

임베디드 환경의 효율적인 Motion-JPEG 스트리밍 기법

김용진, 이동식, 고미애, 김영모
경북대학교 전자공학과
e-mail : zini08@palgong.knu.ac.kr

Effective Streaming System for Motion-JPEG on Embedded Environment

Yong-Jin Kim, Dong-Sik Lee, Mi-Ae Ko, Young-Mo Kim
Dept. of Electrical Engineering and Computer Science,
Kyungpook National University

요 약

본 논문에서는 제한요소가 많은 임베디드 환경에서의 실시간 Motion-JPEG 전송을 초점으로 하고 있으며, 현재에 얻어진 이미지와 이전의 이미지를 비교하여 움직임이 있는 부분의 블록만을 전송하는 시스템을 제안한다. 제안한 시스템에서는 처리시간과 처리 데이터 용량에서 이득을 얻기 위해 DCT 영역에서 두 영상을 블록 단위로 비교 한다. 영상비교에 필요한 임계값 추출은 물체가 이미 지상에서 여러 픽셀과 블록에 연속적으로 존재하는 현상을 이용하여 계산된다. 비교 시 DCT 연산 자체의 특징인 저주파로의 에너지 패킹현상을 이용하며 전체 DCT 계수들을 비교하지 않고 DC 성분을 포함한 일정 개수의 AC 성분만 사용한다.

1. 서론

지식 정보화 사회가 발달함에 따라 처리해야 하는 데이터의 용량이 점차 증가하고 있다. 이에 따라 대용량의 정보 처리 및 저장, 검색에 필수적인 데이터 압축기술이 등장하게 되었다. 특히 영상의 전송이나 저장의 경우 처리에 필요한 데이터의 용량이 다른 데이터에 비해 월등히 많으므로, 제한된 하드웨어 자원의 효율적인 이용과 전송속도 개선을 위한 연구가 예전부터 있어왔다. 그 결과로 JPEG, MPEG 등의 압축에 대한 국제 표준이 나왔으며, 이러한 기술을 기반으로 하여 임베디드 시스템에 적용시킨 많은 제품들이 나오고 있다. 그 예로서 PC 카메라를 이용한 인터넷 방송에서부터, 디지털 카메라, 캠코더 등의 가전기기, 사무실, 은행, 주택의 원격 모니터링 등을 일상생활에서 쉽게 접할 수가 있다.

위 압축기술 중 JPEG은 정지영상을 대상으로 했기 때문에 실시간 영상을 스트리밍 하기에는 처리해야 하는 데이터의 용량이 많아서 적합하지 않지만, 동영상

을 기반으로 하는 MPEG의 경우에 그 알고리즘의 구현을 여러 제한요소를 가지는 임베디드 환경에서 소화시키기엔 힘들다. 그래서 JPEG을 동영상으로 확장 적용한 M-JPEG(Motion JPEG)을 이용해서 상용화된 제품이 더 많은 것이 현실이다.

본 논문에서 제안하는 시스템은 M-JPEG을 대상으로 했으며, 실시간 영상 전송을 위해서 이전 영상과 현재 영상을 DCT 영역에서 비교하여 움직임이 있는 블록을 찾아내고 그 부분만을 압축처리 한 후 전송한다. 영상복원 시에는 이전 블록의 정보와 전송된 데이터를 이용하여 이전영상을 변화시킴으로써 전송하는 데이터의 용량을 줄일 수 있다. 이로 인해 M-JPEG을 이용하는 임베디드 환경의 한정된 자원을 효율적으로 사용 가능하게 하며, 실제 상용화 되어 있는 여러 제품에 적용시키기도 수월하다.

게다가 이전 영상과의 차이영상을 얻어내는 과정에서 JPEG 압축의 디코딩 과정 중 가장 많은 시간을 요구하는 DCT 및 I-DCT를 하지 않고 DCT 영역에서 두 영상을 블록 단위로 비교하므로 차이영상의 추출에 필요한 처리시간과 처리용량을 감소시킬 수 있다.

2. 본론

본 논문에서 제안하는 시스템에서 두 영상을 비교해서 서로 다른 영역을 찾아내는 연산은 DCT 영역의 각 블록 별로 이루어 지게 된다.

DCT 영역에서의 영상처리는 그 대상 이미지의 압축을 풀어서 각 픽셀 단위로 영상처리를 한 후 다시 압축을 해야만 하는 공간영역의 영상처리 알고리즘보다 50~100 배 정도의 시간 단축효과를 얻을 수 있다.

