

WFA를 이용한 이미지 압축 알고리즘에 대한 분석

업준형 김태환

한국과학기술원 전산학과

첨단정보기술 연구센터(AITrc)

jhum@vlsisyn.kaist.ac.kr tkim@cs.kaist.ac.kr

Analysis on Image Compression using Weighted Finite Automata

Junhyung Um Taewhan Kim

Dept. of EECS, Korea Advanced Institute of Science and Technology
and Advanced Information Technology Research Center(AITrc)

요 약

본 논문에서 우리는 grey scale 영상을 weighted finite automata(WFA)로써 기술하는 두개의 알고리즘[2, 4]을 분석하였다. 또한, 원영상과 WFA를 이용하여 압축된 영상간의 error를 분석하고 그 결과를 제시하였다. 구체적으로, 영상복원 tolerance δ 를 이용하여 찾아진 automaton에 의해 복원된 영상과 원영상의 l^2 -norm의 차이가 δ 보다 작거나 같음을 증명하였다.

1 서론

Weighted Finite Automata(WFA)는 [2]에서 소개되고, 많은 부드러운 함수나 프랙탈의 모습을 띠는 함수를 포함하는 실함수를 계산하는 도구로써 연구되어 왔다. [3]에서 Culki II과 Kari는 WFA를 실함수를 정의하는 도구로써 사용하였다. 특히나, 그 중에서도 grey tone 영상을 정의하기 위한 도구로써 이용하였다. 즉, 임의의 grey-tone 영상(디지털 화 된 사진)을 WFA로 자동으로 인코딩하기 위한 알고리즘을 제시하였다. 그러나, [3]에서는 이러한 알고리즘이 왜 동작하는지에 대한 증명이 제시되어 있지 않았다. 본 논문에서는 이러한 인코딩 알고리즘이 실제로 원상을 정확히 복원함을 증명하고, 구체적인 예를 들어 어떻게 영상이 인코딩 되는지 간단히 설명하며, 알고리즘의 linear-algebraic한 성질에 대해 논한다. 그리고, 마지막으로 실제적인 인코딩 알고리즘에 대한 예 분석을 수행한다.

2 영상 함수와 WFA

우리는, 2^n by 2^n 의 정수값(실제적으로 이 값은 0부터 225 사이의 값을 갖는다)을 갖는 픽셀로 이루어진 디지털화된 grey-tone 영상을 finite-resolution 영상으로 정의하며 2^n by 2^n ($n = 0, 1, \dots$)의 resolution 영상으로 이루어진 서로 compatible한 영상들의 집합을 multi-resolution 영상으로 정의한다. 우리는, 2^n by 2^n resolution 의 각 픽셀에 알파벳 $\Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$ 로 이루어진 길이 n 인 word를 할당한다. 그러므로, 우리의 형식에서 multi-resolution 영상은, Σ^n 을 길이 n 인 모든 word의 집합으로 정의할때, $\Sigma^n := \cup_{n=0}^{\infty} \Sigma^n$ 에서 정의되는 함수라 할 수 있다. 2^n by 2^n resolution에는 $2^n \times 2^n = 4^n$ 개의 픽셀이 존재하며, Σ 위의 길이 n 인 word의 개수 또한 4^n 개이므로, 이러한 wording의 방법은 합리적이라 할 수 있다. 서로 다른 resolution의 compatibility는, $F: \Sigma^n \rightarrow \mathbb{R}$ 가 average preserving 함수일때 성립한다고 정의한다. 함수 $F: \Sigma^n \rightarrow \mathbb{R}$ 가 average preserving(ap)임은 다음과 같은 성질이 성립할때이다:

모든 $w \in \Sigma^*$ 에 대해,

$$F(w) = \frac{1}{4}[F(w(0,0)) + F(w(0,1)) + F(w(1,0)) + F(w(1,1))]. \quad (1)$$

이제, $[0, 1]^2$ 에서의 dyadic subsquare들이 Σ 위의 word로써 어떻게 생성되는지를 설명한다. 만약 $w \in \Sigma^*$ with $w = (a_1, b_1)(a_2, b_2) \dots (a_k, b_k)$ 이면, 우리는 w 가 왼쪽 아래의 점의 좌표를 $(\sum_{i=1}^k \frac{a_i}{2^i}, \sum_{i=1}^k \frac{b_i}{2^i})$ 로 갖고, 변의 길이를 2^{-k} 로 갖는 dyadic subsquare의 주소라 한다. 예를 들어, 전체 $[0, 1]^2$ square의 주소는 ϵ 이고, ($\epsilon :=$ empty word), 그럼 1(a)의 점은 사각형의 주소는 $(0,0)(1,0)(1,0)$ 이다.

