JPEG 영상 복원에서 컬러변환의 계산 최적화

김영주*
*신라대학교 컴퓨터정보공학부
e-mail: yjkim@silla.ac.kr

Computation Optimization of Color Conversion in JPEG Image Decoding

Young-ju, Kim*
*Division of Computer Information, Silla University

요 약

최근 모바일폰에 500만 화소 이상의 카메라 모듈이 장착되는 등 모바일 장치에서 고해상도 영상의 인코딩 및 디코딩에 대한 요구가 크게 늘어남에 따라 저성능 시스템에서 실시간으로 동작하는 영상 코덱 구현에 대한 필요성이 증대되고 있다. 본 논문은 JPEG 디코딩의 마지막 단계인 컬러변환 과정에 대해 계산 복잡도를 최적화하는 기법을 제안하고 성능을 평가하였다. 제안된 기법은 JPEG 디코딩 과정에서 IDCT(Inverse Discrete Cosine Transform) 변환과 컬러변환 간의 선형성을 바탕으로 이들 연산 순서를 재배열함으로써 컬러변환 과정에서 요구되는 계산 횟수를 줄이고, 재배열된 부동소수점 연산에 대해 정수 맵핑을 적용하여 계산 복잡도를 줄임으로써 실행시간을 최적화하였다. 임베디드 시스템 개발 플랫폼에서의 성능 평가를 통해 제안된 기법이 기존의 컬러변환 기법들과 비교하여 실행시간을 크게 단축함을 알 수 있었으나 복원 영상의 화질이 상대적으로 저하됨을 확인하였다.

키워드 : 모바일 장치, JPEG 디코더, YCbCr-RGB 컬러변환, 연산재배열, 정수 맵핑

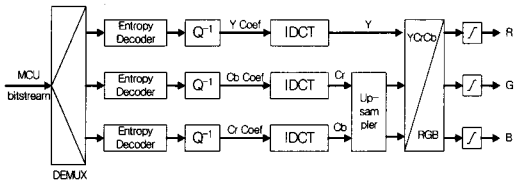
I. 서론

최근 컴퓨터 및 이동통신 기술의 발달로 PDA, 모바일폰 등 모바일 장치가 크게 발전하여 그 시장을 확대하고 있다. 또한 모바일 장치에서의 주요 응용으로는 영상통화, 정지영상 및 MP3 디스플레이 등 멀티미디어 데이터 처리가 주를 이루고 있고, 고해상도 영상 처리 등 고용량 데이터 처리 요구가 지속적으로 증대되고 있다. 이는 저성능 모바일 장치에 큰 시스템 부하를 야기하며, 따라서 모바일 장치에서의 멀티미디어 데이터 처리를 위해 효율적인 알고리즘 개발이 요구된다. 본 논문은 모바일 장치에서 JPEG 압축 영상의 빠른 복원을 위해 디코딩 과정의 마지막 단계인 컬러변환 과정의 계산 복잡도를 최적화하는 기법을 제안한다.

JPEG 표준[1]에서 기술하고 있는 전형적인 JPEG 영상 디코딩 과정은 그림 1과 같으며, 주요 구성 요소별 상대적인 계산 요구량을 측정한 결과는 표 1과 같다. 이는 Intel Pentium-II 프로세서에서 소프트웨어 JPEG 디코더의 연산 요구량을 분석한 것이다. 계산량을 가장 많이 요구하는 IDCT 연산 과정이나 엔트로피 디코딩 및 역양자화 연산 과정에 대해서도 다양한 연구에서 최적화 기법이 제시되었다[3]. 컬러변환 과정은 JPEG 디코딩 과정에 가장 많은 데이터를 처리하는 단계로서 많은 데이터 처리를 위한 CPU와 메모리사이의 전송 지연 및 부동소수점 연산 등으로 전체 복원 시간 중에 15~25%의 높은 비중을 차지하는 반면에 최적화 연구는 부족한 형편이다[4]. 최근 Yang[5]이 rounding 오류로 야기되는 화질 왜곡이 허용 가능한 범위 내에 있도록 부동소수점 연산에 정수 맵핑(integer mapping) 기법을 적용하여 빠른 컬러변환을 지원하는 기법을 제안하였지만 단지 컬러변환 연산 자체의 계산 복잡도만을 고려하고 있다.