이런 시간 단축은 영상처리를 할 때 다루어야 하는 데이터의 용량을 줄임으로써 가능하며, 이것은 처리에 필요한 메모리의 감소와 더불어 전송 대역폭의 감소 등의 효과를 가지고 온다[1][2][3]. 더불어 이 알고리즘은 DCT 를 기반으로 하고 있는 현재의 많은 영상표준에 쉽게 접목시키는 것이 가능하여 여러 분야에 적용이 가능하다.

2.1 시스템 구성

전체 시스템은 크게 4 부분으로 구성된다(그림[1]).

- a. $JPEG_{t(n)}$ 과 $JPEG_{t(n+1)}$ 파일에 엔트로피 디코딩을 한 다음, 두 JPEG 파일에서 비교에 필요한 헤드정보와 DCT 계수를 추출하는 디코딩 부분.
- b. $JPEG_{t(n)}$ 과 $JPEG_{t(n+1)}$ 의 DCT 계수 비교 시 사용할 블록의 임계값을 추출 알고리즘을 통해서 결정하는 임계값 추출부분.

c. DCT 계수의 값을 블록단위로 비교해서 DCT 계수의 차이가 임계값을 초과할 경우와 그렇지 않을 경우를 처리하는 비교부분.

d. 비교가 끝난 출력파일의 압축 및 전송을 담당하는 압축부분.

2.2 각 블록 당 DCT 계수의 비교 수 조절

영상에 DCT 변환을 취하면 DCT 변환의 특성인 저주파로의 에너지 패킹 현상으로 인해 DC 성분이 가장 많은 에너지를 가지고 있으며 고주파수로 갈수록 지니고 있는 에너지가 적어진다. 따라서 두 영상에서 서로 다른 부분을 찾을 때 DC 성분을 이용하는 것이 가장 일반적이며 이것은 이미지의 특징 추출[4], 움직임 추정[5]등의 많은 부분에 응용 되고 있다.

그렇지만 DC 성분은 원래의 이미지에서 영상의 평균 주파수를 나타내기 때문에 만일 이미지에서 물체가 비교하고자 하는 블록영역의 가장자리 영역에 있다면 그 영상의 평균 주파수에 미치는 영향은 현저하게 줄어든다. 이런 현상은 영상의 비교에 나쁜 효과를 가지고 오게 되고, 이것은 실제로 이미지를 복원했을 때 물체의 경계부분이 블록단위로 깨어져서 매끄럽지 못하게 보이는 현상을 초래하게 된다.

따라서 비교를 할 때 되도록 DC 성분을 포함한 많은 고주파 성분을 사용하는 것이 정확한 결과를 얻을 수 있지만, 비교해야 하는 DCT 계수의 양이 많을수록 처리시간이 증가하며 알고리즘의 구현이 복잡해진다.

실제 시스템 구현에 있어서는 서로가 상충관계에 놓이며 본 논문에서는 5 개의 성분(DC~ AC₄)의 DCT 성분 값을 블록 비교에 이용한다.

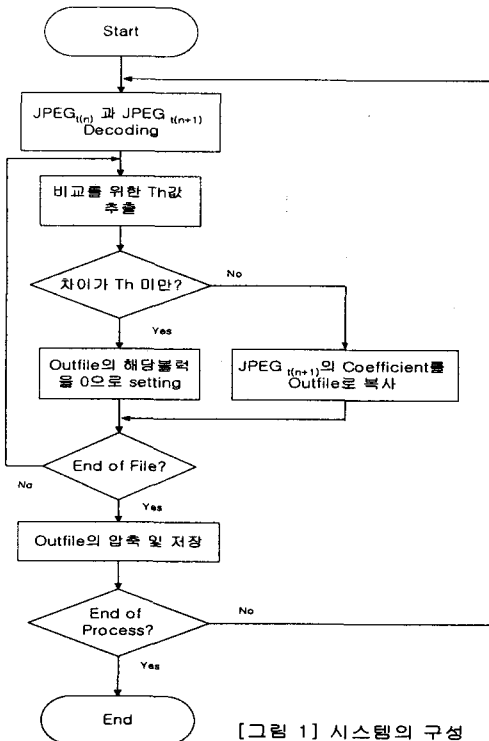
2.3 대상 블록의 경계 블록을 이용한 임계값 조절

통계적으로 물체는 공간영역의 이미지 내에서 특정 픽셀이나 블록에만 존재하는 것이 아니라 여러 픽셀과 블록에 걸쳐 연속적으로 존재한다. 이것은 DCT 변환이 취해진 다음에도 마찬가지이며 비교를 하고자 하는 블록뿐만 아니라 주변 DCT 블록의 계수성분들의 크기도 영향을 끼친다[6][7].

