정사각형 매칭을 이용한 비손실 이미지 압축

임성진° 박근수
서울대학교 전기.컴퓨터공학부
{sjyim°, kpark}@theory.snu.ac.kr

Lossless Image Compression Using Square Matching

Sungjin Lim° Kunsoo Park
School of Computer Science & Engineering
Seoul National University

요 약

사전식 압축 방식이라 불리는 LZ-압축은 빠르고도 좋은 압축률을 가지고 있기 때문에 널리 이용되고 있다. 그래서 LZ-압축 방식을 이미지 압축에 적용하는 시도가 이루어지고 있으나 아직 명확하게 정의가 이루어지지 않거나, 정확한 실험 결과가 제시되지 않은 경우가 많다.

이 논문에서는 기존의 정사각형을 이용한 압축 과정 중 다소 모호했던 파싱 과정을 명확히 하며, 매칭에 사용된 정사각형들이 겹쳐지는 비율에 관한 확률적 분석 및 실험 결과를 제시한다. 또한 Test Image Set에 관한 정확한 압축률을 제시한다. 특히 이 논문은 정사각형들이 겹쳐지는 비율에 관한 최초의 확률적 분석을 제시하고 있다.

1. 서 론

사전식 압축 방식이라고도 불리는 Ziv and Lempel 방식[3]은 좋은 압축률과 빠른 압축 속도를 가지고 있기 때문에 실제 압축에 많이 이용되고 있다. 그러나 대부분의 이미지 압축은 이와 달리 변환에 따른 엔트로피 부호화에 의해 이루어진다. 사전식 압축 방식을 이미지 압축에 적용하는 연구가 이루어지고 있으나, 아직은 초기 단계이다.

사전식 압축에는 일반적으로 두가지 주요한 성능 결정 요소가 있는데, 바로 매칭과 파싱이다. Storer[1]는 Giancarlo[2]의 이차원 접미사 트리를 사용하여 직사각형 매칭을 하는 것을 제안하였으나 비실용적인 것이었고, 실제로는 heuristic을 이용하였다[1,4]. 이 논문에서는 실제적인 가능한 매칭 모양을 정사각형으로 한정하고, 파싱에 초점을 두었다. 매칭의 진행순서로는 Storer[1,4]가 몇 가지를 제시하였으나, 다소 모호한 부분을 남겨 놓았다. Rizzo[5]는 매칭된 모양들의 중복 정도가 압축률에 미치는 정도를 연구하였으나, 정작 중복 정도에 관한 분석은 하지 못했다.

본 논문에서는 매칭의 진행 순서를 명확히 하며, 매칭에 사용된 정사각형들이 겹쳐지는 정도를 확률 및 실험적으로 분석한다. 또한 heuristic이 아닌 정확한 매칭방법을 이용했을 때, Test Image들의 압축률을 제시한다.

이후의 구성은 다음과 같다. 2장에서는 본 압축에 관한 문제 정의 및 성능을 결정하는 요소들을 살펴본다. 3장에서는 압축 진행 과정을 기술하며, 4장은 매칭에 사용된 정사각형들의 중복에 관한 확률적 분석을 제시한다. 5장은 구현 방법 및 실험 결과를, 마지막 장에서는 결론을 제시한다.

2. 문제 정의

편의상 이미지는 n*n의 정사각형 모양을 가지고 있고, 이미지의 한 픽셀 값은 0 또는 1의 값을 취한다고 가정한다. v[x][y]는 (x,y)에서의 픽셀값을 의미한다. m-대각선은 {(x,y) : x+y=m}의 점들의 모임이라고 한다. (x,y)가 (a,b)와 k*k 매칭을 이룬다는 것은 v[a+i][b+j] = v[x+i][y+j](모든 $0 \le i,j \le k-1$에 대해)일 때를 말한다. 또 (x,y)가 (a,b)와 유효한 (k*k) 매칭을 이룬다는 것은 위의 조건을 만족하면서 동시에 모든 $0 \le i,j \le k-1$에 대해 (a+i, b+j)가 압축된 영역에 있는 점일 경우를 말하며, 이 때 매칭을 이루는 k*k의 정사각형을 매칭 정사각형이라 부른다.

