비트 매트릭스 확률모델과 이진 산술부호기 설계

이 호석, 이 제명
호서대학교 컴퓨터공학부
전화 : +041-540-5710, 전자우편 : hslee@office.hoseo.ac.kr

Design of bit matrix model and binary arithmetic coder

Ho Suk Lee, Je-Myung Lee
Dept. of Computer Engineering, Hoseo University
Tel : +041-540-5710, E-mail : hslee@office.hoseo.ac.kr

요약

본 논문은 비트 매트릭스(bit matrix) 확률 모델과 이를 입력으로 사용하는 개량된 이진 산술부호 알고리즘을 제안한다. 비트들로 이루어진 비트 평면에서 3x3 비트 매트릭스를 정의하였다. 그리고 비트 평면을 조사하여 2연속 혹은 3연속 비트 매트릭스들에 대한 확률모델을 구하였다. 본 연구에서는 3 가지의 확률간격(interval)을 가지는 개량된 이진 산술부호기를 사용하였다. 개량된 이진 산술부호 알고리즘의 장점은 구조가 간결하고 또한 부호화가 진행되는 도중에 결과 비트스트림을 생성하는 특징이 있다. 이진 산술부호기는 2연속 혹은 3연속 비트매트릭스를 입력하여 산술부호화를 수행하도록 한다.

1. 서 론

멀티미디어 시대에 영상 정보는 매우 큰 비중을 차지하게 되어, 방대한 양의 영상 정보를 손실 없이 빠르게 저장하고 전달하고자 영상 압축 알고리즘이 개발되어 왔다. 이러한 알고리즘 가운데 DCT 기반의 JPEG이 많이 사용되어 왔다. 그러나 오늘날 압축율의 한계성으로 인하여 차세대 영상 압축의 표준으로 JPEG2000[1]이 개발되었다. JPEG2000은 영상 분해를 위하여 웨이블릿 변환[1,2,3]을 사용하고 비트스트림 생성을 위하여 산술부호 방식을 사용한다.

JPEG2000은 웨이블릿 변환에 의하여 생성된 웨이블릿 계수들을 비트 평면으로 구성하고, 비트평면에 대하여 산술부호화[4,5]를 수행하여 최종적인 비트스트림을 생성한다. 따라서 JPEG2000에서 압축율 향상에 관련이 많은 부분은 웨이블릿 변환과 산술부호화 부분이다. 그러나 일반적으로 웨이블릿 변환 부분에 대하여서는 언급이 많이 되고 있으나, 산술 부호화 부분에 대하여서는 많이 언급 되고 있지 않다. 사실 JPEG2000은 거대한 산술부호기라고 해도 과언이 아니다.

본 논문은 현재 멀티미디어 연구실에서 개발중인 JPEG2000에 사용할 목적으로 비트 매트릭스 모델과 이진 산술 부호기의 설계에 대하여 설명한다. 비트평면을 조사하여 3x3 비트 매트릭스의 종류를 파악하였다. 실험에서 사용한 AT&T 영상에 대하여 모두 512가지 종류의 비트 매트릭스를 파악하여 누적 분포를 구하였다. 가장 많은 분포를 나타내는 비트 매트릭스는 모든 값이 0 값을 가진 비트 매트릭스였으며 50,359개가 발견되었다. 1이 하나 존재하는 비

트 매트릭스는 5가지였으며 가장 큰 개수는 695였다. 모든 값이 1 값을 가진 비트 매트릭스는 646개였다.

2. 본 론

2.1 비트 매트릭스와 확률 계산

다음은 파악된 3x3 비트 매트릭스의 일부분이다.

0	0	0
0	0	0
0	0	0

다음에 이러한 3x3 비트 매트릭스 2개 혹은 3개를 연속으로 구성하여 비트평면에 대하여 확률을 계산한다. 확률은 발생 개수를 구하여 계산하는 방법이 있고, 조건부 확률을 구하여 계산하는 방법이 있다. 확률 사건 C_1과 C_2에 대한 조건부 확률식은 다음과 같다.

$$P(C_2 \mid C_1) = P(C_1 \cap C_2) \,/\, P(C_1)$$

확률 사건 C_1, C_2, C_3에 대한 조건부 확률식은 다음과 같다.

$$P(C_3 \mid C_1 \cap C_2) = P(C_1 \cap C_2 \cap C_3) \,/\, P(C_1)\, P(C_2 \mid C_1)$$

2.2 산술부호 알고리즘

ISO/IEC의 JPEG2000은 IBM에서 개발한 Q-coder를 사용하여 구현되었다. 기존의 산술부호기는 다음과 같은 단점들이 있다.

(1) 산술부호를 수행하는 과정에서 확률간격 계속 좁아져, 높은 정밀도의 계산이 필요하게 된다.
(2) 확률간격 계산을 수행하기 위하여 곱셈을 사용하기 때문에 속도가 느리게 된다.
(3) 모든 기호들을 마지막까지 완전히 읽어서 처리할 때까지 비트가 생성되지 않는다.

이러한 문제점들을 극복하기 위하여, 산술부호 알고리즘[4,5]을 다음과 개량하였다.