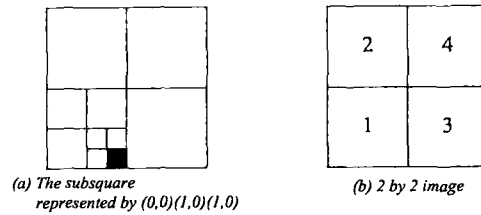


그림 1: 영상과 WFA에 대한 예

$A = (Q, \Sigma, f, \alpha, \beta)$ 에 대한 Weighted finite automaton (WFA)는 다음과 같은 성질을 가지는 5-tuple을 의미한다:

1. Q 는 state들의 유한한 집합을 의미한다.
2. Σ 는 유한 알파벳이다. (본문에서는, $\Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$).
3. $W: Q \times \Sigma \times Q \rightarrow \mathbb{R}$ 는 weight 함수이다.
4. $\alpha: Q \rightarrow \mathbb{R}$ 는 initial distribution이다.
5. $\beta: Q \rightarrow \mathbb{R}$ 는 final distribution이다.

우리는 $f(p, a, q) \neq 0$ 가 성립하는 경우 $(p, a, q) \in Q \times \Sigma \times Q$ 를 A 위의 *transition*이라 하며, a 를 transition의 *label*이라 한다. 이때, WFA A 는 다음과 같은 distribution 함수 $\delta: Q \times \Sigma^* \rightarrow (-\infty, \infty)$ 를 정의한다:

- (a) 모든 $q \in Q$ 에 대해 $\delta(q, \epsilon) = \alpha(q)$.
- (b) 모든 $q \in Q, w \in \Sigma^*$ 와 $a \in \Sigma$ 에 대해 $\delta(q, wa) = \sum_{p \in Q} \delta(p, w)f(p, a, q)$.

WFA A 는 $\phi_A(w) = \sum_{q \in Q} \beta(q)\delta(q, w)$ 로 표현되는 영상함수 $\phi_A: \Sigma^* \rightarrow (-\infty, \infty)$ 를 정의하며, $w = a_1 \dots a_n$ 에 대한 $\phi_A(w)$ 값은 다음과 같이 구해진다:

$$\begin{aligned} \phi_A(w) &= \phi(a_1 a_2 \dots a_n) = \sum_{q \in Q} \beta(q)\delta(q, a_1 a_2 \dots a_n) \\ &= \sum_{q_n \in Q} \delta(q_n, a_1 a_2 \dots a_n)\beta(q_n) \\ &= \sum_{q_n \in Q, q_{n-1} \in Q} \delta(q_n, a_1 a_2 \dots a_{n-1})f(q_{n-1}, a_n, q_n)\beta(q_n) \\ &= \sum_{q_n, \dots, q_0 \in Q} \alpha(q_0)f(q_0, a_1, q_1)f(q_1, a_2, q_2) \dots \\ &\quad f(q_{n-1}, a_n, q_n)\beta(q_n). \end{aligned} \tag{2}$$

3 WFA와 이미지 압축에 대한 여러가지 성질

[3]는 WFA를 이용한 a lossless encoding 알고리즘을 제시하였으나, 실제로 왜 해당 알고리즘이 동작하는지에 대한 증명이 주어지지 않았다. 우리는 이 단락에서 encoding 알고리즘[3]이 원상을 정확히 복구함을 증명한다.

먼저, WFA의 *average preserving*을 정의한다. WFA A 가 *average preserving* (ap-WFA)임은 모든 $p \in Q$ 에 대해 다음의 성질이 성립하는 것과 동치이다:

$$\sum_{a \in \Sigma, q \in Q} f(p, a, q)\beta(q) = 4\beta(p). \tag{3}$$

Theorem 1 A 를 WFA라 하자. 만약 A 가 *average preserving* 함수이면, ϕ_A 또한 *average preserving* 함수이다.

