

임베디드 마이크로 프로세서 기반의 웨이블릿 영상 부호화기

論 文

51P-4-6

The wavelet image coder based on the embedded microprocessor

朴成煜* · 金永鳳** · 朴種旭***

(Sung-Wook Park · Young-Bong Kim · Jong-Wook Park)

Abstract - In this paper, we proposed a wavelet image coder based on the portable embedded microprocessor. The proposed coder stores the bit level information of the wavelet coefficient in the 2D significance array. Using this information, the coder make the significance check for coefficient and bit level scanning at the same pass. The proposed method has the advantage that we can reduce the scan iteratively and the memory usage for the coding process. Experimental results show that the proposed method outperforms popular image coders such as JPEG, EZW and SPIHT in based on the portable embedded system environment.

Key Words : Image Compression, Image Coder, Image Processing, Wavelet Transform

1. 서 론

최근 PC시장의 포화 상태, 개인용 휴대장비의 다양화, 기술의 표준화 및 고도화에 따라 개인 휴대 단말기(personal digital assistant : PDA), 지리정보 시스템(geographic information system : GIS), 의료정보 단말(Picture Archiving & Communication System : PACS), 주식시장 단말기와 같은 임베디드 시스템 분야가 매우 활성화되고있다. 임베디드 시스템이란 마이크로 프로세서와 같이 저장된 프로그램이 있는 단순한 형태의 시스템으로 보통 크기나 성능에 관계없이 마이크로 프로세서가 삽입된 시스템을 총칭한다. 이중 GPS, PDA나 PACS 단말기 등과 같은 휴대형 임베디드 마이크로 프로세서가 내장된 시스템인 경우 전체 시스템 가격이나 소비전력을 낮추기 위해 시스템에 많은 제한을 가하는 특성이 있으며, 범용 운영체제를 사용하기보다는 특화된 실시간 운영체제를 사용하거나 혹은 운영체제 없이 모니터 프로그램에 의해 입력되어 필요한 기능만을 수행하는 단일 프로그램으로 소프트웨어가 구성된다[1]. 그리고 이러한 제한된 기능의 휴대형 임베디드 시스템에서도 유·무선 통신을 이용한 영상 데이터의 처리 및 전송은 필수적인 기능으로 요구되고 있으나, 제한된 시스템 특성으로 저장 용량 및 전송 채널 용량의 한계가 있기 때문에 효율적인 영상 부호화 방법의 필요성이 요구된다.

영상 부호화 기법 중에서 웨이블릿 변환을 이용한 영상 부호화 방법은 JPEG과 같은 블록 기반 부호화기에서 발생하는 블로킹 현상(blocking effect)을 없애주고 고압축이 용

이하다는 특징으로 인해서 웨이블릿 기반 부호화 기법에 대한 많은 연구가 이루어지고 있다[2~5]. Shapiro는 웨이블릿 분해된 영상이 가지는 같은 방향의 부대역간의 자기 유사성(self-similarity)을 이용한 연속적인 근사 양자화(successive approximation quantization:SAQ)를 사용한 임베디드 제로 트리 웨이블릿(embedded zerotree wavelet : EZW) 알고리즘을 제안하였다[6]. EZW 부호화 방법은 원하는 비트율로 영상을 정확히 전송할 수 있는 비트율 제어(bit-rate control)가 가능하기 때문에 통신 채널의 대역폭에 따라서 비트율을 조절할 수 있을 뿐만 아니라, SAQ를 사용하여 크기가 큰 웨이블릿 계수들을 먼저 전송하기 때문에 주어진 비트율에서 보다 우수한 화질을 제공하고, 알고리즘이 빠르고 단순하다는 장점이 있다. EZW 부호화 방법은 SAQ 과정에서 각 웨이블릿 계수들을 계수의 중요도와 제로트리 구조에 따라서 POS, NEG, ZTR, IZ 등의 네 가지 기호중 하나에 할당하고, 이 기호들을 적응 산술 부호화를 사용하여 무손실 부호화 한다. 이러한 SAQ 과정은 임계치를 감소시키면서 원하는 비트율이 만족될 때까지 수행된다. 그러나 이러한 과정에서 상대적으로 중요도가 덜한 정보를 가지고 있는 고주파 부대역 내의 웨이블릿 계수들이 반복적으로 스케닝되어, 부호화 과정에서 중복성이 발생할 수 있는 단점이 있다. Said와 Pearlman은 Shapiro의 방법을 개선하여 계층적 트리 구조내에서의 집합 분할(Set Partitioning in Hierarchical Trees : SPIHT) 알고리즘을 제안하였다[7]. SPIHT는 EZW와 마찬가지로 자기 유사성 탐색이라는 개념을 근거로 한다. 그러나 Shapiro의 연구에서는 각 부밴드의 계수들 중에서 크기가 가장 큰 계수의 절대값들의 집합을 이용하여 부밴드들이 중요한 대역인지 중요하지 않은 대역인지를 판단한 후, 중요한 부밴드만 부호화 하기 때문에 부호화의 효율성이 높다. 그러나 SPIHT를 이용한 다중 부호화의 경우, 정확한 복호를 위해서는 부호기에서 만들어진 패

* 正 會 員 : 仁川大學校 工大 電子工學科 博士課程

** 正 會 員 : 仁荷工業專門大學 電氣科 教授

*** 正 會 員 : 仁川大學校 工大 電子工學科 教授 · 工博

接受日字 : 2002年 9月 18日

最終完了 : 2002年 10月 18日

스를 복호기에서도 따라야 한다고 가정하기 때문에 언제나 이용하는 가장 낮은 해상도 비트열은 패스와 관계되는 모든 정보를 가져야만 한다. 따라서 저해상도 비트열이 전체의 대부분을 차지하였다. 또한 복호의 경우 하나의 임베디드 비트열(embedded bitstream)에서 다중 해상도 정보를 얻기 때문에 해상도가 다르더라도 복호 시간은 일정하다는 단점이 있다.

