA stored in 2406 bytes ($109 \times$ compression 0.07346 bpp).

We display WFT using similar diagrams as are used for finite automata. States are represented by circles, the initial and final distribution is shown inside the circles. If $(W_{a,b})_{i,j} \neq 0$, then there is an edge from state i to state j labeled by a, b:$(W_{a,b})_{i,j}$.

EFA squeeze shown in Fig. 3(a) implements the sum of two affine transformations:

$$x_1 = \frac{x}{2}, \quad y_1 = y \text{ and } x_2 = \frac{x+1}{2}, \quad y_2 = y,$$

The image Squeeze(*Carol*) is shown in Fig. 3(b).

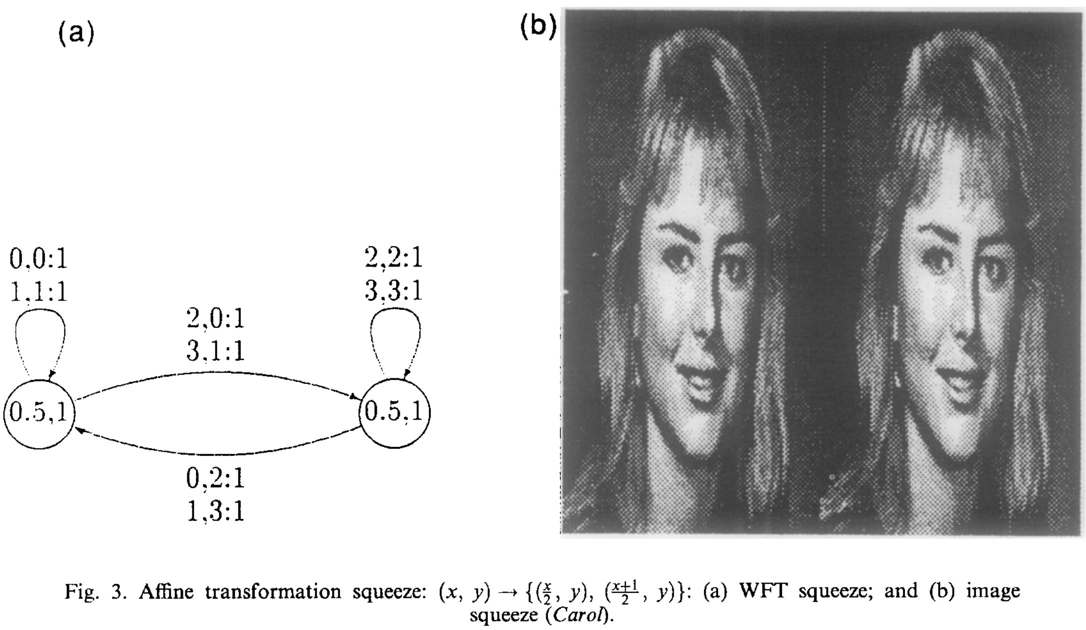

Fig. 3. Affine transformation squeeze: $(x, y) \to \{(\frac{x}{2}, y), (\frac{x+1}{2}, y)\}$: (a) WFT squeeze; and (b) image squeeze (*Carol*).

