

정규논문 (Regular Paper)

방송공학회논문지 제19권 제2호, 2014년 3월 (JBE Vol. 19, No. 2, March 2014)

http://dx.doi.org/10.5909/JBE.2014.19.2.227

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

낮은 복잡도의 준무손실 압축을 위한 향상된 예측 기법

손지덕^{a)}, 송병철^{a)†}

Enhanced Prediction for Low Complexity Near-lossless Compression

Ji Deok Son^{a)} and Byung Cheol Song^{a)†}

요 약

본 논문은 영상처리용 SoC에서 외부 메모리 대역폭을 효과적으로 낮추기 위한 near-lossless 이미지 코더의 압축 성능을 향상시키는 새로운 예측 기법을 제안한다. 먼저, RGB 간 correlation을 고려하여 이미 복원된 G 성분을 기반으로 R과 B 성분을 효과적으로 예측하는 inter-color prediction을 수행한다. 다음으로 가변 블록 예측을 통해 예측 성능을 향상시킨다. 마지막으로 이전 프레임에서 sampling된 템플릿 dictionary를 이용해 G 성분 예측 시 최소한의 내부 메모리만을 사용하여 시간 축 예측 성능을 개선시키는 방법을 제안한다. 실험 결과를 통해 자연 영상의 경우 기존 기법 대비 평균적으로 약 30%의 코딩 효율 향상을 보이고, CG 영상의 경우에는 평균 60% 정도의 성능 향상을 보임을 알 수 있다.

Abstract

This paper proposes an enhance prediction for conventional near-lossless coder to effectively lower external memory bandwidth in image processing SoC. First, we utilize an already reconstructed green component as a base of predictor of the other color component because high correlation between RGB color components usually exists. Next, we can improve prediction performance by applying variable block size prediction. Lastly, we use minimum internal memory and improve a temporal prediction performance by using a template dictionary that is sampled in previous frame. Experimental results show that the proposed algorithm shows better performance than the previous works. Natural images have approximately 30% improvement in coding efficiency and CG images have 60% improvement on average.

Keyword : Line-based near-lossless coding, inter-color correlation, variable block size, fast template matching, mode decision.

1. 서 론

최근 HDTV (high definition television), UDTV (ultra-

high definition television), 디지털 카메라 등 영상 기기들의 발달로 고해상도 영상이 널리 사용되고 있다. TV의 경우 4K×2K 크기의 UDTV가 상용화되면서 유력한 차세대 TV로 떠오르고 있다^[1,2]. 고해상도 디지털 기기는 방대한 영상 크기로 인해 더 큰 메모리가 요구되고, 고해상도 영상을 다루는 system on chip (SoC)의 경우 넓은 대역폭을 필요 하는데 이는 결국 높은 비용을 초래한다. 이런 문제점

a) 인하대학교 전자공학과 (Department of Electronic Engineering, Inha University)

† Corresponding Author : 송병철 (Byung Cheol Song)

E-mail: bcsong@inha.ac.kr

Tel: +82-32-860-7413

Manuscript received January 22, 2014 Revised March 4, 2014

Accepted March 4, 2014

을 해결하기 위해 SoC에서 외부 메모리 접근 시 영상을 무손실 혹은 준무손실로 압축 및 복원할 수 있는 최소한의 라인 메모리만을 사용하여 지연을 최소화하는 라인 기반 압축 기법들이 제안된 바 있다^[3,4]. 또한, 이러한 준무손실 압축 기법은 SoC 내 다른 영상처리 모듈 대비 가벼운 연산량을 갖도록 하여 SoC의 gate count에 대한 부담을 최소화해야 한다.

영상 압축은 주로 손실 압축에 초점이 맞춰져 연구되어 왔지만, 의료 영상, image archiving, remote sensing 등 무손실 압축이 필요한 분야도 존재하기 때문에 무손실 압축 기법에 대한 연구도 꾸준히 진행되어 왔다^[5]. 또한, 압축 기법에서 성능에 가장 큰 영향을 주는 예측 부분의 경우 많은 연구가 진행되었는데 예를 들면, JPEG-LS에서는 상위, 좌측, 상좌측 세 개의 에지 중 중간 값을 택하는 median edge detection (MED) 예측을 사용한다^[6,7]. CALIC에서는 수평 방향과 수직 방향의 gradient를 이용해 에지 방향성을 파악하여 예측하는 방식인 gradient adaptive predictor (GAP)을 사용하고^[8], Kim et al. 은 픽셀들의 평균값과 복사값을 이용하는 낮은 복잡도를 갖는 hierarchical average and copy prediction (HACP)를 제안하였다^[10]. Wige et al. 은 잡음 영상의 압축 시 효율을 향상시키기 위해 non-local mean (NLM) 기반 잡음 제거 기법을 이용한 예측을 제안하였다^[11].