본 논문은 JPEG 디코딩 과정에서 IDCT 변환과 컬러변환 간에 존재하는 선형성을 바탕으로 이들 연산 순서를 재배열함으로써 컬러변환 과정에서 요구되는 연산횟수를 근본적으로 줄이고 재배열된 부동소수점 연산에 대해 정수 맵핑을

적용함으로써 컬러변환의 전체 계산 복잡도를 줄이는 최적화 기법을 제안하고 성능을 평가하였다.



〈그림 1〉 전형적인 JPEG 디코더 구조

〈표 1〉 JPEG 디코더의 구성 요소별 계산 요구량 분석

연산 과정	MOPS
엔트로핑 디코딩	17
역양자화	9
Inverse DCT	51
채도 재구성(업샘플링)	12
컬러 변환	24

II. 관련 연구

최근 JPEG-2000, MPEG-4 AVC/H.264 등의 고성능 영상 코덱 표준의 개발 과정에서 다양한 형태의 새로운 컬러 변환 기법인 ORCT, RDCT, YCoCg-R, 그리고 YSbSr 등이 제안되었다[6]. 이들 컬러변환 기법은 컬러변환의 가역성(reversiblity)에 의한 오류 최소화와 계산 복잡도 최소화에 초점이 맞추어져 있다. 본 논문은 JPEG 코덱의 디코딩 과정을 최적화하기 위해 JPEG 표준에서 채택한 RGB 및 YCbCr 컬러공간사이의 변환만을 연구 범위로 한다.

RGB와 YCbCr 사이의 컬러변환은 일반적으로 ITU-R Recommendation BT.601-5 표준[2]에 제정되어 있는 변환 과정을 적용하며, 그 변환 과정은 식 (1)과 같다.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.499 \\ 0.499 & -0.418 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad \cdots\cdots (1)$$

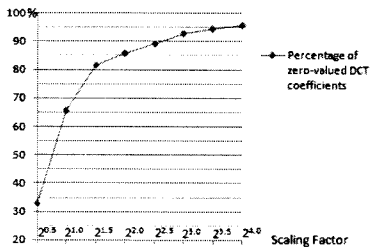
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.3444 & -0.714 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb-128 \\ Cr-128 \end{bmatrix}$$

식 (1)의 변환 과정을 구현하기 위해 다양한 알고리즘들이 제시되었으며, 특히 Yang[5]은 컬러변환 연산의 양자화 오류로 야기되는 복원 영상의 왜곡 범위를 분석하고, 정의된 왜곡 범위 내에서 정수 맵핑(integer mapping) 기법을 사용하여 정수 연산과 쉬프트 연산만을 이용한 빠른 변환 알고리즘을 제안하였으며, 변환 과정은 식 (2)와 같다.

$$\begin{aligned} R &= Y + Cr + Cr \gg 2 + Cr \gg 3 + Cr \gg 5; \\ G &= Y - (Cb \gg 2 + Cb \gg 4 + Cb \gg 5) \\ &\quad - (Cr \gg 1 + Cr \gg 3 + Cr \gg 4 + Cr \gg 5); \\ B &= Y + Cb \gg 1 + Cb \gg 2 + Cb \gg 6; \end{aligned} \quad \cdots\cdots\cdots (2)$$

Yang[5]을 비롯한 현재까지 제안된 컬러변환 알고리즘들은 단순히 컬러변환 연산 자체의 계산 복잡도를 낮추는데 초점을 맞추었으며 컬러변환 연산 횟수 등 계산량 측면은 고려하지 않았다.