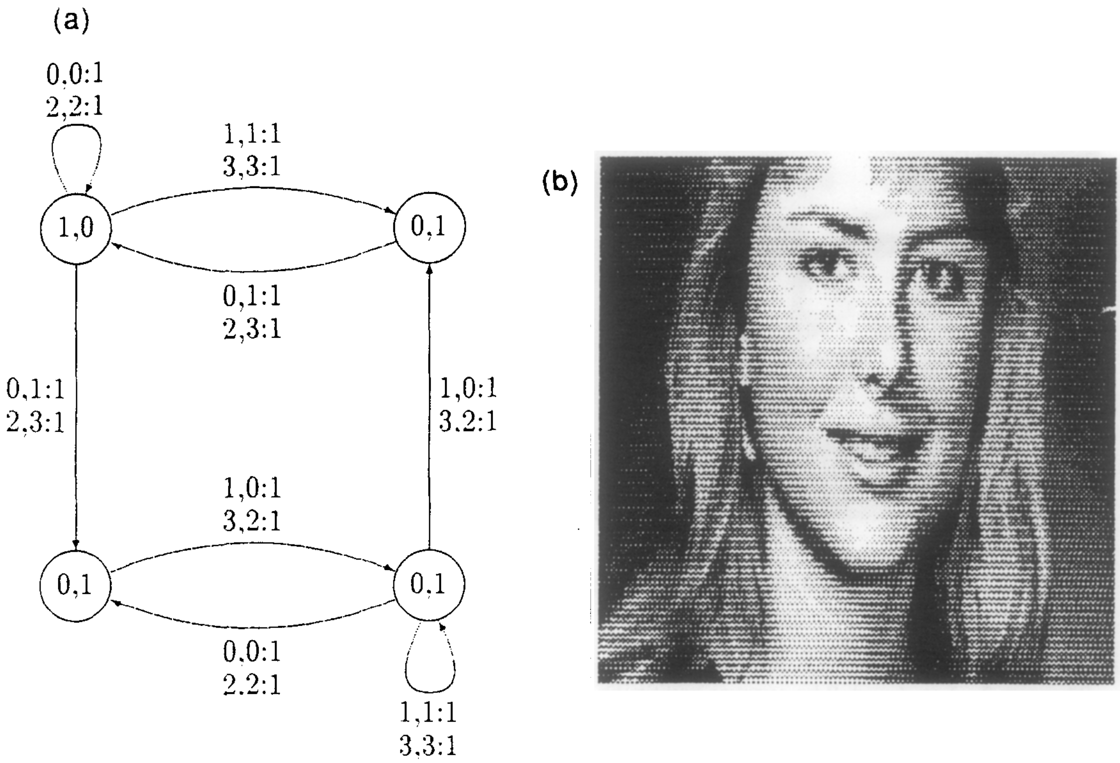

Fig. 4. Affine transformation ϕ: $(x, y) \to (x, \frac{2}{3}y)$: (a) WFT ϕ; and (b) image ϕ(*Carol*).

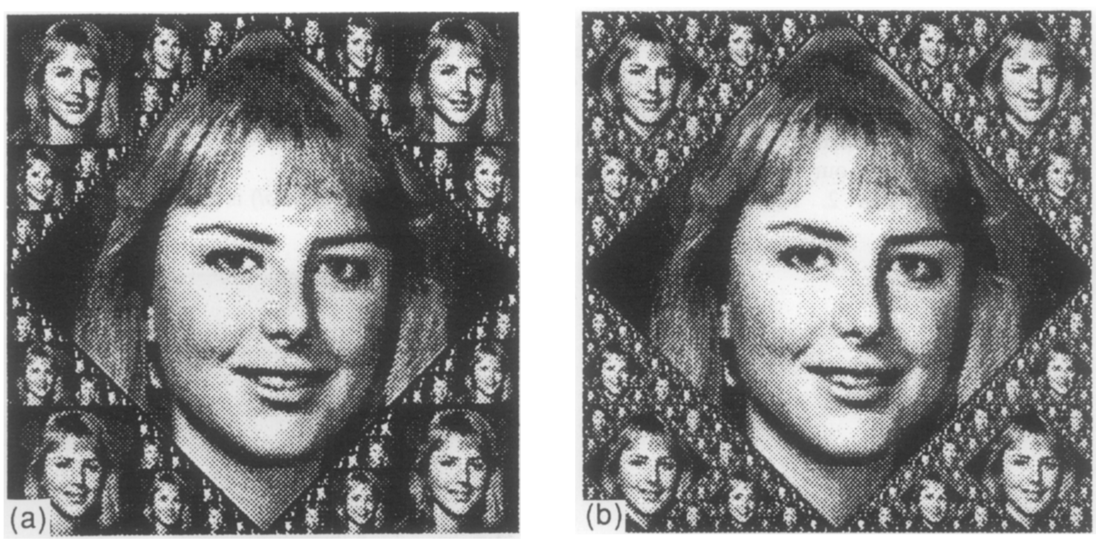

Fig. 5. Application of $(\alpha+\beta)$ to image *Carol*: (a) $(\alpha+\beta)$ (*Carol*); and (b) $(\alpha+\beta)^3$(*Carol*).

The WFT φ that implements the affine transformation $\varphi:(x,y) \to (x, \frac{3}{2}y)$ is shown in Fig. 4(a), and the image φ (*Carol*) in Fig. 4(b).

Next we will do complicated cutting and pasting. WFT α shown in Fig. 6(a) copies the portion of the image in the triangular half of each quadrant closer to the center and leaves zero in the rest. WFT β shown in Fig. 6(b) makes "diminishing copies" of the input image in the other halfs of the quadrants. The application of the sum $\alpha+\beta$ to image *Carol* is shown in Fig. 5.