실제로 고해상도의 영상들을 완벽하게 무손실로 전송하려면 넓은 대역폭이 필요할 뿐만 아니라 가변 압축률의 특성 때문에 최소 대역폭이 보장되지 않는 문제점이 있다. 그러나 약간의 화질 저하를 허용하는 준무손실 압축을 사용하면 상대적으로 메모리 대역폭을 더 줄일 수 있고, 최소 대역폭을 보장할 수 있다. 표준 이미지 코더인 JPEG-LS도 신뢰할만한 성능을 지닌 준무손실 모드를 제공하지만, 고해상도 영상에 대해 실시간 처리가 가능할 정도로 낮은 연산량을 가지고 있지 않으며, 지연 없이 즉시 인코딩 및 디코딩을 할 수 없다는 단점을 지닌다. 그러므로 좀 더 낮은 복잡도와 지연을 갖는 고성능 준무손실 이미지 코더가 필요하다^[12].

본 논문은 기존의 낮은 지연 및 낮은 복잡도의 준무손실

이미지 코더^[12]의 성능을 추가로 향상시키기 위한 새로운 예측 기법들을 제안한다. 먼저, RGB 성분 간 상관관계를 고려해 이미 복원된 G 성분을 이용해서 R과 B 성분의 예측하는 inter-color correlation prediction (ICP)을 수행한다. 다음으로 복잡도가 높은 영역은 작은 블록 단위 예측을 수행하고, 복잡도가 낮은 영역에서는 큰 블록 단위 예측을 수행하는 가변 블록 예측 기법을 적용한다. 마지막으로, 이전 프레임으로부터 샘플링 된 소정 개수의 템플릿들을 이용해 예측 성능이 좋지 않아 큰 residue를 유발하는 영역들의 residue 값을 대폭 낮출 수 있는 temporal 템플릿 매칭 기반 예측을 수행한다. 실험결과를 통해 제안 기법이 기존 기법과 비슷한 PSNR (Peak Signal-to-Noise Ratio)을 유지하면서 상당히 향상된 압축 성능을 보임을 알 수 있었다.

본 논문은 다음과 같은 순서로 구성되어 있다. II장에서 기존의 낮은 복잡도를 갖는 준무손실 이미지 코더에 대해 리뷰하고, III장에서 제안하는 예측 기법들에 대해 설명한다. IV장에서는 실험 결과를 보이고, V장에서 본 논문의 결론을 짓는다.

II. 기존 기법

본 논문에서는 라인 단위로 압축을 수행하여 기존의 저지연 준무손실 이미지 코더를^[12] 기반으로 새로운 예측 기법을 제안하여 더 높은 압축률을 얻고자 하는데 이에 앞서 기존의 이미지 코더에 대해 설명한다. 각 과정은 블록 단위로 수행되며 복원된 직전 한 라인을 이용해 예측하고 residue를 구해 양자화하여 코딩한다. Rate control (RC)을 이용하는 경우에는 블록 단위 4개 크기인 32×1 블록마다 수행되어 quantization parameter (QP) 값을 업데이트 해줌으로써 원하는 비트량에 맞춰 압축을 수행할 수 있다.

1. 예측

예측 과정은 현재 코딩하려는 블록과 가장 비슷한 값을 갖는 블록을 탐색하는 과정으로 두 블록 간 차이인 residue가 작을수록 비트 발생량이 작기 때문에 예측 성능이 곧

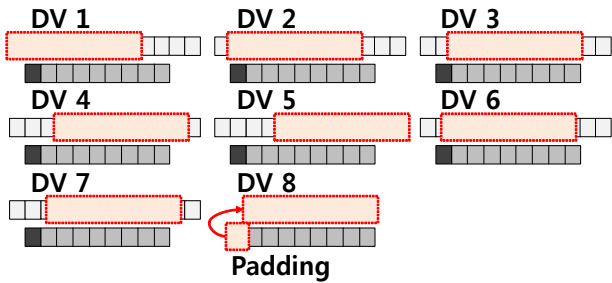


그림 1. DV 탐색의 예측 모드
 Fig. 1. Prediction modes of DV search

압축 효율에 큰 영향을 미친다. 여기에서 사용되는 displacement vector (DV) search 방법은 복원된 픽셀을 이용해 만든 8개의 예측 모드 중 원본 블록과 가장 비슷한 예측기를 선택하는 방법이다. 그림 1을 보면 중간 밝기 블록은 현재 코딩하려는 블록을 나타내고, 가장 밝은 블록은 현재 블록의 직전 라인에 있는 블록을 뜻하며 이 값들을 이용해 8개의 예측 모드를 생성한다. 가장 어두운 블록은 현재 블록의 바로 이전 픽셀로 복원된 값이다. 그림 1과 같이 1~5번 예측 모드는 각각 (-2,-1), (-1,-1), (0,-1), (1,-1), (2,-1) 위치의 픽셀들을 이용한 것으로 점선으로 묶인 블록이 예측 모드가 된다. 6, 7번 예측 모드는 bilinear를 이용해 생성된 (-0.5,-1), (0.5,-1) 위치의 half-pel을 이용하고, 8번 모드는 현재 블록의 바로 왼쪽에 위치한 하나의 픽셀을 반복하여 생성한다. 8개의 예측 모드 중 최소 Sum of Absolute Differences (SAD) 값을 갖는 모드를 선택하여 예측기로 사용하고, 현재 블록의 DV가 이전 블록의 DV와 같다면 1비트 크기의 모드 정보를 전송하여 DV 값을 코딩 하지 않고 이전 블록의 DV를 그대로 사용한다.