DCT 기반 압축 영상 데이터에서 '0'인 양자화된 DCT 계수의 비율은 영상 블록의 내용이나 압축 비트율에 따라 달라진다. 일반적으로 '0'인 양자화된 DCT 계수의 비율은 복잡한 자연 이미지일수록 낮아지며 단순한 그래픽 이미지일수록 매우 높아진다. 이는 DCT 변환 원리에 의해 직관적으로 이해할 수 있다. He[7]는 DCT 기반 영상 코딩에서 압축 비트율과 '0'인 양자화된 DCT 계수의 비율 사이에는 거의 선형 관계, 즉, 압축 비트율이 증가하면 역으로 양자화된 DCT 계수 중에 '0'인 계수 비율이 선형적으로 감소함을 증명하였다. 본 논문에서는 간단하게 양자화 테이블에 대한 스케일링 인수로 8개의 2의 승수 $2^{0.5}$, 2^{1},..., 2^{4} 을 사용하여 비트율을 제어한 후에 '0'인 양자화된 DCT 계수의 변화를 측정하여 압축 비트율과 양자화된 DCT 계수사이의 선형 관계를 확인하였다. 샘플 영상인 "Lena"에 대해 스케일링 인수를 변화해가면서 측정한 '0'인 양자화된 DCT 계수의 변화는 그림 2와 같다.



〈그림 2〉 "Lena" 영상에서 양자화 테이블 스케일링 인수에 따른 '0'인 양자화된 DCT 계수 비율의 변화

본 논문은 일반적으로 JPEG 압축 영상에서 '0'인 양자화된 DCT 계수의 분포가 높음에 근거하여 컬러변환 연산의 일부를 IDCT 변환 이전에 수행하도록 연산 순서를 재배열하여 컬러변환에서 요구되는 연산횟수를 줄임으로써 계산 복잡도를 최적화하는 기법을 제시한다.

III. 연산 재배열에 의한 YCbCr-to-RGB 컬러변환 기법

시스템 이론 관점에서 볼 때에 그림 1에서 제시된 JPEG 디코더의 핵심 알고리즘, 즉 IDCT(Inverse Discrete Cosine Transform) 변환과 컬러변환 과정은 선형 시간종속 시스템(linear time-dependent system)이라고 할 수 있다[4]. 이들 변환사이의 선형 시스템 특성은 자유스럽게 두 개의 연산 과정을 순서를 바꾸어 실행 가능함을 의미하며, IDCT 변환을 YCbCr 컬러공간 또는 RGB 컬러공간 어디에서 실행하여도 결과가 같고, 그리하여 컬러변환 연산을 IDCT 변환 전에 수행할 수 있음을 나타낸다. 연산 순서를 재정렬하여 컬러변환 연산을 IDCT 변환 전에 수행하는 것은 많은 수의 '0'인 DCT 계수에 대해 컬러변환 연산을 제거함으로써 컬러변환

연산 전체 또는 일부에서 요구하는 계산 복잡도를 줄일 수 있다. 본 논문에서는 이러한 선형 시스템 특성을 바탕으로 연산 재배열을 통해 컬러변환 연산의 계산 복잡도를 줄이고자 한다.

IDCT 변환과 컬러변환의 연산 순서 재배열에서 가장 주의해야 할 요소는 채도 데이터의 업샘플(upsampling) 과정으로 실제 처리되는 데이터가 증가하는 과정이다. 예를 들어, JPEG 엔코딩 과정에서 4:2:0 모드의 서브샘플링을 사용한 경우에 업샘플링을 통해 채도 데이터가 4 배로 증가한다. 연산 재배열 통해 IDCT 변환 전에 업샘플링으로 채도 데이터가 증가할 경우 상대적으로 IDCT 변환 연산량이 증가하게 된다. 이에 본 논문에서 Bartkowiak[4]가 제안한 방법과 유사하게 식 (1)에서 주어진 컬러변환 행렬을 곱셈 및 덧셈 행렬로 분해하고 컬러변환 연산에서 계산 오버헤드가 큰 곱셈 연산 부분만을 IDCT 변환 전에 수행하도록 재배열하는 방법으로 접근하였다. 컬러변환 행렬의 곱셈 및 덧셈 행렬로의 분해는 식 (3)과 같다.