The technique used in the examples allows to suggest the following definition and theorem.

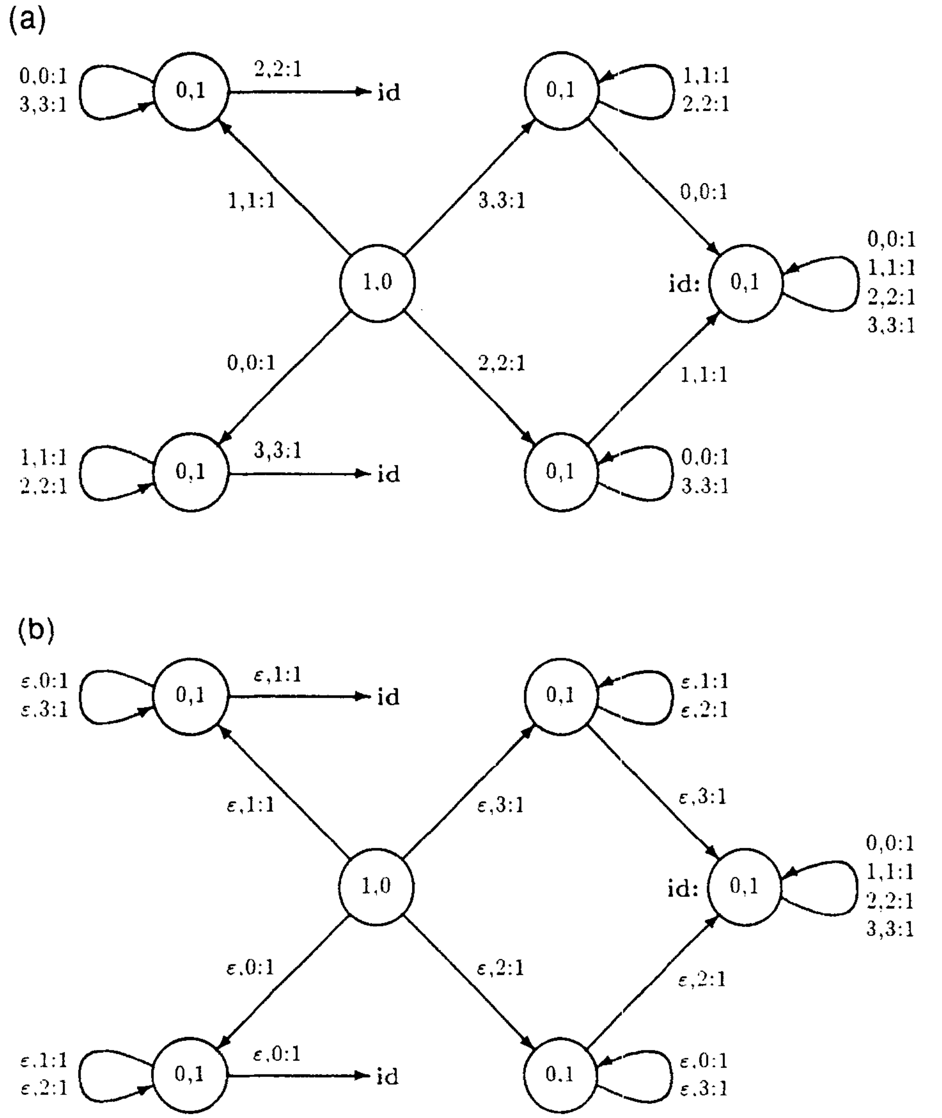

Fig. 6. WFT for "restricted identity" and for "fractal copies".

Definition 1—A transformation composed from affine transformations by cutting (along "rational" lines), pasting and addition is called piecewise affine transformations.

Theorem 4—Every piecewise affine transformation can be implemented by a WFT.