본 논문에서는 제한된 구조를 가지는 휴대형 임베디드 마이크로 프로세서 기반의 시스템에서 효율적으로 수행될 수 있는 웨이블릿 영상 부호화기를 제안한다. 제안한 방법은 빠르고 간단한 EZW 알고리즘과 중요 부밴드만 선택적으로 부호화하는 SPIHT의 개념을 사용함으로써, 웨이블릿 계수들의 반복적인 스캐닝에 의한 부호화 과정의 중복성을 효율적으로 개선하여 제로트리 부호화를 효과적으로 수행할 수 있는 부호화 방법이다. 이를 위해 웨이블릿 계수들의 비트 레벨 정보를 저장하는 2D 중요계수 배열을 사용하여, 계수의 중요도 검사와 계수의 스캐닝을 동시에 수행할 수 있어서 영상 부호화를 위해 사용되는 메모리 사용량과 처리 속도를 개선할 수 있다. 또한 제안한 방법은 기존의 EZW와 마찬가지로 임베디드 비트 스트림을 생성하기 때문에 정확한 비트율 제어가 가능하며, 중요한 정보를 가지는 웨이블릿 계수들을 먼저 전송하기 때문에 점진적 영상 전송이 가능하다는 장점이 있다.

본 논문은 다음과 같이 구성된다. II장에서는 웨이블릿 기반의 부호화에 대하여 간략히 기술하고, III장에서는 본 논문에서 제안된 부호화 방법에 대하여 기술한다. 그리고 IV장에서는 제안한 방법에 대한 실험 결과를 제시하고, 마지막 V장에서 결론을 맺는다.

2. Wavelet 기반의 영상 부호화 기법

2.1 EZW 부호화 방법

EZW 부호화 방법은 웨이블릿 변환 계수들의 대역간 자기 유사성을 이용하여 영상을 압축한다. 그림 1에서 웨이블릿 피라미드에서 주파수가 낮은 대역에 있는 웨이블릿 계수를 parent라 하고, 이 대역보다 높은 주파수 대역내의 계수들을 descendant라 한다. 이러한 웨이블릿 트리 구조는 parent에 해당하는 계수가 작은 값을 가지면, 이 계수의 descendant들이 대부분 작은 값을 갖는 특징이 있다. EZW 부호화 방법에서는 이러한 트리 구조와 SAQ를 이용하여 영상을 부호화 한다. SAQ는 $T_i = T_{i-1}/2$ 를 만족하는 일련의 임계치 T_1, \dots, T_N 들을 각 웨이블릿 계수들에 반복적으로 적용하면서 웨이블릿 계수들의 중요도를 판단한다. 여기서 초기 임계치 T_1 은 모든 웨이블릿 계수 $c_{i,j}$ 에 대해 $|c_{i,j}| < 2T_1$ 의 조건을 만족하는 값이다. 현재 임계치 T_i 보다 크기가 큰 계수를 중요 계수(significant)라 하고, 작은 계수를 무의미한 계수(insignificant)라 한다. SAQ 과정에서는 웨이블릿 계수들을 저주파 대역에서부터 고주파 대역까지 계층적 순서에 따라 피라미드를 스캔하면서, 각 계수의 크기가 임계치 T_i 보다 큰지, 작은지를 결정한다. 만약 계수가 significant라면 계수의 부호에 따라 POS(positive

significant), NEG(negative significant) 기호로 나타내고, 반대로 계수가 insignificant라면 이 계수에 대한 descendant 중에 significant 계수가 있는지를 조사하여 significant 계수가 하나도 없으면 ZTR(zerotree root) 기호를 할당하고, significant 계수가 하나라도 있으면 IZ(isolated zero) 기호를 할당한다. 이와 같이 생성된 기호들은 적응적 산술 부호화를 이용하여 무손실 부호화된다. 이러한 과정을 주어진 비트율을 만족할 때까지 임계치를 낮추어 가면서 반복적으로 수행한다.

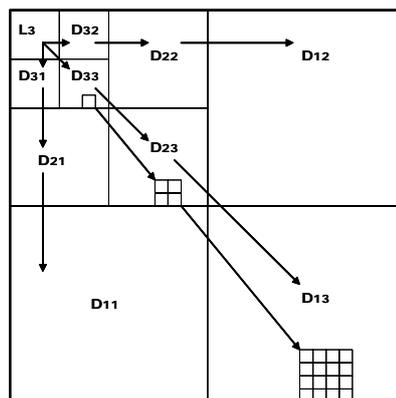


그림 1 웨이블릿 평면상의 자기 유사성

Fig. 1 Parent-child dependencies of subbands

2.2 SPIHT 부호화 방법

SPIHT는 EZW와 마찬가지로 자기 유사성 탐색이라는 개념을 근거로 한다. SPIHT 알고리즘에는 두 개의 주된 단계가 있다. 정렬 패스(sorting pass)와 세분 패스(refinement pass)이다. 정렬 패스에서는 계수들을 특정 임계값과 비교해서 중요도(significant or insignificant)를 알아내고 세분 패스에서는 정렬 패스에서 구한 중요한 계수 값을 세밀화 한다. SPIHT에서 사용되는 몇 가지 정의는 다음과 같다.