압축이 진행되면서 압축된 영역과 압축이 되지 않은 부분을 압축 경계선이라 부른다. 그리고, (x-1,y), (x,y-1)의 점이 압축된 부분에 있으나, (x,y)가 압축되지 않은 부분에 있을 때, (x,y)를 Growing Point[1]라고 한다(이후에는 GP로 표기한다.)

S는 매칭에 사용된 모든 정사각형의 넓이의 합이라고 할 때, S/(n*n)를 Redundancy Rate라 정의하고, R이라 표기한다. 서로 겹치는 정사각형이 없을 경우 R = 1이 되는 것을 쉽게 알 수 있으며 가장 이상정인 파싱이 이루어진 경우이다. 그러나 R이 불필요하게 커질 경우, 중복되는 정보를 표현했다는 것을 의미하기 때문에 이상적인 압축에서 점점 멀어지게 된다.

Rizzo[5]는 Redundancy Rate가 압축률에 미치는 영향을 연구하였다. 그 영향을 간단한 식으로 표현하였으며,

몇 가지 실험 증거를 제시하였다.

압축률에 영향을 미치는 또 하나의 중요한 요소는 사용된 정사각형의 개수이다. 한 개의 정사각형은 보통 3~4개의 정보를 coding 해야 한다. 그러므로 사용된 정사각형의 개수를 줄이는 것은 압축 효율에 있어 매우 중요하다. 또한 사용된 정사각형을 어떻게 비트로 표현하는가도 실제 압축률에 큰 영향을 준다.

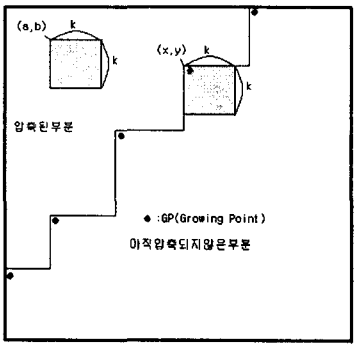

그림 1. 압축 개괄

3. 압축 진행 과정

각 이미지의 픽셀은 그 점이 압축된 영역에 포함되는지 안되는지를 나타내는 정보를 가지고 있다. 그림 2와 같이 m 번째 scan line이 m-diagonal을 따라 가며 아직 압축되지 않은 점, 즉 m-diagonal 상의 GP에서 최대 정사각형 매칭을 찾고, 찾은 영역의 각 픽셀에 대해서 압축되었음을 표기한다. 그리고 다음 (m+1)번 째, scan line에 대해서도 동일한 작업을 반복한다.

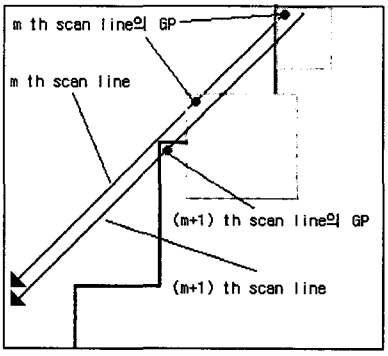

그림 2. 압축 진행 과정

4. Redundancy rate에 관한 분석

현재 GP가 (x,y)이고, x+y = m일 때,

NP(m,k)=$_{def}$ {(x,y)에서 유효한 k*k 매칭이 없을 확률}

($\approx (1-1/2^{k*k})^{0.5m*m}$,픽셀값이 랜덤일 때)

P(m,k) =$_{def}$ {(x,y)에서 최대 k*k 매칭이 일어날 확률}

= (1-NP(m,k)) - (1-NP(m,k+1))

= NP(m,k+1) - NP(m,k)

($\approx (1-1/2^{k*k+2k+1})^{0.5m*m}-(1-1/2^{k*k})^{0.5m*m}$, 픽셀값이 랜덤일 때)

S(m) =$_{def}$ {1~m diagonal에 왼쪽 위 끝을 두고 있는 모든 정사각형의 넓이의 합}이라 하자.

GP(x,y)를 (x,y)가 GP인 사건이라 하자.