Proof. [3]에 제시되어 있다. □

다음에 주어지는 lemma와 Theorem 2는 ap-WFA로 나타내어지는 두개의 함수의 합 또한 ap-WFA로 나타내어 진다는 점에서 [3]의 Theorem 2를 가 성립함을 보이며, 또한 ap-WFA로 나타내어지는 영상들의 family가 vector-space가 되는 것을 보인다.

Lemma 1 ap-WFA로 나타내어지는 두 함수의 합 또한 ap-WFA로 나타내어 진다.

Theorem 2 ap-WFA로 나타내어지는 영상들의 family는 vector-space이다.

Culik II 과 Kari [3]는 다음의 encoding 알고리즘을 증명 없이 제시하였으며, Theorem 3에서 이를 증명한다. 알고리즘에서 우리는 다음과 같은 기호들을 사용한다. Ap-함수 ϕ 와 word $w \in \Sigma^*$, ϕ_w 는 다음과 같이 정의되는 ap-함수로 정의한다:

$$\phi_w(u) = \phi(wu), \forall u \in \Sigma^*.$$

이때, 함수 ϕ_w 는 dyadic subsquare w 에서 정의되는 ap-함수 ϕ 의 영상을 나타낸다.

Encoding 알고리즘 A([3])

$\Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$ 라 하자.

Ap-함수 $\phi: \Sigma^* \rightarrow \mathbb{R}$ 에 의해 정의되는 영상에 대해, an ap-WFA $A = (\{q_0, q_1, \dots, q_N\}, \Sigma, f, \alpha, \beta)$ ($\phi_A = \phi$)를 만든다.

N : state가 만들어진 마지막 index

i : 프로세스가 진행되지 않은 첫번째 state의 index

$\gamma: Q \rightarrow \Sigma^*$: state에서 subsquare로의 mapping mapping

1. Set $N = 0, i = 0, \beta(q_0) = \phi(\epsilon), \gamma(q_0) = \epsilon$.
2. Process q_i , 즉, $w = \gamma(q_i)$ 과 $a = (0,0), (0,1), (1,0), (1,1)$ 에 대해,
 - (a) 만약 다음의 성질을 만족하는 스칼라 c_0, \dots, c_N 가 존재하면, $\phi_w = c_0\phi_0 + \dots + c_N\phi_N$ ($\phi_j = \phi_{\gamma(q_j)}$ for $j = 0, \dots, N$), $f(q_i, a, q_j) = c_j$ ($j = 0, \dots, N$),
 - (b) 그렇지 않다면, 다음과 같이 놓는다: $\gamma(q_{i+1}) = wa, \beta(q_{N+1}) = \phi(wa), f(q_i, a, q_{N+1}) = 1, N = N + 1,$
3. $i = i + 1$. 만약 $i \leq N$, 2로 진행.
4. $\alpha(q_0) = 1, \alpha(q_j) = 0$ for $j = 1, \dots, N$.

우리는 단락 4에서 사용될 Σ^* 위의 order를 정의한다. Σ^* 는 다음과 같은 order \preceq 를 사용하는 linearly-ordered(well-ordered) 집합을 의미한다:

1. ϵ 는 모든 word에 우선한다.
2. $(0,0) \preceq (0,1) \preceq (1,0) \preceq (1,1)$.
3. 만약 a 와 b 가 $|a| = n$ 이고 $|b| = m$ ($n < m$)인 word이면, $a \preceq b$.
4. 만약 $a = a_1 \dots a_n, b = b_1 \dots b_n, a \neq b$ 이면, j 가 a 가 b 가 다른 첫 번째 'digit'라 할때, $a \preceq b$ 는 $a \preceq b$ 이고 $a \neq b$ 임을 의미한다.

Theorem 3 A 를 encoding 알고리즘 A 에 의해 만들어진 WFA라 하고, $Q = \{q_0, \dots, q_N\}$ 라 하자. 그러면, $\phi_A(w) = \sum_{r_0, r_1, \dots, r_N \in Q} \alpha(r_0) f(r_0, a_1, r_1) f(r_1, a_2, r_2) \dots f(r_{N-1}, a_n, r_N) \beta(r_N)$ 는 dyadic subsquare w 의 mean value를 나타낸다.