2. 양자화

양자화 과정에서는 영상의 화질을 제어하기 위해 최대 코딩 에러를 나타내는 QP를 이용한다. 예를 들어, QP가 2 라면 최대 코딩 에러는 2가 되고, QP를 0으로 두면 무손실 모드로 손실 없이 원본과 완전히 같은 영상을 복원할 수 있다. 식 (1)을 통해 양자화를 수행하고, 식 (2)를 통해 양자화된 값을 복원시키는 역양자화를 수행한다.

$$Q = \begin{cases} \frac{D+QP}{2QP+1}, & D > 0 \\ -\frac{|D|+QP}{2QP+1}, & otherwise \end{cases} \quad (1)$$

$$DQ = Q \times (2QP + 1) \quad (2)$$

D는 residue고, Q는 양자화된 residue, DQ는 역양자화된 residue를 나타낸다. 블록 내의 모든 Q 값이 0이라면, 이 블록은 코딩하지 않고 스킵 모드로 진행한다. 스킵 모드인 경우에는 블록 내의 모든 값을 0으로 역양자화 한다.

3. 엔트로피 부호화

Residue를 코딩하기 위해 현재 블록을 4x1 서브 블록으로 나누고, 코드 길이와 residue 값을 나눠서 fixed length coding (FLC)을 수행한다. 코드 길이는 서브 블록 단위로 정하고, 서브 블록 내의 가장 긴 코드 길이를 따른다. 예를 들어, 그림 2와 같은 경우라면 4번째 값인 30의 길이가 5로 가장 길기 때문에 이 서브 블록의 코드 길이는 5가 되고, 부호 길이까지 고려한 총 코드 길이는 6이 된다. DV를 코딩할 때와 마찬가지로 코드 길이가 이전 블록의 코드 길이와 같거나 ±1 차이가 난다면 2비트의 코드 길이 정보를 이용하여 코드 길이를 따로 코딩하지 않고, 그 외의 경우에는 코드 길이도 함께 코딩 해준다. Residue 값은 코드 길이에 맞춰 코딩되는데 그림 2를 보면, 코드 길이가 6이므로 그에 맞춰 001001, 111001, 110000, 011110을 이어서 코딩한다. 하나의 서브 블록 코딩이 끝나면 다음 서브 블록을 코딩하고, 이는 블록 내의 모든 서브 블록 코딩이 끝날 때까지 반

	9	-7	-16	30
Sign	0(+)	1(-)	1(-)	0(+)
FLC	00001001	11111001	11110000	00011110
Length	/ ₀ = 5	/ ₁ = 4	/ ₂ = 5	/ ₃ = 6
Length of the sub-block = max(/ ₀ , / ₁ , / ₂ , / ₃) = 6				

그림 2. 서브 블록 코딩 예
 Fig. 2. The example of sub-block coding

복된다.

4. Rate control

RC는 버퍼 상태에 따라 QP를 조절하여 버퍼의 오버플로우를 방지하고 원하는 압축률로 코딩을 가능하게 한다. RC는 4개의 블록 단위로 수행되는데 이를 Rate Control Unit (RCU)라 하고, RC 수행을 위해 복잡도를 측정하여 타겟 비트량에 따라 프레임, 라인, RCU 레벨 별로 비트를 할당 해주며 버퍼의 사용량에 따라 할당량을 수정함으로써 QP를 조절해준다^[12,13]. 현재 블록이 영상 내의 다른 블록들보다 더 복잡하면 더 많은 비트를 할당하고, 덜 복잡하면 더 적은 비트를 할당한다. 또한, 버퍼 사용량이 특정한 한계치보다 크면 QP 값을 증가시켜 비트 발생량을 줄여서 버퍼의 오버플로우를 막고, 특정한 한계치보다 작으면 QP 값을 감소시켜 복원 영상의 질을 향상시키면서 타겟 비트량에 맞춰 압축을 수행할 수 있다.

III. 제안 기법

II장에서 언급한 준무손실 이미지 코더는 SoC의 메모리 대역폭을 줄여주지만, 압축 성능은 JPEG-LS보다 떨어진다. 본 논문은 이 기존 기법의 압축 효율을 향상시키기 위한 진보된 예측 방법을 제안한다. 첫 번째로, 컬러 간에는 상관관계가 존재하기 때문에 복원된 G 성분을 이용하여 R과 B 성분의 예측기를 구할 수 있다. 두 번째로, 예측 모드와 residue의 비트량이 예측 블록 크기에 따라 trade-off 관계에

있기 때문에 가변 블록 크기를 이용하면 비트 이득을 얻을 수 있다. 마지막으로 DV search 모드 중 선택 빈도가 낮은 모드를 대신해 템플릿 기반 예측을 사용하여 복잡한 텍스처 영역의 예측 성능을 향상시킬 수 있다.