P(GP(x,y)|GP(x-1,y)∨GP(x,y-1))는 m이 충분히 클 때, 사실상 0인 성질을 이용하면,

GPN(m)

=$_{def}$ {m diagonal 위에 있는 평균 GP의 개수}

=$\sum_{x+y=m}$P(GP(x,y))

=$\sum_{x+y=m}$P(GP(x,y)|GP(x-1,y)∨GP(x,y-1))*P(GP(x-1,y)∨GP(x, y-1)) + P(GP(x,y)|~GP(x-1,y)∧~GP(x,y-1))*P(~GP(x-1,y)∧~GP(x, y-1))

=$\sum_{x+y=m}$P(GP(x,y)|~GP(x-1,y)∧~GP(x,y-1))*P(~GP(x-1,y)∧~GP(x, y-1))

=$\sum_{x+y=m}$P(GP(x,y)∧~GP(x-1,y)∧~GP(x, y-1))

=$\sum_{x+y=m}$(x-1,y)을 덮는 정사각형이 자신에서 끝나는 확률) * (x,y-1)을 덮는 정사각형이 자신에서 끝나는 확률)

=$\sum_{x+y=m}$((x-1,y)을 덮는 정사각형이 자신에서 끝나는 확률))^2

=$\sum_{x+y=m}$($\sum$ (x-1,y)을 덮는 정사각형의 size가 k*k일 확률)*(1/k))^2

$\approx m(\sum_k P(m-1,k)/k)^2 \approx m(\sum_k P(m,k)/k)^2$

A(m) =$_{def}$ {m diagonal 위에서 시작하는 매칭 정사각형의 평균 넓이}

$=\sum P(m,k)*k^2$

이 되고,

S(m)=S(m-1)+(m diagonal 위의 평균 GP의 개수)*(매칭 정사각형의 평균 넓이)

= S(m-1) + GPN(m) * A(m)

= S(m-1) + $m(\sum(P(m,k)/k))^2 \sum(P(m,k)*k^2)$

$$\therefore S(m) = S(m-1) + m(\sum(P(m,k)/k))^2 \sum(P(m,k)k^2)$$

$$r(m) =_{def} (\sum\{P(m,k)/k\})^2\sum\{P(m,k)k^2\}$$

라 하자.

$\sum$ P(m,k) = 1이고, 1/k, k^2 함수는 아래로 볼록한 함수이기 때문에, 젠센 부등식을 이용하여 r(m)>=1임을 쉽게 알 수 있다. 그리고 모든 m에 대하여 r(m)=1일 때, Redundancy Rate가 1이 되는데, 특정한 k'에 대해

$$P(m,k)\begin{cases} 0 & \text{if } k \neq k' \\ 1 & \text{if } k = k' \end{cases}$$

과 같을 때 일어난다.

그러나 P(m,k) 가 다른 분포를 갖게 되면(예를 들어, P(m,1) = P(m,2) =… P(m,t) = 1/t인 경우) r(m,k)가 m이 커지면서 무한대로 발산하게 된다. r(m,k)는 P(m,k)의 분포에 밀접한 관련을 갖고 있다.

그리고 0~m diagonal위에 왼쪽 위 끝을 두는 정사각형의 개수는

$$N(m) = N(m-1) + m(\Sigma\{P(m,k)/k\})^2$$

임을 쉽게 알 수 있다.

5. 구현 방법 및 실험 결과

Naive한 매칭 방법을 사용할 경우, time complexity는 $O(n^6)$이 된다. Test에 사용되는 이미지들은 보통 2000*2000 정도의 size를 갖기 때문에, naive한 방법으로는 너무 많은 시간이 소요된다. 따라서 다음과 같은 매칭 방법을 사용하였다.