Proof outline—It has been shown in ref. [5] that every affine transformation (restricted to $[0, 1]^2$ can be implemented by a WFT. A practice of "rational" cutting, *i.e.* the restriction to a regular set can, clearly, be implemented by a WFT. Pasting is a special case of addition of WFT. Hence by the closure of WFT under composition and addition we can implement every piecewise affine transformation. Actually, using the operation of con-

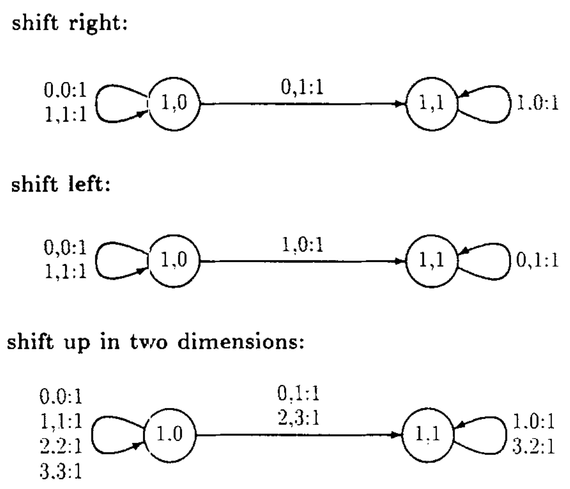

Fig. 7. WFT computing circular shifts by one pixel.

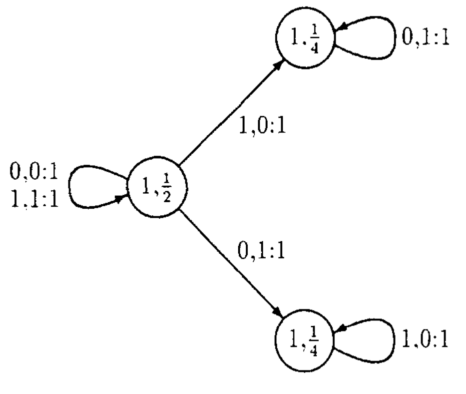

Fig. 8. A WFT computing the filter $F(x)=\frac{1}{4}f(x-h)+\frac{1}{2}f(c)+\frac{1}{4}f(x+h)$.

catenation we can paste infinite number of copies as long as the infinite copying can be expressed by a regular set (WFA if the grayness is not uniform).

Note that every affine transformation is implemented by an average preserving WFT. This is not necessary case for piecewise affine transformation, however, every piecewise affine transformation can be implemented by a WFT in which all states visited more than uniformly bounded number of times satisfies the average preserving condition (4).

Many interesting WFT can be designed using the technique of value shifting. In one dimension, the shift by one pixel (one step in a function table) for resolution 2^n, $n\geqslant 1$ requires to move the value at address $w10^r$ to the address $w10^r$ for $0\leqslant r\leqslant n-1$ and $w\in\Sigma^{n-r-1}$. Therefore, somewhat surprisingly, we

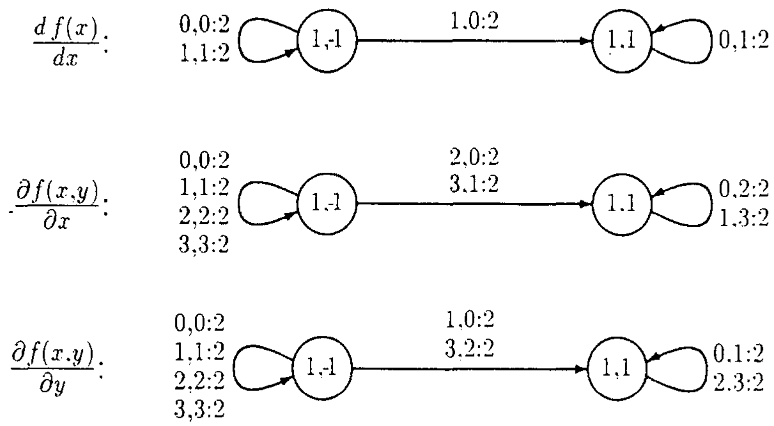

Fig. 9. WFT computing derivative and partial derivatives.