다음의 예는 알고리즘의 동작을 보인다. 우리는 인코딩 알고리즘 A 에 의해 만들어진 WFA를 decode하기 위해 수식(2)를 사용한다. (빠른 decoding 알고리즘은 [3]에 제시되어 있다.)

Example 1 Encoding 알고리즘 A 를 그림 1(b)의 영상에 적용하여 ϕ_A 를 생성하는 ap-WFA를 찾는다. 먼저, state q_0 를 square ϵ 에 적용하면, $\beta(q_0) = \frac{1}{4}$ 이 된다(ϵ 에서의 average greyness). 이제, 주소 $(0,0), (0,1), (1,0), (1,1)$ 를 갖는 네개의 subsquare에 대해 생각해 보자. 첫번째 subsquare $(0,0)$ 의 영상은 $\gamma(q_0)$ 의 선형 결합으로 표현될 수 없기 때문에, 두번째 state q_1 가 생성된다. $f(q_0, (0,0), q_1) = 1$ 와 $\beta(q_1) = 1$ 로 각각 정의하자(subsquare $(0,0)$ 의 average greyness는 1이다). ϕ_1 를 square $(0,0)$ 의 ϕ_A 로 주어지는 영상으로 놓자. 그러면, address $(0,1)$ 의 두번째 subsquare의 영상은 $2\phi_1$ 로 표현가능하다(이는 영상 ϕ_1 의 contrast를 두배 증가시키면 된다.) 그러므로, $f(q_0, (0,1), q_1) = 2$ 이고, 주소 $(1,0)$ 과 $(1,1)$ 를 가지는 세번째와 네번째의 subsquare는 각각 $3\phi_1$ 와 $4\phi_1$ 로 표현 가능하기 때문에, $f(q_0, (1,0), q_1) = 3$ 이고, $f(q_0, (1,1), q_1) = 4$. Initial distribution α 는 다음과 같이 정의된다: $\alpha(q_0) = 1, \alpha(q_1) = 0$. 알고리즘은 여기서 끝나며, WFA가 생성되었다.

이제, $\phi_A(w)$ 를 계산하여 Encoding Algorithm A 에 의해 생성된 WFA가 영상을 정확하게 encoding하는지를 확인한다.

$$\begin{aligned} \delta(q_0, (0,0)) &= \sum_{p \in Q} \delta(p, \epsilon)f(p, (0,0), q_0) \\ &= \sum_{p \in Q} \alpha(p)f(p, (0,0), q_0) = \alpha(q_0)f(q_0, (0,0), q_0) \\ &= 1 \cdot f(q_0, (0,0), q_0) = 0. \end{aligned}$$

$$\begin{aligned} \delta(q_1, (0,0)) &= \sum_{p \in Q} \delta(p, \epsilon)f(p, (0,0), q_1) \\ &= \sum_{p \in Q} \alpha(p)f(p, (0,0), q_1) = \alpha(q_0)f(q_0, (0,0), q_1) \\ &= 1 \cdot f(q_0, (0,0), q_1) = 1. \end{aligned}$$

$$\begin{aligned} \delta(q_0, (0, 1)) &= \sum_{p \in Q} \delta(p, \epsilon) f(p, (0, 1), q_0) \\ &= f(q_0, (0, 1), q_0) = 0. \\ \delta(q_1, (0, 1)) &= \sum_{p \in Q} \delta(p, \epsilon) f(p, (0, 1), q_1) \\ &= f(q_0, (0, 1), q_1) = 2. \end{aligned}$$

그러므로,

$$\begin{aligned} \phi((0, 0)) &= \sum_{q \in Q} \beta(q) \delta(q, (0, 0)) = \sum_{q \in Q} \beta(q) \delta(q, (0, 0)) \\ &= \beta(q_1) \cdot \delta(q_1, (0, 0)) = 1. \\ \phi((0, 1)) &= \sum_{q \in Q} \beta(q) \delta(q, (0, 1)) \beta(q_1) \cdot \delta(q_1, (0, 1)) = 2. \end{aligned} \tag{4}$$

또한, 비슷한 방법으로 $\phi_A((1, 0)) = 3$ 와 $\phi_A((1, 1)) = 4$ 이 성립함을 보일 수 있다.