이미지의 각 점에서의 4*4의 sub array의 hashing 값에 따라 65536개의 bucket를 갖는 hash를 구현하였고, 각 bucket은 연결 리스트를 가지고 있어서 slot을 무한히 가질 수 있도록 하였다. 그리고 bucket의 연결 리스트에 대하여 정렬을 수행하고, LCP(Longest Common Prefix)를 구하도록 하였다. 특정 이미지에서는 것은 어떤 해싱 값 entry에 여러 정보가 중복되는 경우가 많기 때문에, LCP를 이용하여 평균적으로 빠르게 매칭을 찾도록 한 것이다. 실제 CCITT Test Image Set[6]에서 어떤 entry에는 10000개 넘는 중복이 일어나기도 하기 때문에 이와 같은 처리는 필수적이다.

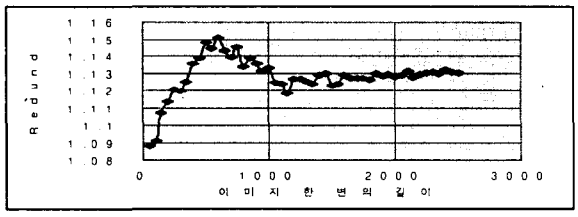

그림 3. Random Image에서 Redundancy Rate 변화

Random Image에서 R은 대부분 1.1~1.16의 값을 갖는 것을 관찰하게 되었다. 그러나 실제의 이미지에서는 1.6 정도의 값을 갖게 되어 앞에서 분석한 것과 같이 P(m,k)가 특정 k에 대해서 거의 1에 가까운 값을 갖을 때, R이 1에 가까운 값을 갖게 되고, 반대의 경우에는 큰 값을 갖게 되는 것을 관찰할 수 있다.

실제 압축은 매칭 정사각형이 가리키는 포인터와 정사각형의 크기를 naive한 방법으로 인코딩한 방법을 사용하였다. 그리고 0으로만 이루어진 정사각형과 1로만 이루어진 정사각형은 별도로 인코딩하였다.

압축률을 (압축 전 파일 크기)/(압축 후 파일 크기) 로 정의하면 Test Image Set[6]에 대하여 13.6 ~ 21.8 정도의 압축률을 보이게 되었다.

6. 결론

LZ 방식이 적용된 압축 방식은 아직 연구할 것이 많은 분야이다. 직사각형 매칭이 이미 소개되었지만 직사각형 매칭이 이론 및 실제적으로 어렵기 때문에 정사각형 매칭은 여전히 이러한 압축 방식의 근간을 이루게 된다. 기존의 논문에서는 압축의 진행 과정이 다소 모호하게 기술되었다. 또한 정사각형 매칭에 대한 heuristic한 구현만 이루어졌기에 정확한 매칭이 이루어졌을 때의 압축률이 기록되지 않았다. 그리고 Redundancy Rate에 영향을 미치는 요소에 관한 분석은 이루어지지 않았다.

본 논문에서는 정사각형 매칭의 진행 방향을 명확히 하였고, Redundancy Rate를 k*k의 매칭이 일어날 확률에 관한 함수로 표현하였다. 여러 경우에 대한 Rundancy Rate를 살펴 보았다. 따라서 이 논문은 이후에 파싱에 관해 이루어질 연구에 대해 귀중한 가이드 라인을 제시한다.

7. 참고 문헌

[1] J. A. Storer and H. Helfgott, "Lossless Image Compression by Block Matching" , The Computer Journal 40:2/3, 137-145., 1997.

[2] R. Giancarlo, R. Grossi, "On the Construction of Classes of Suffix Trees for Square Matrices: Algorithms and Applications" , Information and Computation 130, 151-182, 1996.

[3] T. Bell, J. Cleary, I. Witten, Text Compression, Prentice Hall, Englewood Cliffs, NJ., 1990.

[4] C. Constantinescu, J. A. Storer, "Improved Techniques for Single-Pass Adaptive Vector Quantization." , Proceedings of the IEEE, Vol. 82, No. 6, pp. 933-939., 1994.

[5] F. Rizzo and J. A. Storer, "Overlap in Adaptive Vector Quantization", Proceedings Data Compression Conference, IEEE Computer Society Press, 401-410.,2001.

[6] CCITT International Telegraph Consultive Committee-CCITT. Standardization of Group 3 Facsimile Apparatus for Document Trasmission, Recommendation T.4. Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, Recommendation T.6.,1980,1984.