- T : 특정 임계값
- $O(i, j)$: 노드 (i, j) 의 모든 offspring의 좌표들의 집합
- $D(i, j)$: node (i, j) 의 모든 후손들의 좌표들의 집합
- H : 가장 상위 레벨 내에 있는 노드들의 좌표들의 집합.
- $L(i, j) = D(i, j) - O(i, j)$

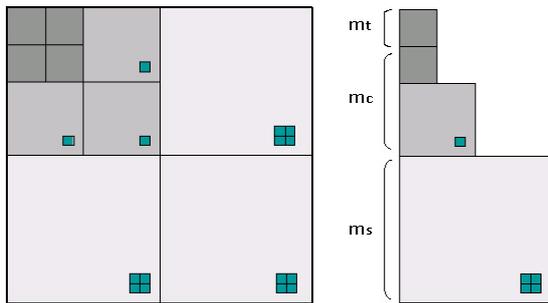
SPIHT 알고리즘은 LSP(list of significant pixel), LIP(list of insignificant pixel), LIS(list of insignificant set)라는 세 개의 리스트를 사용해서 계수들의 좌표를 관리한다. LIP는 중요하지 않은 화소들의 리스트를 의미하며, LSP는 중요한 화소들의 리스트 그리고 LIS는 중요하지 않은 집합들의 리스트를 의미한다. 처음에는 LSP를 비워두고 H에 속한 계수들을 LIP에 넣고, LIP의 계수들 중에서 descendants를 LIS에 넣는다. 정렬 패스에서 LIP를 차례대로 조사해 가면서 T보다 큰 값은 LSP에 넣고 T보다 작은 값은 LIP에 추가시킨다. Descendants에 관한 정보는 $D(i, j)$ 에 대해서 A형으로, $L(i, j)$ 에 대해서 B형으로 LIS에 넣는다. 세분 패스에서 LSP를 차례대로 세분하는 정보를 만든 후, 정렬 패스와 세분 패스를 원하는 비트율에 도달할 때까지 반복한다.

3. 제안된 웨이블릿 영상 부호화기

휴대형 임베디드 마이크로 프로세서 기반의 시스템은 앞서 언급한 바와 같이 간편한 휴대성을 위해 전체 시스템의 기능이나 소비전력을 낮추기 위해 시스템에 많은 제한을 가하는 특성이 있다. 이러한 시스템을 위해 본 논문에서는 부호화시 시스템의 메모리 사용량을 최소화하고, 부호화 속도를 향상시킬 수 있는 웨이블릿 영상 부호화 방법에 대하여 설명한다.

3.1 계수의 비트-레벨 정보를 위한 2D 중요계수 배열

웨이블릿 변환된 모든 계수들의 중요도 검사를 위해 많은 연산 시간이 소비된다. 본 연구에서는 계수의 중요도 검사를 빠르고 효과적으로 수행하기 위해 계수들의 비트-레벨 정보를 저장하기 위한 2D 중요계수 배열을 제안한다. 제안된 이차원 중요계수 배열은 그림 2(b)와 같이 웨이블릿 계수 피라미드에서 수평방향의 상세 정보를 저장하는 D_{nl} 대역 피라미드와 같은 크기와 모양을 가지는 이차원 데이터 구조이다. 자료 구조 내의 각 원소(element)는 그림 2(a)에서와 같이 동일한 레벨 또는 그 이하의 레벨에서의 동일한 위치에 있는 각 방향 부대역의 가장 큰 계수를 나타내는데 필요한 비트들의 정보를 가진다.



(a) 웨이블릿 계수 피라미드 (b) 중요 계수 배열
그림 2 계수들의 비트-레벨 정보를 위한 2D 중요 계수 배열

Fig. 2 2D significant coefficients array for bit-level information of wavelet coefficients.

이차원 중요계수 배열의 생성을 위해 웨이블릿 계수 피라미드의 높이를 H 라 하고, 너비를 W 라 하자. 그리고 저해상도 영상 정보를 가지는 스케일 레벨(L)을 제외한 피라미드 레벨의 수를 S 라 하면, 스케일 레벨(L)의 높이(H_L)와 너비(W_L)의 크기는 $H_L = H/2^S$ 와 $W_L = W/2^S$ 이다. 따라서 이차원 자료구조 내에서 각 레벨의 크기는 다음 식 (1)과 같이 정의될 수 있다.

$$\begin{aligned} W_{x,y} &= W_L 2^{\lfloor \log_2(\max\{1, \frac{x}{H_L}\}) \rfloor} \\ H_{x,y} &= H_L 2^{\lfloor \log_2(\max\{1, \frac{y}{H_L}\}) \rfloor} \end{aligned} \quad (1)$$

또한, 이차원 중요계수 배열의 원소 $m_{x,y}$ 는 그림 2(b)와

같이 $m_i(x,y)$, $m_c(x,y)$ 그리고 $m_s(x,y)$ 로 정의될 수 있다. 여기서 $m_i(x,y)$ 는 웨이블릿 계수들 중 가장 상위의 레벨(L)에 있는 중요 계수 정보를 가진다. 그리고 $m_s(x,y)$ 는 웨이블릿 계수들 중 최하위 레벨(Level 1)에 있는 계수 정보를 가지며, $m_c(x,y)$ 는 그 이외의 레벨에 존재하는 계수 정보를 포함한다. 따라서 중요계수 배열의 원소 $m_{x,y}$ 에 대하여 정리하면 아래 식과 같다.

$$\begin{aligned} m_i(x,y) &= \max\{m_{x,y+H_s}, \\ &\quad \lfloor \log_2(|c_{x,y}|) \rfloor + 1\} \\ m_c(x,y) &= \max\{m_{2x,2y}, m_{2x+1,2y}, \\ &\quad m_{2x,2y+1}, m_{2x+1,2y+1}\} \\ m_s(x,y) &= \lfloor \log_2(\max\{|c_{x,y}|, |c_{x+W_x,y}|, \\ &\quad |c_{x+W_x,y-H_x,y}|\}) \rfloor + 1 \\ m_{x,y} &= \begin{cases} m_i(x,y), & \text{if } x < W_L \wedge y < H_L \\ m_i(x,y), & \text{if } x < W_L \wedge y < H_L \\ m_s(x,y), & \text{if } x \geq W/2 \wedge y \geq H/2 \\ \max\{m_s(x,y), m_c(x,y)\}, & \text{otherwise} \end{cases} \end{aligned} \quad (3)$$

여기서 x, y 는 각각 $0 \leq y < H, 0 \leq x < W_L 2^{\lfloor \log_2(\max\{1, \frac{x}{H_L}\}) \rfloor}$ 의 범위를 가지며, $\lfloor x \rfloor$ 기호는 x 를 넘지 않는 최대의 정수를 의미한다. 표 1에 식 (2), (3)에 의한 2D 중요계수 배열 생성을 위한 알고리즘을 나타내었다.