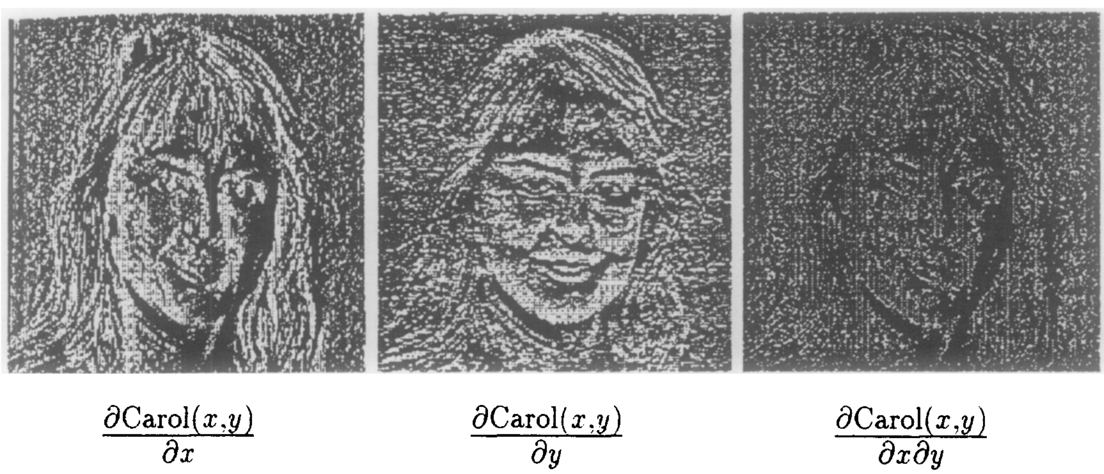

Fig. 10. The partial derivatives of *Carol*.

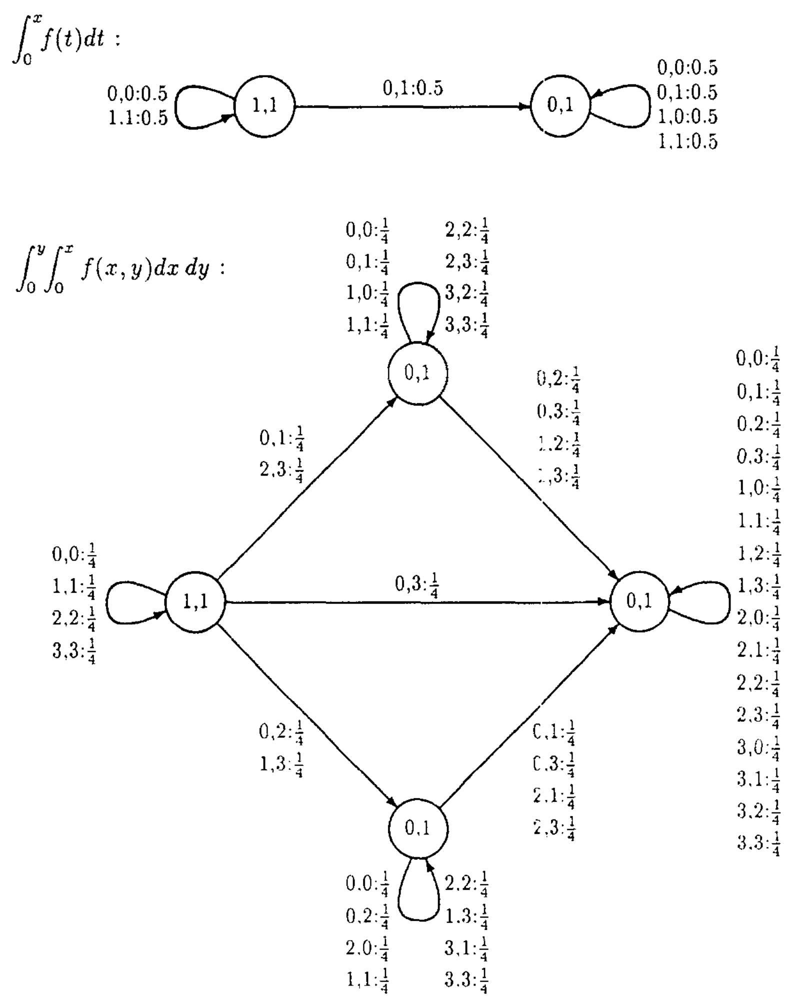

Fig. 11. WFT computing integrals.

easily design a WFT with only two states which performs this shift. By appropriately choosing the initial distribution we can make the shift circular. WFT computing various shifts are shown in Fig. 7.

Since WFT are closed under composition and addition we, clearly, can design a WFT that computes any linear combination of the values of each cell and any fixed finite set of its neighbors. Example of such WFT are a WFT that simulates the moves of a knight on a $2^n \times 2^n$ chess-board, WFT that implements any type of low or high pass filter (one such filter is shown in Fig. 8) or WFT Diff which for any finite resolution computes $\frac{f(x+h-f(x)}{h}$ for table step h. Thus WFT Diff computes the derivative $\frac{\mathrm{d}f(x)}{\mathrm{d}x}$ in the limit (infinite resolution). WFT Diff and similar WFT that compute the partial derivates for functions of two variables are shown in Fig. 9. In Fig. 10 we show the sign of the partial derivatives of *Carol*. Any positive value is represented by white color, negative by black.