본 논문에서는 두 영상에서 다른 부분을 찾아내기 위해서 우선 그림[2]의 (a)와 같이 서로 다른 시간에 얻어진 이미지에서 DCT 블록성분의 차(D: Difference of Block)를 구한다.

두 이미지를 비교했을 때 서로 다른 부분일수록 그 부분에 대한 차이 값(D)는 커지며, 본 논문에서는 이 블록간의 차이가 일정한 임계값(Th₁)를 넘을 때 그 블록을 이전 영상과 달라진 부분으로 가정한다.

비교하고자 하는 블록의 임계값(Th₁) 추출 연산은 일정한 기준 값(Basis)에서 주변 블록 중 변화가 있는 블록의 개수를 감소 시킴으로써 이루어진다(그림[2]의 (b)). 이것은 주변 블록 중에 변화된 블록이 많을수록 현재 블록이 변화된 블록일 확률이 높다는 사실을 이용한 것이다.



[그림 1] 시스템의 구성

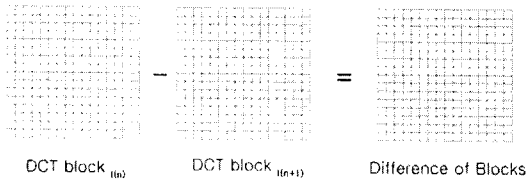
$$Th_1 = \text{Basis} - \text{Sum}$$

Basis: 실험으로 정해진 기준값

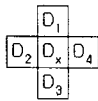
Sum: 주변블록 중 Th_2 를 넘는 블록의 개수

임계값(Th_1) 지정 시 참조로 하는 D의 개수를 조절하면서 임계값(Th_1)의 정확도를 조절할 수 있다. 일반적으로 D의 개수가 많을수록 영상의 잡음에 덜 민감하지만 그것은 곧바로 계산량의 증가를 가지고 오게 되며 전체 수행시간에 영향을 끼친다.

본 논문에서는 비교를 하고자 하는 블록 주위의 4개 블록을 참조했다. 알고리즘에서는 주변블록을 참조할 때 임계값(Th_2)를 조절하면서 임계값(Th_1)의 민감도를 조절할 수 있다. Th_2 가 낮을수록 복원된 영상이 더 깨끗하지만 잡음에 영향을 받을 수 있으며, Th_2 가 높으면 잡음에는 영향을 덜 받지만 복원된 영상에서 깨진 부분이 생길 수 있다.



(a) 블록계수간의 차이



$$Th_1 = \text{Basis} - \text{Sum}$$

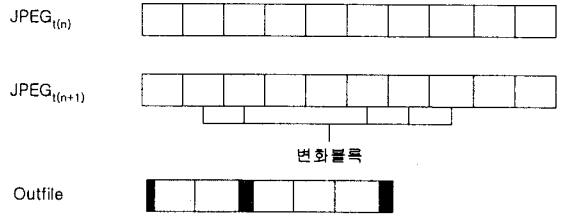
(b) Th 추출시 사용되는 마스크

[그림 2] 임계값 선정

2.4 차이 영상의 저장

DCT 영역에서의 비교연산 후 그 결과들은 엔트로피 코딩을 거쳐 M-JPEG의 형식으로 압축이 된다.

우선 $JPEG_{(n+1)}$ 의 중요 헤더정보들을 미리 결과파일에 저장해 놓는다. 영상의 비교연산 후 그 영역이 변화가 있다고 가정되는 블록일 경우 DCT 계수의 값을 결과파일로 옮긴다 (그림 [4]의 (c)참조). 비교 부분이 이전 영상과 비교해서 변화된 영역이 아닐 경우에는 결과파일의 DCT 계수 값을 0으로 채워 넣는다. 이 블록들은 엔트로피 코딩 도중 Jpeg 내부 Marker 중의 하나인 EOB(End of Block)로 바뀌게 된다(그림 [3]). 복원에서는 이 결과파일의 DCT 계수 값을 계산한 다음, DCT 계수가 모두 0인지 검사한다. 블록내부 계수의 모든 값이 0이 아니면 변화가 있는 영역으로 간주하고 바뀐 DCT 계수들(그림 [4]의 (c))을 이용하여 저장되어있는 영상(그림 [4]의 (a))을 변화시킨다.