이제, 우리는 제시된 알고리즘의 linear-algebraic 성질에 대해 언급하고자 한다. 원 이미지 ϕ 가 $2^n \times 2^n$ resolution을 가지고 있다고 가정하자. 이제, ϕ_w 를 subsquare w 의 ϕ 에 의해 정의된 normalized vector라 하자. 길이 k 의 word w 에 대해, ϕ_w 를 $2^{n-k} \times 2^{n-k}$ 의 vector라 하자. 또한, ϕ_w 를 그림 2과 같은 $2^n \times 2^n$ vector라 하자. 그림 2에서, 영상 ϕ 는 왼쪽에 보이고, $\phi_w = \phi_{(0,1)}$ 는 중앙에, 그리고 $\phi_w = \phi_{(0,0),(1,1)}$ 는 오른쪽에 보인다.

3	5	6	7	21	21	34	34	22	22	22	22
11	13	5	4	21	21	34	34	22	22	22	22
35	22	21	34	7	7	8	8	22	22	22	22
1	2	7	3	7	7	8	8	22	22	22	22

그림 2: 벡터 $\phi_{i\epsilon}$, $\phi_{(0,1)}$, $\phi_{(0,0),(1,1)}$ 의 예

다음의 정리는, 최소의 개수의 state를 가진 encoding 알고리즘 A가 원 영상을 정확히 정의하는 것이 $\{\phi_w | w \in \Sigma^*\}$ 가 유한 dimension d 의 선형공간을 생성하는 것과 동치라는 것을 증명한다. generates a linear space of finite dimension d .

Theorem 4 함수 $\phi: \Sigma^* \rightarrow \mathbb{R}^d$ 가 WFA를 이용해서 정의될 수 있는 것과 Σ^* 위에서 정의된 ϕ 함수(즉, 집합 $\{\phi_w | w \in \Sigma^*\}$)의 집합이 유한 dimension d 의 linear space를 생성하는 것은 동치이다.

이제, 2^n by 2^n 의 pixel로 이루어진 multi-resolution 영상을 생각해 보자. Encoding 알고리즘 A를 영상 ϕ 에 적용하는 것과 $\sum_{k=0}^n 4^k$ 개의 vector $\{\phi_w | w \in \Sigma^*\}$ 로부터 Hamel basis를 찾는 것이 동치라는 것은 쉽게 알 수 있다. 각 multi-resolution level j 에 대해 (즉 $\{\phi_w | w \in \Sigma^j, 0 \leq j \leq n\}$) ϕ_w 는 blowing-up 전에는 실제적으로 2^{n-j} by 2^{n-j} 의 행렬이고 $|\{\phi_w | w \in \Sigma^j, 0 \leq j \leq n\}| = 4^j$ 이기 때문에, level j 의 dimension은 적어도 $\min\{4^j, 4^{n-j}\}$ 이다. 그러므로, ϕ by ϕ 의 모든 subsquare에서 정의되는 vector들은 유한 dimension k 의 다음과 같은 linear space를 생성한다.

$$k \leq \begin{cases} 2 \sum_{k=0}^{\frac{n}{2}-1} 4^k + 4^{\frac{n}{2}} = \frac{2}{3} 4^{\frac{n}{2}} - \frac{2}{3}, & \text{if } n \text{ is even,} \\ 2 \sum_{k=0}^{\frac{n-1}{2}} 4^k = \frac{2}{3} (4^{\frac{n+1}{2}} - 1), & \text{if } n \text{ is odd.} \end{cases}$$

본문에서 주어진 예에서, linear space의 basis는 $\{\phi_0, \phi_1\}$ 이다. 그러므로, 그림 2의 dimension은 2이다.