그림 3 (a)는 전체 알고리즘 블록도로 입력 영상이 들어오면 예측을 수행한 뒤 residue를 계산해 양자화하고, FLC를 수행한다. 양자화된 값은 역양자화 후 복원 픽셀을 생성하여 예측부에서 이용한다. 그림 3 (b)는 제안하는 예측 파트의 동작도다. 그림과 같이 RGB 모두 DV search를 수행하는 예측과 G 성분에 대해서는 DV search와 템플릿 매칭을 이용해 예측 모드를 구하고, R, B 성분은 Inter-color correlation based prediction (ICP)를 수행하는 예측 방법 중 성능이 더 좋은 것을 선택한다. 그런 다음, 그림과 같이 4가지 크기의 가변 블록을 이용하여 얻어진 최적의 예측 모드를 통해 예측을 수행한다.

1. Inter-color correlation based prediction

그림 4 (a)는 1280×720 크기인 parkrun 영상의 첫 번째 프레임에 존재하는 모든 픽셀들의 R, G 성분의 밝기 값을 나타낸 것으로 한 프레임 내에서 R 성분과 G 성분 간에 상관관계가 있음을 확인할 수 있다. 실제 압축 과정에서는 64×1 블록 단위로 코딩을 수행하기 때문에 64×1 크기의 블록 내에서 R과 G 성분의 상관관계를 확인해본다. 그림 4 (b), (c)는 그림 4 (a)에 사용된 프레임과 동일한 프레임의 (200, 10), (300, 15)번째 64×1 블록에 대해 R과 G 성분의 밝기 값을 나타낸 것이다. 프레임 전체의 상관관계보다 더 높은 상관관계를 갖는 것을 확인할 수 있고, 선형 모델로

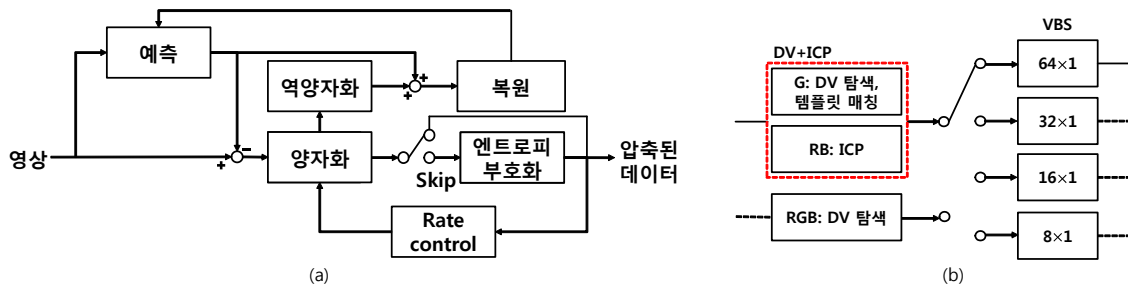


그림 3. 제안 기법의 블록도 (a) 전체 알고리즘 (b) 제안하는 예측 알고리즘

Fig. 3. The block diagram of proposed algorithm (a) The entire algorithm (b) The proposed prediction algorithm

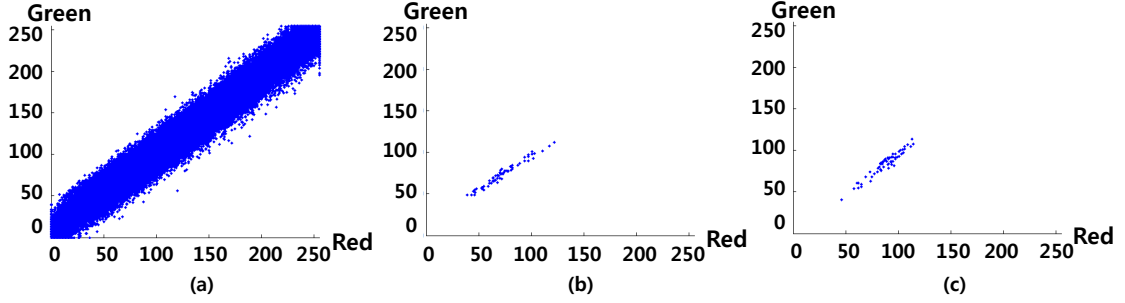


그림 4. Red와 Green 성분의 상관관계 (a) 첫 번째 프레임 (b) (200,10)번째 64×1 블록 (c) (300,15)번째 64×1 블록
 Fig. 4. The correlation between Red and Green components (a) First frame (b) (200,10)th 64×1 block (c) (300,15)th 64×1 block

이러한 관계를 잘 나타낼 수 있다. 그러므로 복원된 G 성분을 선형 모델에 적용시켜 R, B 성분의 예측기를 얻을 수 있다. 이렇게 얻은 예측기는 확률적으로 DV search로 얻은 예측기보다 더 정확하게 예측되는 경우가 많아 낮은 residue를 생성할 수 있어 코딩 효율을 높일 수 있다. 이러한 방법을 우리는 ICP라 한다.