Computing of integrals requires to compute linear combinations of values from an unbounded number of neighbors but that can be done, too. WFT shown at the top of Fig. 11, for each resolution, computes $h[f(0)+f(h)+f(2h)+\cdots+f(x)]$ where h is the table step (distance of two neighboring pixels). Thus for the infinite resolution WFT int computes:

$$\int_0^x f(x)\,\mathrm{d}x = \lim_{h\to 0} h[f(0)+f(h)+f(2h)+\cdots+f(x)].$$

WFT computing (in the limit) $\int_0^y \int_0^x f(x, y)\,\mathrm{d}x\,\mathrm{d}y$ is shown at the bottom of Fig. 11.

In all the examples above the same small WFT computes the desired transformation for every resolution and for the infinite resolution as well. For the shifts the limiting case is the identity transformation. However, it is easy to give an average preserving WFT which does not converge to any mapping on infinite resolution. Consider one state WFT with $I=F=1$, and transitions $(0, 0:1)$, $(1, 0:1)$, $(2, 2:1)$ and $(3, 2:1)$.

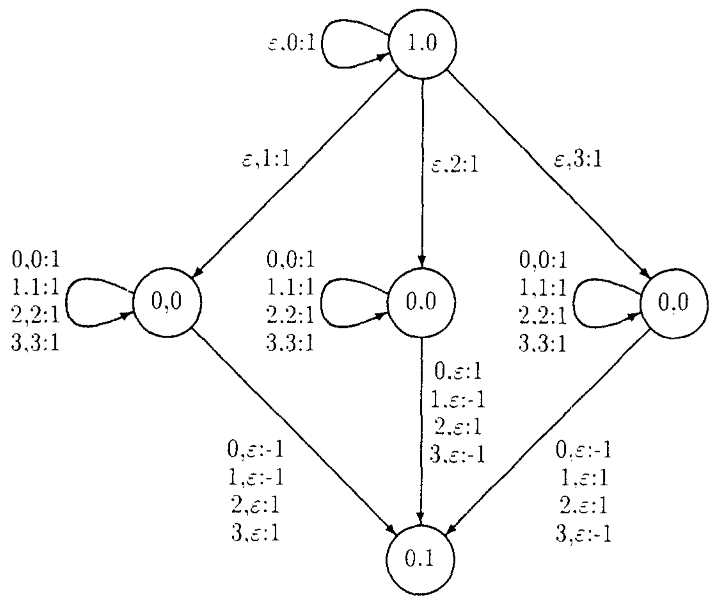

Fig. 12. WFT computing Haar wavelet coefficients in Mallat form.

Theorem 5—Let $\Sigma=\{0,1\}$ and A be *a* WFA over Σ, Then we can construct WFA B such that:

(i) $\hat{f}_B(x)=\frac{\mathrm{d}f_A(x)}{\mathrm{d}x}$ for all $x\in[0,1]$ for which $\frac{\mathrm{d}f_A(x)}{\mathrm{d}x}$ exists,

(ii) $\hat{f}_B(x)=\int_0^t f_A(t)\,\mathrm{d}t$ if the Rieman integral exists.

Proof—Follows from the examples above and the closure of WFA under WFT [5]. □

Note that (ii) is an extension of a result in ref. [1] where this result was shown for the restricted case of the average preserving WFA. However, if WFA A has n states the construction in ref. [1] yields B with $n+1$ states while the application of our two state integrating WFA yields WFA B with $2n$ states.

Theorem 6—Let $\Sigma=\{0, 1, 2, 3\}$ and A be *a* WFA over Σ with $\hat{f}_A$:$[0,1]^2\rightarrow\mathbb{R}$. Then we can construct WFA B such that:

(i) $\hat{f}_B(x,\ y)=\frac{\partial \hat{f}_A(x,y)}{\partial^i x\,\partial^j y}$ for all $(x,y)\in[0,\ 1]^2$ for which the derivative exists and all $i,j\geqslant 1$.

(ii) $\hat{f}_B((x,y)=\int_0^x\hat{f}_A(t,\mathrm{y})\,\mathrm{d}t$ for all $x\in[0,1]$ for which the Rieman integral exists.