4 Error 분석

영상을 압축할 때, 원상을 정확히 생성하는 ap-WFA를 생성하는 것은 좋은 방법이 아니다. 어떤 finite resolution 영상이라도 ap-WFA를 이용하면 원상이 정확하게 복구된다. 그러나, 대부분의 경우 이러한 automaton은 상당히 크게 된다. 반면에, 다음에 주어진 encoding 알고리즘 B [3]에서처럼 어느 정도의 에러를 허용하면, 훨씬 적은 양의 automaton이 생성된다. 이 단락에서 우리는 주어진 tolerance에 대한 에러의 upper bound를 제시한다. 그러므로, 주어진 원상에 근접하는 영상을 주어진 tolerance에서 재생성하는 Encoding 알고리즘 B의 tolerance를 미리 정할 수 있다.

Encoding Algorithm B

$\Sigma = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. 주어진 영상을 정의하는 ap-함수 $\phi: \Sigma^* \rightarrow \mathbb{R}^d$ 와 tolerance $\delta > 0$ 에 대해, $\phi_{A_\delta} \approx \phi$ 인 ap-WFA $A_\delta = (Q_\delta, \Sigma, f_\delta, \alpha_\delta, \beta_\delta)$ 를 생성한다.

N : 마지막 state가 생성된 index

i : 알고리즘이 적용되지 않은 첫번째 state의 index

δ : 주어진 tolerance

$\gamma: Q \rightarrow \Sigma^*$: state에서 subsquare로의 mapping

1. $N = 0, i = 0, \beta_\delta(q_0) = \phi(\epsilon), \gamma_\delta(q_0) = \epsilon$.

2. Process q_i , 즉,

주어진 $w = \gamma_\delta(q_i)$ 와 각 $a \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ 에 대해

(a) 만약 다음을 만족하는 c_0, \dots, c_N 가 존재하면,

$$\|\phi_w - (c_0 \phi_0 + \dots + c_N \phi_N)\|_{l^2(2^n \times 2^n)} \leq \delta,$$

(이때, $\phi_j = \phi_{\gamma(q_j)}$ ($j = 0, \dots, N$) 이고 $c_0 \phi_0 + \dots + c_N \phi_N$ 는 ϕ_w 를 $\text{span}\{\phi_0, \dots, \phi_N\}$ 로 orthogonal projection 한 결과이다) $f_\delta(q_i, a, q_j) = c_j$ 로 놓는다($j = 0, \dots, N$).

(b) 그렇지 않다면, $\gamma_\delta(q_{i+1}) = wa, \beta_\delta(q_{N+1}) = \phi(wa),$

$$f_\delta(q_i, a, q_{N+1}) = 1, N = N + 1.$$

3. $i = i + 1$. 만약 $i \leq N$ 이면, goto 2.

4. $\alpha_\delta(q_0) = 1, \alpha_\delta(q_j) = 0$ for $j = 1, \dots, N$.

Theorem 6 ϕ 를 주어진 $2^n \times 2^n$ -resolution의 영상이라 하자. $A = (Q_\delta, \Sigma, f_\delta, \alpha_\delta, \beta_\delta)$ 는 ϕ 를 정확히 생성하는 automaton 이고, $A_\delta = (Q_\delta, \Sigma, f_\delta, \alpha_\delta, \beta_\delta)$ 를 tolerance δ 를 가지고 찾아진 automaton이라 하자. 그러면,

$$\|\phi - \phi_{A_\delta}\|_{l^2(2^n \times 2^n)} \leq \delta.$$

즉, 원 영상과 A_δ 에 의해 생성된 영상의 차이의 l^2 -norm은 δ 보다 작거나 같다.

감사의 글: 본 논문은 첨단정보기술 연구센터(AITrc)를 통하여 과학재단의 지원을 받았다.

참조 서적

- [1] M. F. Barnsley, *Fractals Everywhere*, Academic Press (1988).
- [2] K. Culik II, S. Dube and P. Rajcaney, *Finite Automata Computing Real Functions*, TR 9105, Univ. of South Carolina (1991).
- [3] K. Culik II, F. Kari, *Image Compression Using Weighted Finite Automata*, Comput. and Graphics Vol.17, No 3, pp.305-313 (1993).
- [4] K. Culik II and J. Karhumaki, *Efficient compression of wavelet coefficients for smooth and fractal like data*, Proceedings of STACS '93, Lecture notes in Computer Science 655, Springer-Verlag, Berlin, 343-353 (1993).