- 개량된 이진 산술부호 알고리즘 -

(1) 확률의 낮은(low) 값을 0으로, 높은(high) 값을 65535로 한다. Range = (high-low)+1을 구한다. 입력 기호의 낮은(low) 값과 높은(high) 값을 계산한다.
(2) 새로운 확률간격이 [0,1/2), [1/4,3/4), [1/2,1)에 완전하게 포함되지 않으면, 산술부호를 완료하고 생성된 비트스트림을 출력한다.
(3) 만약, 새로운 확률간격이 모두 [0, 1/2)에 포함되면 0을 출력하고, 이전 기호를 처리하고 나서 출력되지 않고 산술부호기속에 남아있는 1을 출력한다. 확률간격 [0,1/2)을 오른쪽 방향으로 2배 확대한다.

(4) 만약, 새로운 확률간격이 모두 [1/2,1)에 포함되면 1을 출력하고, 이전 기호를 처리하고 나서 출력되지 않고 산술부호기속에 남아있는 0을 출력한다. 확률간격 [1/2,1)을 왼쪽방향으로 2배 확대한다.
(5) 만약, 새로운 확률간격이 모두 [1/4, 3/4)에 포함된다면, 미래의 출력을 위하여 이 사실을 기억하고, 확률간격 [1/4, 3/4)을 가운데 방향으로 2배 확대한다.

여기서, 단계 (5)에서 출력되는 비트는 이전 단계에서 0을 출력하였으면 반대로 1을 출력하고 1을 출력하였으면 반대로 0을 출력하여 서로 반대로 출력하도록 한다.
이러한 방식의 산술부호화 알고리즘은 위의 3가지 문제점들을 극복하여 확률간격이 좁아지지 않으며 계산이 간단하고 또한 수행도중에 비트들을 출력시킨다.

- 이진 산술복호 알고리즘 -

(1) 확률의 낮은(low) 값을 0으로, 높은(high) 값을 65535로 한다. Range = (high-low)+1을 구한다. 입력 값의 확률을 이용하여 출력 기호를 찾는다. 출력 기호의 낮은(low) 값과 높은(high) 값을 계산한다.
(2) 만약, 새로운 높은(high) 값이 1/2보다 작다면, low = 2x low, high=2x high를 계산하고 다음 입력 비트를 읽어 들인다.
(3) 만약, 새로운 낮은(low) 값이 1/2보다 크거나 같다면 입력 값, 낮은(low) 값, 높은(high) 값에서 모두 1/2을 감한다. 다음에 low = 2x low, high = 2x high를 계산하고 입력 비트를 읽어 들인다.
(4) 만약, 새로운 낮은(low) 값이 1/4보다 크거나 같고 높은(high) 값이 3/4보다 작다면 입력 값, 낮은 값, 높은 값에서 모두 1/4를 감한다. 다음에 low = 2x low, high = 2x high를 계산하고 입력 비트를 읽어 들인다.
(5) 위의 (2),(3),(4) 조건식을 만족시키지 않으면 산술복호를 중단하고, 복호된 기호를 출력한다.

2.3 비트 매트릭스 입력

위에서 설명한 2연속 혹은 3연속 비트매트릭스를 개량된 산술부호기의 입력으로 한다. 비트매트릭스 입력을 위해서는 비트매트릭스들을 서로 비교하는 비트매트릭스 비교 모듈이 필요하다. 비트매트릭스와 비트매트릭스의 확률은 테이블 형태로 저장한다. 비트평면을 읽으면서 비트매트릭스들을 테이블에서 탐색하여 확률값을 구하여 산술부호기의 입력으로 한다. 각 비트평면당 비트스트림을 구하는 방법은 여러 가지를 생각할 수 있다. 하나는 전체 비트평면에 대하여 산술부호를 적용할 수도 있고, 다른 하나는 비트평면을 서브블록으로 구분하고, 각 서브블록

3. 결 론

이 논문에서는 비트매트릭스의 구성과 확률모델링, 개량된 이진 산술부호기 설계, 그리고 비트매트릭스 입력 방법에 대하여 제안하였다. 2개 혹은 3개의 연속된 비트매트릭스는 입력기호의 문맥(context)으로 생각할 수 있다. 처리방법은 2단계 방식을 사용하여, 우선 전체 비트매트릭스와 확률값을 구하고 다음에 산술부호화를 적용하는 방식으로 설계하였다. 이 방식

이 구현이 용이하고 확률값을 일괄적으로 구하기 때문에 더욱 최적화된 산술부호화 결과를 낼 수 있다.
  그리고 유사한 비트매트릭스의 통합을 통하여 더욱 높은 압축을 기대할 수 있다.

참고문헌

[1] David Taubman, Michael Marcellin, JPEG2000, Kluwer Academic Publishers, 2002.
[2] Michael David Adams, Faouzi Kossentini," Reversible Integer-to-Integer Wavelet Transforms for Image Compression : Performance and Evaluation, " IEEE Trans. on IP, Vol. 9, No. 6, pp. 1010-1024, June 2000.
[3] Wim Sweldens, "Building your own wavelets at home," In Wavelets in Computer Graphics, ACM SIGGRAPH Course Notes, 1996.
[4] Paul G. Howard, Jeffrey Scott Vitter, Practical Implementations of Arithmetic Coding," Technical Report No. CS-91-45, Dept. of Computer Science, Brown Univ. 1992.
[5] I. Witten, R. Neal, J. Cleary, "Arithmetic Coding for Data Compression," CACM 30, pp. 520-540, June 1987.
마다 산술부호를 적용할 수도 있다.