$$T = T_s T_m$$

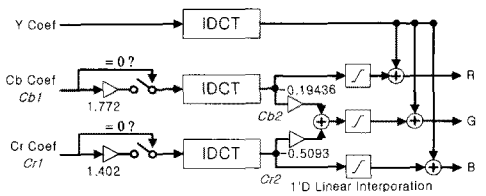
$$where\ T_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1.402 \\ 0 & 0 & 0.714 \\ 0 & 0.3444 & 0 \\ 0 & 1.772 & 0 \end{bmatrix}, T_s = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

또한, 채도 성분 *Cb*와 *Cr* 각각에 대해 두 개의 곱셈이 병렬로 요구되는데, 이를 하나의 순차 연결된 곱셈 연산으로 대치할 수 있다. 즉, 식 (3)에서 0.3444x*Cb*는 0.19436x(1.772x*Cb*)로, 0.714x*Cr*은 0.5093x(1.402x*Cr*)로 대치할 수 있다. 이러한 연산 재배열 과정은 컬러변환 연산에서 요구되는 대부분의 곱셈 연산을 IDCT 변환 전에 수행하도록 할 수 있다.

그리고 연산 자체의 계산 복잡도를 낮추기 위해 재배열된 부동소수점 곱셈 연산에 대해 Yang[5]이 제안한 정수 맵핑 기법을 적용하였다. 양자화 오류의 허용 가능한 범위는 8-비트의 *Y, Cb, Cr*를 8-비트의 *R, G, B*로 변환하는 부동소수점 연산의 rounding 오류인 0.5가 되며, 정수 맵핑의 양자화 오류가 이 범위 내에 있도록 Yang[5]의 기법을 적용하면 부동소수점 곱셈 연산을 식 (4)과 같이 정수 값에 대한 쉬프트와 덧셈 연산으로 변환 가능하다.

$$\begin{aligned} 1.772 \times Cb_1 &= Cb_1 + Cb_1 \gg 1 + Cb_1 \gg 2 + Cb_1 \gg 6 \\ 1.402 \times Cr_1 &= Cr_1 + Cr_1 \gg 2 + Cr_1 \gg 3 + Cr_1 \gg 5 \\ 0.5093 \times Cb_2 &= Cb_2 \gg 1 + Cb_2 \gg 4 + Cb_2 \gg 5 \\ 0.19436 \times Cr_2 &= Cr_2 \gg 3 + Cr_2 \gg 4 + Cr_2 \gg 7 \end{aligned} \quad \cdots\cdots (5)$$

상기와 같이 연산 재배열 및 정수 맵핑을 적용하여 제안한 핵심 JPEG 디코딩 과정은 그림 3과 같다.



〈그림 3〉 제안된 컬러변환 기법 기반의 핵심 JPEG 디코딩 과정

IV. 성능평가 및 분석

본 논문에서 제안한 컬러변환 기법은 JPEG 디코딩 과정에서 컬러변화 단계에 연산 재배열 및 정수 맵핑을 적용하여 전체 계산 복잡도를 줄인 알고리즘이며, PDA나 모바일폰 등과 같은 저성능 임베디드 시스템에서 동작하는 소프트웨어 JPEG 디코더에 적용하는 것을 목적으로 한다. 본 논문은 이러한 관점에서 성능 평가를 위해 임베디드 시스템 개발 플랫폼에서 제안된 알고리즘을 구현하고 몇몇 실험 영상을 이용하여 성능을 평가하였다. 구체적인 실험 환경은 표 2와 같다.