[그림 3] 결과 파일의 생성

3. 실험결과 및 고찰

실험영상은 카메라에서 직접 들어오는 640*480 크기의 JPEG 영상을 기준으로 했으며 JPEG 파일의 압축과 해제는 IJG(Independent JPEG Group)의 라이브러리를 사용했다.



(a) $Jpeg_{(n)}$

(b) $Jpeg_{(n+1)}$



(c) Difference Image

(d) Result Jpeg

[그림 4] 실험영상

	실험 1	실험 2	실험 3
영상의 변화량	0.3%	14.197%	66.45%
수행시간	10.5ms	10ms	10.2ms
출력파일의 상대적인 크기	17.4k/58.7k (29.64%)	23.8k/58.8k (40.47%)	46.1k/54.4k (84.74%)
MSE/PSNR	0/inf	6.9/39.73	1.95/45.21

[표 1] 실험결과

실험에서 Th_1 값과 Th_2 값은 각각 5와 1을 사용했으며, 임계값 추출 때 참조하는 주변 블록의 개수는 4개로 한정했다(그림 [2]의 (b)). 임계값을 추출 후 각 블록

간에 비교하는 DCT 계수는 DC-AC₄ 의 5 개 성분을 이용했다.

복원된 영상의 평가는 실제영상과 복원된 영상의 인간의 시각에 의한 평가와 PSNR 을 구함으로써 이루어졌다. 영상의 복원 후 생기는 화질열화는 시각적으로 확인하기 힘들며(그림[4]의 (b)와 (d)), PSNR 의 경우 영상에 따라 평균 35.8-47.52 정도의 분포를 가진다. 대상이 되는 영상의 크기가 일정할 때 수행시간의 변화는 그 영상의 변화 정도에 관계없이 거의 일정하다(표 [1]). 출력파일의 크기는 영상에서 변화된 부분이 없을 경우, 원래 이미지 파일 크기의 30%정도로 줄어들며(표[1]의 실험 1), 영상의 변화가 있는 경우에도 파일크기가 크게 감소하는 것을 확인할 수 있다.

4. 결론

본 논문에서는 하드웨어적으로나 소프트웨어적으로 많은 제약이 따르는 임베디드 환경에서 M-JPEG 이미지를 효율적으로 전송하는 시스템을 제안했다. 영상의 비교 시, 현재 이미지와 이전의 이미지를 DCT 영역에서 계산하여 움직임이 있는 블록만을 전송하는 방법을 사용했다. 처리시간과 처리 데이터 용량을 줄이기 위해 DCT 영역에서 두 영상을 비교 했으며, 영상비교에 필요한 임계값 추출은 물체가 이미지상에서 여러 픽셀과 블록에 연속적으로 존재하는 현상을 이용하여 주변블록들을 참조하여 계산되었다. 이때 DCT 연산 자체의 특징인 저주파로의 에너지 패킹현상을 이용하여 모든 DCT 계수들을 비교하지 않고 DC 성분을 포함한 일정 개수의 AC 성분만 비교에 사용했다.

본 논문에서 제안하는 시스템은 JPEG 을 기반으로 제품화 되어있는 여러 분야에 쉽게 적용이 가능하다. 게다가 전송 혹은 저장해야 하는 이미지 데이터의 용량을 줄임으로써 전송속도의 향상을 가지고 오며, 한정된 하드웨어 자원의 효율적인 사용을 가능하게 한다.

참고문헌

- [1]B. Smith and L.A.Rowe, "Algorithms for Manipulating Compressed Images", IEEE Computer Graphics and Applications 13, 1993
- [2]B.Chen and I.K Sethi, "Inner-block Operation on compressed image" ACM Multimedia 95, vol.3 1995
- [3]Merhav N, Bhaskaran, "Fast Algorithms for DCT Domain Image Down-sampling and for Inverse Motion", IEEE Transactions on Circuits & Systems for Video Technology, 1997
- [4] Sung-Kon Oh, "A Study on Image Processing in DCT Domain", School of Engineering Information and Communications University, July.3. 2000
- [5]Kwon Cheol Lee, "A Study on Motion Estimator Design Using DCT DC Value", 전자공학회 2001
- [6]Sang Gil Lee, "Bit-rate Reduction by

Interleaving DCT Coefficients for Differential Images", 전자공학회 1993

[7]Hyung-jin Jeon, "A DCT Coefficients Error Concealment by Using Adjacent Blocks in Spatial Domain", Department of Electrical Engineering, 1994

[8]후지와라 히로지, "최신 MPEG" 교보문고 1995