예측기의 선형 모델은 기울기와 절편을 이용해 다음과 같이 나타낼 수 있다^[4].

$$r_{pred(i,j)} = \alpha_R \times g_{rec}(i,j) + \beta_R \quad (3)$$

식 (3)은 R 성분의 예측기인 r_{pred} 를 선형 모델로 나타낸 것으로 g_{rec} 은 G 성분의 복원된 값이고, α_R, β_R 은 R 성분의 선형 모델에 필요한 기울기와 절편을 나타내는 파라미터이다. 코딩 성능을 높이기 위해서는 R 성분과 r_{pred} 의 차 값이 작아져야 하는데 이러한 residue의 에너지를 나타내는 J를 정의한다.

식 (4)는 M×N 사이즈인 R 성분 블록과 r_{pred} 간의 residue 에너지 J를 나타낸 것이다. J를 최소화하는 α_R, β_R 값을 찾기 위해 편미분을 이용하면, 식 (5), (6)을 얻을 수 있다. \bar{x}, \bar{y} 는 각각 현재 블록을 둘러싸고 있는 이웃한 픽셀들의 $x(i,j)$ 와 $y(i,j)$ 의 평균값이다. R 성분에 대한 정보가 없으면

α_R, β_R 값을 추가로 코딩해 줘야하기 때문에 이러한 비트 발생량을 없애기 위해 디코더에서 추가 정보 없이 사용 가능한 causal 정보들을 이용한다. R 성분의 코딩을 마친 후, B 성분도 R 성분과 동일한 방법으로 b_{pred} 를 구할 수 있다.

$$\alpha_R = \frac{R_{g_{rec}r}}{R_{g_{rec}g_{rec}}}, \beta_R = \bar{r} - \alpha_R \cdot \bar{g_{rec}} \quad (5)$$

$$R_{XY} = \sum_{all(i,j) \in Neighbor} (x(i,j) - \bar{x})(y(i,j) - \bar{y}) \quad (6)$$

그림 5와 같이 기존에는 RGB 성분을 모두 묶어 하나의 DV를 구해 예측기로 사용하는 반면 제안 기법은 G 성분만 DV search를 수행하여 더 낮은 residue를 구할 수 있다. 또

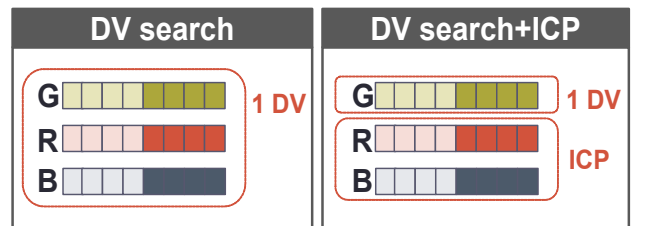


그림 5. DV search와 ICP 수행 방법
 Fig. 5. The method of DV search and ICP

$$J = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (r(i,j) - r_{pred}(i,j))^2 = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (r(i,j) - (\alpha_R \times g_{rec}(i,j) + \beta_R))^2 \quad (4)$$

한, 많은 영역에서 G 성분과 R, B 성분이 선형 모델로 표현 가능한 상관관계를 갖기 때문에 ICP를 이용해 얻은 R, B 예측기는 매우 낮은 residue를 얻을 수 있다. 일부, 선형 모델이 잘 들어맞지 않는 영역을 커버하기 위해 ICP와 DV search를 선택적으로 사용하고, 이러한 예측 모드를 알려주는 1비트의 정보를 전송한다.

2. Variable Block Size Prediction

블록 크기가 작을수록 예측의 정확도가 증가해 낮은 residue를 갖지만 추가 전송해야 할 예측 모드의 비트량이 증가하고, 블록 크기가 클수록 추가 전송 비트량이 줄지만 예측의 정확도가 떨어져 높은 residue를 갖는다. 평탄한 영역은 64x1 블록 크기를 사용하는 것과 8x1 블록 크기를 사용하는 것의 예측 오차가 크게 차이 나지 않는다. 예측 모드 비트량은 현재 블록의 DV와 이전 블록의 DV가 같은지 판단하는 1bit와 현재 블록의 DV와 이전 블록의 DV가 다른 경우, DV 값을 코딩하는데 필요한 3bit로 구성된다. 그러므로 8x1 블록 크기를 사용했을 때 모든 블록의 DV가 동일한 값을 가져 DV 코딩이 전혀 없는 경우에는 최소 8비트의 비트량이 필요하고, 모든 블록에 대해 DV 코딩을 해줘야 하는 경우에는 최대 32비트의 비트량을 필요로 하지만, 64x1 블록을 사용했을 때 필요한 예측 모드 비트량은 이전 DV와 동일한 경우에는 1비트, DV 코딩이 필요한 경우에는 4비트로 64x1 블록을 사용했을 때 최소 4비트에서 최대