〈표 2〉 성능 평가 실험 환경

H/W	CPU	Intel Xscale PXA255 400MHz
	Memory	128 MB
	Flash Memory	64 MB Intel Strata NOR Flash Memory
S/W	O.S	Embedded Linux Kernel 2.4.20
	Compiling Tool	Cross-Toolchain(gcc 3.2.1/glibc 2.2.5) - supports software emulator for floating point operation
	JPEG Codec S/W	IJG JPEG Library version 6b[8] + arm patch

제안된 컬러변환 기법의 상대적인 성능을 평가하기 위해 다음의 두 가지의 컬러변환 기법과 성능을 비교한다.

(1) ITU-R Rec. BT.601-5 표준 컬러변환 기법[2]

(2) Yang[5]의 정수 맵핑 기반의 빠른 컬러변환 기법

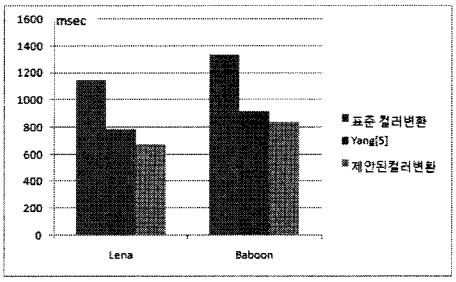
실험에 사용된 영상으로는 그림 4와 같이 2가지 영상을 사용하였으며, 이들 영상들은 일반적으로 JPEG 코덱 성능 평가에 주로 사용되는 영상으로서 "Lena" 영상은 공간 주파수 분포가 고른 특징이 있으며, "Baboon" 영상의 경우는 높은 공간 주파수 신호 분포가 높은 특징을 가진다.

(a) Lena (512x512x24bits) (b) Baboon (512x512x24bits)

〈그림 4〉 실험 영상

우선, 컬러변환 기법간의 계산 복잡도를 비교하기 위해 그림 5와 같이 실험 영상의 전체 복원 시간을 측정하여 비교하였다. 이는 컬러변환 기법을 JPEG 코덱 라이브러리에 적용하고 전체 영상 복원 시간을 비교함으로써 실제 적용 상황을 반영할 뿐만 아니라 제안된 기법의 경우 컬러변환 연산이 재배열되어 IDCT 변환과정과 복합적으로 고려하기 위함이다. 그림 5에서 제안된 기법이 표준 컬러변환 기법과 비교하여 평균적으로 37%~41%의 실행 시간을 단축하였는데, 이런 큰 차이는 임베디드 시스템에 탑재되는 대부분의 프로세서가

부동소수점 연산 명령어를 지원하지 않아 표준 컬러변환 기법에서 요구하는 부동소수점 연산을 소프트웨어 에뮬레이션으로 처리하고 이로 인해 큰 지연이 발생한 것이다.



〈그림 5〉 컬러변환기법에 따른 실험 영상의 복원 시간 비교

또한 Yang[5]의 기법에 비해 8%~18%의 실행 시간을 단축하고 있는데, 이는 연산 재배열을 통해 축소되는 곱셈 연산의 비중이 상대적으로 크며 그로 인해 계산 복잡도를 크게 줄일 수 있음을 의미한다. 그리고 "Lena" 영상이 "Baboon" 영상에 비해 컬러변환 기법간의 복원 시간 차이가 상대적으로 높은데 낮은 공간 주파수 신호 분포가 높기 때문에 '0'값을 가진 DCT 계수의 비중이 늘어나고 이와 비례하여 곱셈 연산 횟수도 줄어들기 때문이다.

제안된 기법의 계산 복잡도 감소가 복원 영상의 화질에 미치는 영향을 분석하기 위해 컬러변환 기법 별로 복원 영상들에 대해 객관적인 화질 측정치인 PSNR(Peak Signal to Noise Ratio)을 측정하여 비교하였다.