(iii) $\hat{f}_B(x,y)=\int_0^y\hat{f}_A(x,t)\,\mathrm{d}t$ for all $y\in[0,1]$ for which the Rieman integral exists.

(iv) $\hat{f}_B(x,y)=\int_0^x\int_0^y\hat{f}_A(t,s)\,\mathrm{d}s\,\mathrm{d}t$ for all $(x,y)\in[0,1]^2$ for which the Rieman integral exists.

Proof—Follows from the examples above, the closure of WFA under composition and the closure of WFA under WFT [5]. □

WFT Mallat shown in Fig. 12 computes the coefficients of the discrete Haar wavelet transform for any finite resolution and presents them in the Mallat form [9]. It computes the continuous Haar wavelet transform for the infinite resolution. The image *Mallat*(*Carol*) is shown in Fig. 13 with the contrast increased. In order to display better all the coefficients we compute them unscaled, that is use wavelets which are not orthonormal. To get the orthonormal case it is sufficient just to adjust the weights of the ε transitions of our WFT.

Fig. 13. The image *Mallat*(*Carol*).

All the examples of image transformations here were produced using a menu-driven X-windows based system wftx implemented by P. Rajcani. In wftx, images are represented either in pixel form or by WFA. The conversions between these representations are implemented using the WFA inference and the WFA decoding algorithms from refs [4, 6]. All the operations on images, WFA and WFT, except for the concatenation closure are implemented in wftx. In particular, a WFT can be applied to an image in pixel form (resolution $2^n \times 2^n$, $n \geqslant 1$) and produce again an image in possibly different resolution, or a WFT can be applied to a WFA and produce a WFA. Any of the WFA computing integrals is a typical example of a WFA which has a small number of states but it is highly nondeterministic. For such a WFT, it is much more efficient to apply it to an image in the WFA representation even if it requires converting the image to the WFA representation and then the WFA decoding.

5. CONCLUSIONS

In the previous section we have shown that many relatively complicated image transformations can be expressed by simple WFT. Since there exists an efficient implementation of WFT, we can use WFT as a powerful design tool for image transformation. In addition, using the wftx system, any WFT can be applied to an image either in the pixel form or in the WFA-compressed form. Due to this property, we can process compressed images without decoding them. It should also be noted that WFT provide a rigorous mathematical description of image transformations.

WFA and WFT can also be used with alphabets of different size than $2^2 = 4$. Using an alphabet with $3^2 = 9$ letters enables us to manipulate images of size $3^n \times 3^n$, using $3 \times 2 = 6$ letters images of size $3^n \times 2^n$, *etc.*

If image approximations are sufficient, we can process images of arbitrary rectangular size by squeezing/stretching them first to the closest size $2^n \times 2^n$ and after manipulation unsqueezing/unstretching them back to the original size. This method has been implemented in the wftx system.

Acknowledgement—Research was supported by the National Science Foundation under Grant No. CCR-9202396.

REFERENCES

1. K. Culik II and J. Karhumäki, Automata computing real function, *SIAM J. Comput* (to appear) Technical Report TR 9105, University of South Carolina, Columbia (1991).
2. K. Culik II and J. Kari, Image compression using weighted finite automata, *Comput. Graphics* **17**, 305–313 (1993).
3. K. Culik II and J. Kari, Image-data compression using edge-optimizing algorithm for WFA inference, *J. Inform. Process. Mgmt*, to appear (1995).
4. K. Culik II and J. Kari, Efficient inference algorithm for weighted finite automata, *Fractal Image Encoding and Compression* (Ed. Y. Fisher), Springer-Verlag (1995).
5. K. Culik II and I. Fris, Weighted finite transducers in image processing, *Discr. Appl. Math.* **58**, 223–237 (1995).
6. S. Eilenberg, *Automata, Languages and Machines*, Vol. A, Academic Press, New York (1974).
7. K. Culik II and P. Rajcani, Iterative weighted finite transductions, Technical Report TR 0094, University of South Carolina (1994).
8. K. Culik II and S. Dube, Balancing order and chaos in image generation, *Comput. Graphics* **17**, 465–486 (1993).
9. R. A. DeVore, B. Jawerth and B. J. Lucier, Image compression through wavelet transform coding, *IEEE Trans. Inform. Theory* **38**, 719–746 (1992).
10. M. F. Barnsley, *Fractals Everywhere*, Academic Press, New York (1988).