표 1 2D 중요 계수 배열의 생성 알고리즘
Table 1 2D significant coefficient array generate algorithm

```
function Make_2D_Array( )
{
  for (l=1; l<S+1; l++) {
    H_l = H/2^min(l,S);
    W_l = W/2^min(l,S);
    for (j=0; j<H_l-1; j++) {
      for (i=0; i<W_l-1; i++) {
        if (l==1) {
          t = 0;
          u = floor(1 + log2(max(|c_{i,j+H_l}|, |c_{i+W_l,j}|, |c_{i+W_l,j+H_l}|)))
        }
        else if (l <= S) {
          t = max{h_{2i,2j}, h_{2i+1,2j}, h_{2i,2j+1}, h_{2i+1,2j+1}};
          u = floor(1 + log2(max(|c_{i,j+H_l}|, |c_{i+W_l,j}|, |c_{i+W_l,j+H_l}|)))
        }
        else {
          t = max{h_{i,j+H_L}, h_{i+W_L,j}, h_{i+W_L,j+H_L}};
          u = floor(1 + log2(max(|c_{i,j}|)))
        }
        h_{i,j} = max(t, u);
      }
    }
  }
}
```

3.2 부호화 알고리즘

압축 알고리즘의 전체 구조는 표 2와 같다. 먼저 공간 영역의 영상은 웨이블릿 분해에 의하여 웨이블릿 공간으로 변환된다. 그리고 계수들의 비트-레벨 정보 저장을 위한 이차원 중요계수 배열을 생성하고, 적응 이진 산술 부호화기(QM-부호화기)가 초기화된다. 웨이블릿 계수들은 비트 스캐닝 알고리즘에 의해 최상위 비트부터 스캔된다. 복호화 과정은 부호화 과정의 역으로 수행된다.

표 2 부호화 알고리즘
Table 2 Coding algorithm

Coding Algorithm

- **Load Image**
- **Transform image to wavelet domain and convert the coefficients to non-negative integers and signs**
- **Build 2D Data Structure for Bit-level information**
- **Initialize QM-Coder**
- **For** ($n = \lfloor \log_2(\max_coefficient) + 1 \rfloor$; $n > 1$; $n--$)
 For ($i = 0$; $i < H_L - 1$; $i++$)
 For ($j = 0$; $j < W_L - 1$; $j++$)
 Scan($j, i, 0, n$);

표 3 스캐닝 알고리즘
Table 3 Scanning algorithm

```
function Scan(i, j, l, n)
{
  if (Continue_Coding()) {
    if ( $h_{i,j} < n$ ) QM_coder(Insigificant, context);
    else {
      if ( $h_{i,j} == n$ ) QM_coder(Sigificant, context);
      if (l == 0) {
        Coeff_Scan(i, j, TopLevel, n);
        Scan(i, j + H_L, l, n);
      } else {
        Coeff_Scan(i, j, Horizontal_Direction, n);
        Coeff_Scan(i + W_i, j, Diagonal_Direction, n);
        Coeff_Scan(i + W_i, j - H_i, Vertical_Direction, n);
        if ((2*j) < H) {
          Scan(2*i, 2*j, l+1, n);
          Scan(2*i+1, 2*j, l+1, n);
          Scan(2*i, 2*j+1, l+1, n);
          Scan(2*i+1, 2*j+1, l+1, n);
        }
      }
    }
  }
}
```

표 3은 부호화 알고리즘의 주요 부분인 계수 스캐닝 알고리즘이다. 스캐닝 알고리즘이 가장 먼저 하는 일은 부호화 완료 조건의 검사이다. 여기서 완료 조건은 부호화 명령이 주어지는 파라미터 값으로 원하는 화질을 나타내는 비트율이다. 그리고 2D 중요계수 배열을 검색하여 현재의 위치에서 제로트리가 시작하는지를 조사한 후 그 결과를 부호화한다. 만일 계수가 제로트리로 시작한다면, 현재 피라미드 레벨 이하의 계수에 대한 비트 부호화를 생략할 수 있다. 또한 피라미드의 최상위 레벨인 스케일링 계수(L=0)를 부호화할 경우, 스캐닝 알고리즘은 재귀적(recursive)으로 구조로 되어 있어 현재 피라미드 레벨 이하의 다음 레벨에 위치하는 동일 위치의 계수를 순환적으로 검색한다. 그리고 최상위 레벨 이하의 계수들을 부호화 할 경우에는 현재 위치대의 수평, 수직, 대각 방향 부밴드의 세 계수들을 모두 처리하게 되며, 만일 그 다음 하위 레벨이 존재한다면 그 레벨의 네 개의 descendant 계수들은 모두 재귀적으로 스캐닝된다.