〈표 3〉 컬러변환 기법에 따른 실험 영상의 PSNR 비교

	Lena	Baboon
표준컬러변환	35.273	27.568
Yang[5]	32.856	24.837
제안된 컬러변환	31.794	22.585

표 3에서 알 수 있듯이 표준 컬러변환 기법과 비교하여 Yang[5] 기법과 제안된 기법에서 화질 저하가 비교적 크게 나타나고 있으며, 제안된 기법에서는 IDCT 변환 전의 정수 맵핑에 따른 양자화 오류가 IDCT 변환과정에서 증폭되어 상대적으로 화질 저하가 좀 더 크게 나타나고 있다. 그리고 "Baboo" 영상에서의 품질 저하가 더 크게 나타난 것으로 높은 공간 주파수 신호의 분포가 높기 때문이다.

제안된 컬러변환 기법은 JPEG 디코딩 과정에서 컬러변환에 대해 계산 복잡도를 크게 줄이나 그에 따른 화질 저하를 야기하므로 이에 대한 보완이 필요하다.

V. 결론

본 논문은 JPEG 디코딩 과정에서 실행시간 비중이 높은 반면, 최적화 시도가 적은 컬러변환 과정에 대해 계산 복잡도를 줄인 컬러변환 기법을 제안, 적용함으로써 JPEG 디코딩의 전체 실행시간을 단축하였다. 제안된 컬러변환 기법은 컬러변환 연산 자체의 계산 복잡도뿐만 아니라 컬러변화에서 요구되는 연산횟수를 줄임으로써 실행시간을 단축한다. JPEG 디코딩 과정에서 IDCT 변환과 컬러변환 단계 사이의 시간종속 선형성을 바탕으로 이들 간의 연산 순서를 재배열, 즉 컬러변환 연산을 곱셈 연산과 덧셈 연산으로 분해하고 곱셈 연산을 IDCT 변환 전으로 재배열하여 '0'인 DCT 계수에 대한 곱셈 연산을 제거함으로써 컬러변환에서 요구되는 연산 횟수를 줄인다. 또한, 재배열된 곱셈 연산에 대해 정수 맵핑 기법을 적용하여 연산 자체의 복잡도도 낮추었다. 임베디드 시스템 개발 플랫폼 환경에서 몇몇 실험 영상을 이용하여 제안된 알고리즘의 성능을 평가한 결과, 표준 컬러변환 기법과 Yang[5]의 정수 맵핑 기반의 빠른 컬러변환 기법에 비해 복원 시간을 크게 단축함을 알 수 있었으나 그에 따른 화질 저하도 야기되는 것을 확인하였다.

향후에는 제안된 컬러변환 기법에서 연산 재배열과 정수 맵핑에 의한 오류 성분을 줄이고 화질 향상을 지원하는 채도 재구성 기법 등을 같이 고려하여 복원 영상의 화질 왜곡을 최소화시키는 연구를 진행할 예정이다.

참고문헌

[1] ISO/IEC IS 10918-1, ITU-R Rec. T.81, Information Technology-Digital compression and coding of continuous-tone still image: Requirements and guidelines, 1993.

[2] ITU-R Recommendation BT.601-5, Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect-ratios.

[3] V. Bhaskaran and K. Konstantinides, Image and Video Compression Standards, Kluwer, 1998.

[4] M. Bartkowiak, "Optimizations of Color Transformation for Real Time Video Decoding," Digital Signal Processing for Multimedia Communications and Services, EURASIP ECMCS 2001, Budapest, 2001.

[5] Y. Yang, P. Yuhua and L. Zhaoguang, "A Fast Algorithm for YCbCr to RGB Conversion," IEEE Transactions on Consumer Electronics, Vol. 53, Issue 4, pp.1490-1493, 2007.

[6] H. M. Kim, W. S. Kim and D. S. Cho, "A new color transform for RGB coding," International Conf. on Image Processing(ICIP '04), Vol. 1, pp.107-110, 2004.

[7] Zhihai He and S. K. Mitra, "A linear source model and a unified rate control algorithm for DCT video coding," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, Issue 11, pp.970-982, 2002.

[8] Independent JPEG Group, JPEG Image Compression Library, http//www.ijg.org.