30비트 이상의 예측 모드 이득을 볼 수 있다. 그러나 복잡도가 높은 영역에 64x1 블록을 사용하면 이러한 예측 모드 이득이 역전될 정도의 큰 residue가 발생해 코딩 효율을 저하시킬 수 있다. 이처럼 영역의 복잡도에 따라 비트 발생량이 다르기 때문에 다양한 블록 크기를 이용해 예측을 수행하면 코딩 성능을 개선할 수 있다.

$$VBS_{Final} = \arg \min_{VBS} (bit_{VBS}) \tag{7}$$

$$VBS \in \{8 \times 1, 16 \times 1, 32 \times 1, 64 \times 1\}$$

식 (7)의 bit_{VBS} 는 variable block size (VBS)에 따라 발생하는 비트량이고, VBS_{Final} 은 최종 블록 크기로 비트 발생량이 가장 작은 VBS 값을 가진다. 기존 기법은 8x1 블록 크기 단위로 코딩했지만 VBS를 이용하기 위해 64x1 블록 크기 단위로 코딩하고, 그림 6과 같이 64x1 블록 내에서 실제 비트 발생량에 따라 8x1, 16x1, 32x1, 64x1 중 서브 블록 크기를 선택한다. 예를 들어, 8x1이 서브 블록 단위로 선택되었다면 첫 번째 8x1 블록에 대해 완전히 코딩한 뒤, 두 번째 8x1 블록에 대해 코딩 과정을 거친다. 이 과정은 코딩 블록 크기인 64x1이 모두 코딩 될 때까지 반복적으로 이루어져, 총 8번의 8x1 블록 코딩을 수행한다. 4가지의 블록 크기에 대한 정보를 전송하기 위해 2비트 크기의 모드를 할당한다.

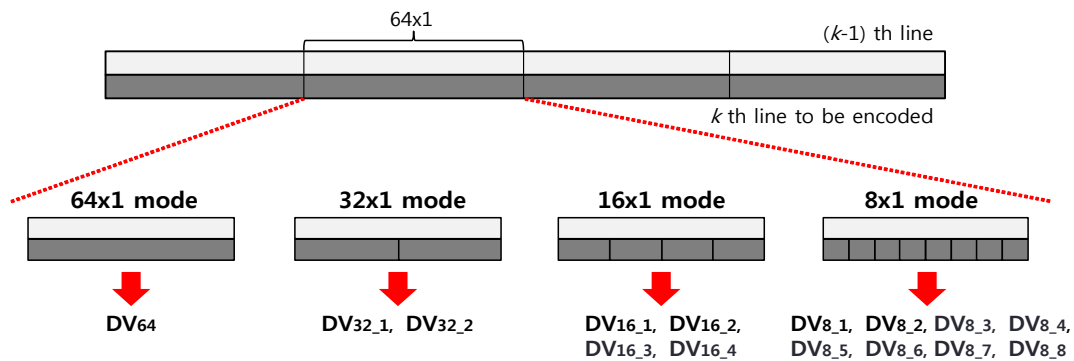


그림 6. 가변 블록 예측의 수행 방법
Fig. 6. The method of variable block size prediction

3. Template based prediction

기본적으로 G 성분에 대한 예측을 위해 인접한 영역에 있는 복원된 픽셀을 이용하여 예측기를 생성하는 DV search 방법을 이용한다. 그렇기 때문에 복잡한 텍스처가 나타나면 이를 예측하기 힘들어 큰 residue 값이 발생하곤 한다. 이러한 점을 보완하기 위해 현재 프레임에서 DV search의 예측 오차가 큰 순서대로 N개의 패턴을 찾아 저장한 후 다음 프레임에서 DV search 오차가 큰 블록에 한해 템플릿 매칭을 이용하면 residue를 낮출 수 있다.

그림 7과 같이 템플릿과 예측기로 구성된 패턴을 저장하기 위해 t번째 프레임에서 G 성분의 DV search를 수행한 뒤, 양자화/역양자화 과정을 거쳐 복원된 G 성분 값을 생성한다. 디코더와 인코더에서 동일한 패턴을 갖기 위해서는 원본 값이 아닌 복원된 G 값을 이용해야 한다. 최적의 예측 모드가 DV search를 이용한 것이라면 복원된 G 성분과 선택된 예측기의 SAD를 기준으로 패턴을 저장하고, 최적의 예측 모드가 템플릿 매칭을 이용해서 얻어진 것이라면 템플릿 매칭을 제외한 다른 예측 모드 중 가장 낮은 SAD를 기준으로 패턴을 저장한다. 왜냐하면 템플릿 매칭을 적용해서 SAD가 낮아진 경우에 패턴을 저장하지 않게 되면, 다음 프레임에서 그 텍스처가 발생했을 때 residue가 커지기 때문이다. 또한, 그림 7과 같이 t-1번째 프레임에서 현재 블록을 둘러싸는 Γ 모양의 템플릿을 이용해 t번째 프레임에서 저장된 템플릿 중 가장 비슷한 것을 탐색하여 그 템플