웨이블릿 계수는 표 4의 Coeff_Scan 알고리즘에 의해서 처리된다. Coeff_Scan 알고리즘은 처음 계수가 앞서 중요 계수로 판별되었는지의 여부를 조사한다. 만일 계수가 중요 계수로 판별되었다면, 계수의 비트 레벨중 현재의 비트를 부호화하고 주어진 평균 제곱 오차(MSE) 비트율(bpp)을 만족하는지를 계산한다. 그렇지 않고 만약 계수가 2^n 보다 작다면, 현재 비트 레벨의 비트를 부호화하고 계수의 최상위 비트(MSB)가 0인지 1인지를 조사한다. 만약 MSB가 1이라면 계수의 부호를 부호화하고 주어진 비트율(bpp)을 만족하는지를 계산한다. 또한 Bit_Return(x, n) 함수는 x의 절댓값에서 n 번째 비트값을 반환하며 $s_{i,j}$ 는 (i, j)에 위치하는 웨이블릿 계수의 부호를 의미한다. Bit_Return(x, n) 함수에 의해 생성된 비트들은 이진 산술 부호화 기능을 담당하는 QM_coder(b, context) 함수에 의해 산술 부호화 된다. 여기서 인수 b는 부호화 되는 비트를 의미하며 context는 현재 부호화 되는 심볼에 대한 확률을 추정하기 위한 패턴을 의미한다. 부호화 알고리즘의 완료는 함수 Continue_Coding()에 의해 결정된다.

표 4 계수 부호화를 위한 알고리즘

Table 4 The algorithm for coefficients coding

```
function Coeff_Scan(x, y, context, n)
{
  if ( $c_{x,y} < 2^n$ ) {
    QM_coder(Bit_Return( $c_{x,y}$ , n), context);
    if (Bit_Return( $c_{x,y}$ , n) == 1) {
      QM_coder( $s_{x,y}$ , Sign);
      Check_Bpp(n,  $c_{x,y}$ );
    } else {
      QM_coder(Bit_Return( $c_{x,y}$ , n), Coeff_Bit);
      Check_Bpp(n,  $c_{x,y}$ );
    }
  }
}
```

이 함수는 완료 조건을 계산하기 위해 부호화가 진행되는 동안 주어진 비트율을 만족했는지를 계산하여 참 또는 거짓을 반환한다. 만일 Continue_Coding() 함수의 반환값이 참이라면 부호화는 계속되고 거짓이면 부호화는 완료된다. 부호화 알고리즘의 완료 조건을 계산하기 위해 부호화가 진행되는 동안 현재 부호화 단계에서 근사화된(approximated) 영상의 화질이 계산된다. 이것은 계수 x 의 n 번째 비트가 부호화 될 때마다 함수 Check_Bpp()를 호출하는 것으로 수행된다. 부호화가 진행되는 동안 수행되는 완료 조건의 계산은 MSE나 bpp를 사용한다.

3.3 부호화 완료 조건

제안된 부호화기가 부호화를 완료해야 하는 시점은 다음과 같은 조건이 발생한 경우이다. 첫 번째로 웨이블릿 계수의 모든 비트가 부호화된 경우이고, 두 번째의 경우 QM-부호화기에 의해 생성되는 비트율이 부호화시 파라미터로 주어진 비트율에 도달했을 때이다. 그리고 마지막으로 출력 영상의 화질이 주어진 임계치(threshold) 이하로 나타날 경우에 부호기는 부호화를 완료해야 한다. 먼저, 첫째 조건의 경우 계수의 모든 비트들이 부호화되면 부호화 알고리즘이 종료되기 때문에 크게 문제되지 않는다. 둘째 조건 역시 QM-부호화기의 출력 루틴에서 나오는 비트나 바이트의 수를 계산하는 것으로 쉽게 수행될 수 있다. 그러나 셋째의 경우 부호화 알고리즘은 복원된 영상의 MSE를 반드시 알고 있어야 하는 문제점이 있다. 복원된 영상의 MSE는 모든 웨이블릿 계수들에 대한 웨이블릿 역변환을 수행해야 하고, 그 결과로부터 MSE를 계산하여야 한다. 반면, 부호화 알고리즘은 매우 빈번하게 완료 조건을 검사해야 하기 때문에, 이 방법을 그대로 사용할 경우 부호화 알고리즘의 성능을 저하시키는 요인이 된다. 따라서 본 논문에서는 Daubechies 웨이블릿 변환의 정규직교성(orthonormality)을 이용한 수정된 MSE방법을 사용한다.

웨이블릿 변환과 같은 정규직교 변환은 식(4)과 같이 변환되기 이전의 영상 픽셀값과 변환후의 값이 동일하다.

$$\sum_{i,j} (x_{i,j})^2 = \sum_{i,j} (c_{i,j})^2 \quad (4)$$

여기서 $x_{i,j}$ 는 공간 영역의 픽셀의 명암값을 의미하고, $c_{i,j}$ 는 웨이블릿 계수를 의미한다. 또한 원 영상과 어떤 단계의 근사화된 영상간의 MSE는 공간 영역과 웨이블릿 영역에서 동등하게 계산될 수 있으므로, MSE를 계산하기 위해서 반드시 웨이블릿 역변환을 수행할 필요는 없다는 것을 알 수 있다. 따라서 본 알고리즘에서는 원 영상과 근사화된 영상간의 MSE를 계산하는 것 대신, 수정된 MSE(mMSE) 기법을 사용하여 계산 효율을 증대시켰다. 즉, 부호화 초기에 사용되는 mMSE 값은 식(5)과 같이 웨이블릿 계수값의 에너지와 같다.

$$mMSE = \sum_{i,j} (c_{i,j})^2 \quad (5)$$

예를 들어 계수 c 의 각 비트가 보내지면, 계수 c 의 이전 approximation에 의해 산출되었던 에러를 빼서 $mMSE$ 를 갱신하고 계수의 새로운 approximation 에러를 추가한다. 계수

c 의 approximation 에러는 마지막 비트의 레벨과 계수 c 자체에 의해서만 결정되는데, 만일 계수 c 의 n 번째 비트 부호화에 대한 $mMSE$ 는 식(6)와 같은 방법에 의해 갱신된다.

$$mMSE = mMSE_{prev} - \begin{cases} [(|c| \& (2^n - 1)) - 2^{n-1}]^2 - [(|c| \& (2^{n-1} - 1)) - 2^{n-2}]^2 \\ c^2 - (2^{n-1} + 2^{n-2} - c)^2 \\ 0 \end{cases} \quad (6)$$