릿에 매칭되어 있는 예측기가 DV search의 결과로 생성된 예측기보다 현재 블록과 유사하다면 최종 예측기로 선정한다. DV search의 8개 예측 모드 중 (-1, -1), (1, -1) 모드는 다른 모드에 비해 선택되는 확률이 낮기 때문에 이들 모드를 템플릿 매칭 모드와 DC 모드로 대체한다. DC 모드는 직전 라인의 $n \times 1$ 블록 평균값을 복사하여 만든 예측기를 뜻한다. 또한, G 성분만이 아닌 RGB 모든 성분에 대해서 템플릿 매칭 모드를 수행하려면, N개가 아닌 $3 \times N$ 개의 패턴이 필요하게 되는데 대부분의 영상에서 ICP로 선택되는 비율이 70% 이상이기 때문에 3배의 메모리를 사용하더라도 그만큼의 성능 향상을 기대할 수 없어 템플릿 매칭 모드는 ICP를 수행할 때에만 적용한다.

4. Computational optimization

템플릿 기반 예측 모드는 한 샘플마다 템플릿 개수인 N 번의 SAD 계산을 수행해야 하기 때문에 연산량이 많다. 또한, 가능한 모든 모드에 대해서 불필요한 후보군을 제거하지 않고, 모든 모드의 비트량을 계산해 가장 최적의 모드를 선택하기 때문에 기존 대비 연산량이 증가하였다. 제안 기법의 연산량을 줄이기 위해 크게 두 가지의 최적화를 수행한다. 첫 번째로, 템플릿 매칭의 고속화 기법을 적용하여 템플릿 기반 예측의 속도를 증가시키고, 두 번째로는 모드 결정을 통해 일부 모드에서만 비트량을 계산해 최적의 모드를 선택한다.

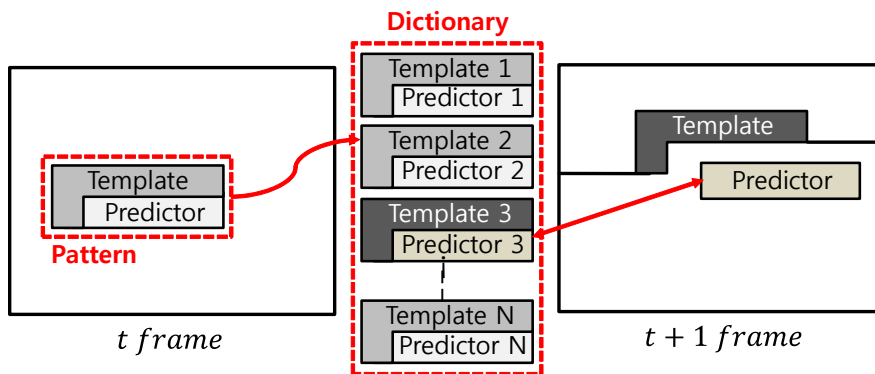


그림 7. 템플릿 기반 예측의 수행 방법
 Fig. 7. The method of template based prediction

4.1 Fast template matching

예측 모드에서 시간을 가장 많이 차지하는 부분은 템플릿 매칭 파트이다. 템플릿 개수가 늘어날수록 시간은 오래 걸리기 때문에 템플릿 매칭의 속도를 증가시킬 필요가 있다. Full search의 속도를 증가시키기 위해 다중 해상도를 구성하여 불필요한 후보군을 제거하는 방식인 multi resolution search algorithm (MRSA)을 이용한다^[15]. MRSA의 적용을 용이하게 하기 위해 기존의 Γ 모양의 템플릿을 사용하지 않고, 그림 8 (a)와 같이 현재 블록과 동일한 길이의 직전 라인을 템플릿으로 사용한 다음, 템플릿의 밝기 값을 이용하여 합 피라미드 구조를 생성해 템플릿을 다중 해상도로 구성한다.

$$X^l(i) = X^{l+1}(2i-1) + X^{l+1}(2i), \quad 1 \leq i \leq 2^l \quad (8)$$

식 (8)의 X^l 은 각 레벨의 히스토그램으로 2^l 의 크기를 갖고, 그림 8 (b)와 같이 $\{X^0, X^1, \dots, X^l, \dots, X^L\}$ 로 구성된 히스토그램의 합 피라미드를 생성한다. $X^L = X$ 로 템플릿 크기의 bin을 갖는 히스토그램이다. 그림 8 (b)를 보면 $X^L = X^4$ 의 가장 오른쪽에 있는 2개의 bin을 더해서 X^3 의 가장 오른쪽 bin이 생성된다. N개의 템플릿에 대해 미리 합 피라미드를 생성해놓고, 현재 블록의 템플릿이 주어지면 합 피라미드로 구성하여 가장 낮은 레벨에서부터 $L_1 - norm$ 을 계산한다. d 는 $L_1 - norm$ 을 나타내고, $d^l(X, Q)$ 은 $d(X^l, Q^l)$ 를 나타낼 때 계산한 $d^l(X_n, Q)$ 가 d_{min} 보다