여기서 $\&$ 는 비트 논리곱을 의미한다. 식(6)는 $n > 1$ 일 경우에만 적용되며, 식에서 첫 번째 if $\lfloor \log_2 c \rfloor > n - 1$ 인 경우는 계수의 부호를 이미 알고있을 때, 부호를 제외한 계수의 비트값이 1인 비트를 찾아내어 mMSE를 계산하는 방법에 대한 정의이다. 두 번째의 if $\lfloor \log_2 c \rfloor = n - 1$ 일 경우는 계수의 n 번째 비트값이 부호 비트일 때의 mMSE 대한 정의이며, 마지막의 if $\lfloor \log_2 c \rfloor < n - 1$ 인 경우는 계수의 비트값이 처음 1 비트를 만나기 전의 0 비트에 대한 정의로서, 이때 mMSE 값은 변경되지 않음을 의미한다.

3.4 적응 산술 부호화기

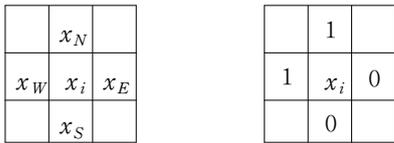
계수 스케닝 알고리즘에 의해 생성된 비트들은 산술 부호화기 보다 우수한 성능을 보이는 QM-부호화기를 이용하여 부호화한다. 적응 산술 부호화 기법의 일종인 QM-부호화기는 JBIG(Joint Bi-level Image Experts Group : JBIG)에서 규정한 이진 영상 압축을 위한 엔트로피 부호기이다[8]

QM-부호화기의 핵심은 고정된 부호구간 $[0.0, 1.0)$ 내에서의 두 개의 확률구간 부할을 기반으로 한다는 것이며, 이것은 부호화가 종료될 때까지 재귀적으로 수행한다. 개념적으로 특정 컨텍스트(context)를 기반으로 입력되는 이진 심볼들은 구간 $[0.0, 1.0)$ 상의 실수 r 로 사상된다. 컨텍스트는 인접한 화소들의 위치를 표시하는 특정한 패턴을 대변하는 정수값이며, 현재의 화소를 부호화하기 위해 사용되는 QM-부호화기의 확률구간 상태를 색인 형태로 제공한다. 그리고 QM-부호화기에서는 현재 심볼에 대한 확률을 추정하는데 있어서 인접한 픽셀들의 패턴을 사용하는데, 이를 컨텍스트 모델링이라 한다. 컨텍스트 모델링을 어떻게 하느냐에 따라서 QM-부호화기의 성능을 향상시킬 수 있다[9][10]. x_1, x_2, \dots, x_n 들로 이루어진 심볼들을 부호화하는데 필요한 최소 코드길이 l 은 다음과 같은 식으로 주어진다.

$$l = -\log_2 \prod_{i=1}^n p(x_i | x_{i-1}, x_{i-2}, \dots, x_1) \quad (7)$$

QM-부호화기에서는 심볼의 분포를 추정하기 위해서 x_1, x_2, \dots, x_n 에 대한 서브시퀀스 S 를 사용하는데, 본 논문에서 사용한 컨텍스트 모델은 그림 3(a)와 같다. 그리고 계수의 부호화를 위해 서로 다른 확률 분포따라 부호, 완료조건, 계수의 첫 번째 한 비트 다음의 비트값, 최상위 레벨의 스케일링 계수 비트값, 웨이블릿 피라미드에서 각 레벨의 제로트리 정보 그리고 각 방향의 피라미드에서 서로 다른 레벨에 위치하는 insignificant 계수의 중요도 검사와 같은 데

이터들의 컨텍스트가 정의된다. 그림 3(b)는 본 논문에서 사용한 컨텍스트 모델의 예를 나타낸다. 즉, 현재 부호화 하고자 하는 x_i 비트에 대한 확률 분포는 $S=\{x_N, x_W, x_E, x_S\} = \{1,1,0,0\}$ 를 사용하여 추정한다. 여기서 x_N 와 x_W 는 현재 임계치 L_i 일 때의 SAQ에 의해 생성된 값이고, x_E 와 x_S 는 바로 이전 임계치인 L_{i-1} 일 때의 SAQ에 의해 생성된 값이다.



(a) 컨텍스트 모델 (b) 컨텍스트의 예

그림 3 컨텍스트 모델링
Fig. 3 Context Modeling

4. 실험 및 결과

본 논문에서 제안한 부호화기의 성능을 평가하기 위해서 그림 4와 같은 구조의 임베디드 시스템을 사용하였다. 사용된 시스템은 Intel의 저전력 32비트 RISC 마이크로프로세서인 StrongARM을 사용하며, 16MB의 Flash ROM과 32MB의 SDRAM으로 구성된다. StrongARM 마이크로 프로세서는 206Mhz의 클럭에 2.1MIPS의 처리 능력을 가지는 임베디드형 마이크로 프로세서이다. 시스템의 운영을 위해 blob-1.0.8-pre2 Boot Loader와 Linux 2.4.1 Kernel을 사용된 시스템에 맞게 수정하여 포팅(porting)하였다. 또한 프로그램은 C언어를 사용하여 작성되었고, 프로그램의 컴파일은 ARM 프로세서용 컴파일러인 ARM Linux gcc 2.95.2 버전을 사용하였다. 모든 작업은 Linux Redhat 6.0기반의 호스트 PC에서 직렬포트 및 이더넷 포트를 통해 임베디드 시스템으로 시리얼 통신 및 텔넷(telnet) 연결에 의해 수행되었다.

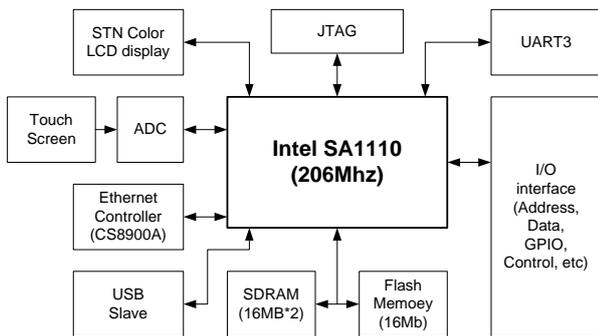


그림 4 임베디드 시스템 블럭도
Fig. 4 Embedded system block diagram

본 논문의 실험을 위해 512x512 크기를 가지는 Lena와 Barbara 영상을 사용하였고, 웨이블릿 변환을 위해 Daubechies 9/7 웨이블릿 필터를 사용하였다. 그리고 알고리즘의 성능 평가를 위해 EZW, SPIHT 그리고 Baseline

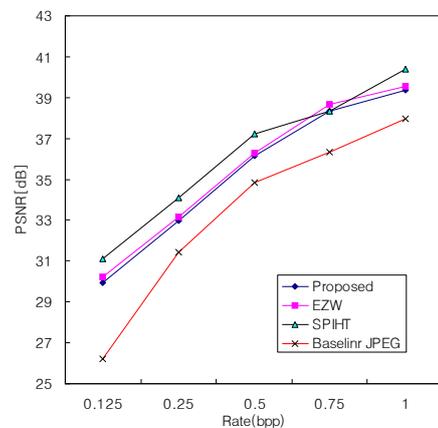
JPEG과의 PSNR(Peak Signal to Noise Ratio : PSNR)을 이용하여 화질을 평가하였다. 본 실험의 초점은 본 논문에서 제안한 2D 중요계수 배열을 사용함에 따른 메모리 사용량과 이것에 의해 발생할 수 있는 화질저하 문제에 대해 맞추었다.

표 5과 그림 5는 비트율에 따른 PSNR 성능을 나타낸다. Lena 영상과 Barbara 영상 모두에서 SPIHT보다는 약간 떨어지지만, JPEG보다는 월등히 높으며 EZW와 비슷한 성능의 결과를 나타내었다. 또한 0.5bpp 이하의 낮은 비트율에서도 다른 방법들과 유사하게 좋은 성능을 보였다. 그림 6은 제안한 방법에 의한 복원영상과 원영상을 나타내었다. 그림에서 볼 수 있는 바와 같이 낮은 bpp에서 복원된 영상도 블록간 왜곡이 없는 좋은 화질을 보임을 알 수 있다. 또한 부호화시 필요한 메모리 사용량의 경우, 측정 방법의 어려움으로 정확한 정량적인 측정은 할 수가 없었다. 그러나 기존 EZW나 SPIHT를 사용할 경우 이론적으로 약 8(계수의 자료형이 8비트 정수) * N 비트의 메모리가 부호화를 위해 필요하지만, 제안한 부호화기의 메모리 사용량은 약 8 * N/3 비트의 메모리만 요구된다. 여기서 N은 계수의 개수이다. 따라서 제안된 부호화기는 해상도가 높은 영상을 처리할 경우에 더욱 효과적으로 수행될 수 있음을 의미한다.

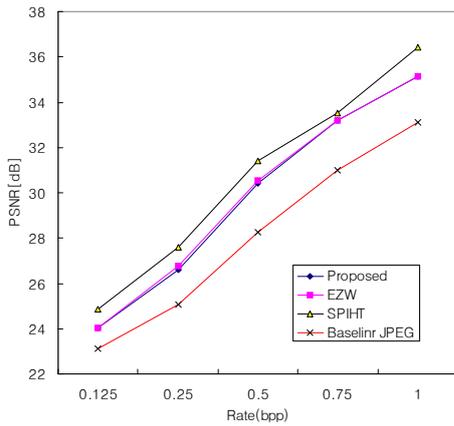
표 5 비트율에 따른 PSNR 결과

Table 5 Comparison of PSNR result at various bit rates

압축방법 \ bpp		bpp				
		0.125	0.25	0.5	0.75	1.0
Lena	Proposed	29.93	32.97	36.15	38.36	39.34
	EZW	30.23	33.17	36.28	38.64	39.55
	SPIHT	31.09	34.11	37.21	38.32	40.41
	JPEG	26.21	31.42	34.84	36.6	37.95
Barbara	Proposed	24.04	26.61	30.41	33.19	35.15
	EZW	24.03	26.77	30.53	33.20	35.14
	SPIHT	24.86	27.58	31.39	33.51	36.41
	JPEG	23.11	25.05	28.27	31.01	33.11



(a) Lena 영상



(a) Barbara 영상

그림 5 비트율에 따른 PSNR 그래프

Fig. 5 Comparison of PSNR graph at various bit rates



(a) original image



(b) 0.5bpp



(c) 0.25bpp



(d) 0.125bpp

그림 6 제안한 부호화 방법에 의한 복호화 영상

Fig. 6 Reconstructed image by proposed coding algorithm

표 6에 제안한 방법을 이용하여 Lena 영상을 부호화/복호화 하는데 필요한 시간을 나타내었다. 표에서와 같이 임베디드 시스템의 계산 성능에 의해 PC에서와는 달리 다소 많은 시간이 소비되었다. 제로트리 부호화와 복호화 속도의 경우 bpp가 올라갈수록 소요되는 시간이 늘어났고, 그 이외의 다른 항목들은 영상의 크기에 따라 소비 시간이 크게 달라졌다. 특히 웨이블릿 변환의 경우 많은 부동 소수점 계산이 필요한데, 대부분의 휴대형 임베디드 마이크로 프로세서는 부동 소수점 계산을 위한 보조 프로세서를 내장하고있지 않아 다른 알고리즘 항목에 비해 월등히 많은 계산 시간이 요구되었다. 이러한 문제는 임베디드 프로세서의 성능향상이 지속적으로 이루어지고 있고, DSP를 내장한 임베디드 프로세서가 출시되고 있기 때문에 큰 문제가 되지는 않을 것으로 생각된다. 그리고 기존의 방법과의 부호화 및 복호화 속도 비교는 프로그램 작성 방식과 코드의 최적화 방법에 따라 큰 속도 차이를 보이기 때문에 본 논문에서는 논하지 않았다.

표 6 Lena 영상의 부호화/복호화 소요 시간(단위:sec)

Table 6 Encoding/Decoding running time of the Lena image

항목		bpp				
		0.125	0.5	1.0	3.0	4.0
부 호 화	Image Loading	1.94	1.95	1.94	1.94	1.94
	Foward Wavelet	44.79	44.8	44.83	44.81	44.8
	Build 2D Array	0.03	0.03	0.03	0.03	0.03
	Zerotree Encoding	0.05	0.2	0.37	0.95	1.24
복 호 화	Data Loding	0.0001	0.0001	0.0001	0.0001	0.0001
	Zerotree Decoding	0.1	0.25	0.42	0.99	1.26
	Inverse Wavelet	50.82	51.2	51.76	53.31	53.63
	Write Image	1.09	1.1	1.09	1.09	1.1

5. 결 론

본 논문에서는 시스템 제약이 많은 휴대형 임베디드 마이크로프로세서 기반의 시스템에서 효과적으로 수행될 수 있는 2D 중요계수 배열을 사용한 웨이블릿 부호화기를 제안하였다. 제안된 부호화기는 웨이블릿 계수의 비트 레벨 정보를 2D 중요계수 배열에 저장하고, 제로트리 부호화시 이 정보를 이용하여 계수의 중요도 검사와 비트 스캐닝을 한번에 수행시키는 구조이다. 따라서 기존의 알고리즘이 가지는 부호화 과정의 중복된 계수 스캐닝 과정을 줄일 수 있었으며, 부호화 과정에 필요한 메모리 사용량도 크게 낮출 수 있었다. 또한 비트율 제어에 의한 부호화 완료 조건 검사를 위해 수정된 MSE 기법을 이용하여 웨이블릿 역변환 과정 없이 현재의 계수값 만을 이용하여 비트율 제어가 가능하였으며, 이진 심볼을 엔트로피 부호화하는데 있어서 적응 이진 산술 부호화기를 사용하여 부호화 효율을 증대시켰다. 그리고 제안한 방법을 기존의 EZW, SPIHT 그리고 JPEG 방법들과 성능을 비교한 결과, 기존 방법들에 비해 비슷하거나 비교적 우수한 성능을 나타내고 있음을 알 수 있었다.

앞으로의 연구 과제로는 부호화기의 웨이블릿 변환시 요구되는 시간 단축을 위해 임베디드 시스템의 처리속도 향상에 관한 연구가 진행되어야 할 것이다.

참 고 문 헌

- [1] Sanjaya Kumar, The Codesign of Embedded Systems : A Unified Hardware Software Representation, Kluwer Academic Publishers, 1995.
- [2] C. Sidney Burrus, Ramesh A. Gopinath, Introduction to Wavelet and Wavelet Transforms, Prentice-Hall, New Jersey, 1998.
- [3] J. Y. Yham, S. Ranganath and A. A. Kassim, "Highly scalable wavelet-based video codec for very low bit-rate environment", IEEE Trans. Selected Areas in Comm., vol. 16, no. 1, pp.12-27, Jan. 1998.
- [4] S. N. Efstratiadis, D. Tzovaras and M. G. Strintzis, "Hierarchical partition priority wavelet image compression", IEEE Trans. Image Processing, vol. 5, no. 7, pp. 1111-1123, Jul. 1996.
- [5] Z. Xiong, K. Ramchandran and M. T. Orchard, "Space-frequency quantization for wavelet image coding", IEEE Trans. Image Processing, vol. 6, no. 5, pp.677-693, May 1997.
- [6] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", IEEE Trans. Signal Processing, vol. 41, no. 12, pp.3445-3462, Dec. 1993.
- [7] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, pp.243-250, June 1996.
- [8] CCITT Draft Recommendation T.82 ISO/IEC Draft International Standard 11544 Coded Representation of Picture and Audio Information Progressive Bi-level Image Compression Recommendation, WG9-SIR5.1, Apr. 1992.
- [9] V. R. Algazi, R. R. Estaes Jr., "Analysis based coding of image transform and subband coefficients", In SPIE, Applications of Digital Image Processing XVII, pp.11-21, 1995
- [10] C. Chrysafis, A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression", In Proc. IEEE Data Compression Conf. '97, pp. 241-250, 1997

저 자 소 개



박 성 욱 (朴 成 煜)

1971년 12월 20일생. 1997년 목원대 컴퓨터공학과 졸업. 1999년 인천대 대학원 전자공학과 졸업(석사). 2000년~현재 동 대학원 전자공학과 박사과정. 2002~현재 (주)로직메카 통신연구소 선임연구원

Tel : 032-770-8441, Fax: 032-764-2371

E-mail : psw@incheon.ac.kr



김 영 봉 (金 永 鳳)

1945년 1월 27일생. 1969년 인하대 공대 전기공학과 졸업. 1978년 동 대학원 전기공학과 졸업(석사). 현재 인하공전 전기과 교수

Tel : 032-870-2194, Fax: 032-870-2507

E-mail : ybongkim@inhac.ac.kr



박 종 욱 (朴 種 旭)

1951년 11월 14일생. 1973년 인하대 공대 전자공학과 졸업. 1978년 동 대학원 전자공학과 졸업(석사). 1985년 동 대학원 전자공학과 졸업(공박). 현재 인천대학교 전자공학과 교수

Tel : 032-770-8441, Fax: 032-764-2371

E-mail : jngw@incheon.ac.kr