크면 X_m 을 후보군에서 제거하고, 다음 후보에 대해 $d^0(X_{n+1}, Q)$ 을 계산한다. 계산한 $d^l(X_n, Q)$ 가 d_{min} 보다 작다면 다음 레벨에서 $d^{l+1}(X_n, Q)$ 을 계산한다. 이와 같은 과정을 반복하여 레벨이 L에 도달해 $d^L(X_n, Q)$ 도 d_{min} 보다 작다면, d_{min} 에 $d^L(X_n, Q)$ 을 대입하고 다음 후보에 대해 다시 $d^0(X_{n+1}, Q)$ 을 계산한다. n이 N을 넘으면 마지막 $d^0(X_{n+1}, Q)$ 을 생성한 가장 잘 매칭된 후보를 선택한다.

4.2 Mode decision

제안 기법에서는 4가지의 가변 블록 모드와 2가지의 예측 모드, 총 8가지의 모드 중 가장 최적인 모드를 선택하여 코딩하고 있다. 하지만 이는 1 pass로 코딩 하는 것 대비 약 8배 정도의 연산량을 가지기 때문에 이를 줄이기 위해 미리 예측 모드를 결정한다. ICP로 코딩한 결과와 DV 탐색으로 코딩한 결과는 대체로 블록 크기와 관계없이 비슷한 경향을 보인다. 64×1 로 코딩 했을 때 ICP가 더 좋은 결과를 냈다면, 32×1 로 코딩 했을 때도 ICP가 더 좋은 결과를 낼 확률이 높다는 것이다. 따라서, 우리는 64×1 크기로 템플릿 매칭을 제외한 ICP와 DV search 예측을 수행하여 발생 비트량을 카운트한다. 비트 카운트를 수행하기 위해 residue를 양자화한 뒤, 서브 블록 내에서 절대값이 가장 큰 값을 선택하고, 그것의 코드 길이를 측정한다. 이 과정을 통해 실제 발생 비트량과 유사한 비트량을 계산해낼 수 있기 때문에 ICP와 DV search 중 더 나은 결과를 내는 예측 모드를

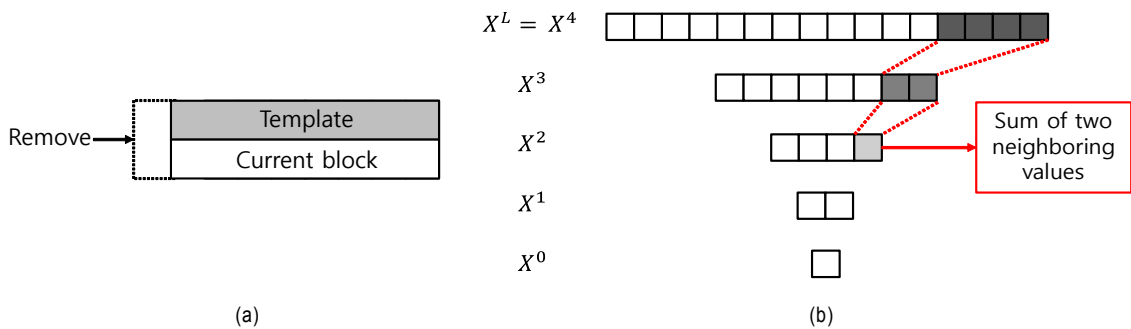


그림 8. 템플릿 매칭 고속화를 위한 합 피라미드 (a) 수정된 템플릿 모양 (b) 합 피라미드 구성
 Fig. 8. Sum pyramid for fast template matching (a) Modified template (b) The composition of sum pyramid

선택한다. 다음으로, 블록 크기를 정하기 위해 이전 단계에서 결정된 예측 모드를 이용하여 4개의 블록 크기에 대한 비트 카운팅을 수행하고, 가장 낮은 비트량을 갖는 모드를 선택한다.

IV. 실험

제안 기법의 성능 평가를 위해 9개의 1280×720 자연 영상: bigships, city, crew, jets, mobcal, parkrun, shields, shuttle, stockholm을 사용하였고, 다양한 웹 페이지를 캡처한 1280×720 크기의 CG 영상 4개를 사용하였다. II장에서 설명한 기존 기법 [12]과 제안 기법을 비교하기 위해 가변 블록 크기를 적용하지 않은 결과에 대해서는 8×1 블록 단위로 코딩을 수행하였고, 예측 방법을 제외한 양자화 및 코딩 방법은 동일하다. 성능 비교를 위해 QP가 1 또는 2일 때, 기존 기법과 기존 기법에 ICP를 적용한 결과, ICP와 가변 블록 기반 예측을 적용한 결과, 3가지 제안 기법을 모두 적용한 결과를 각각 비교한다. 템플릿 기반 예측은 템플릿의 개수를 500개에서 1500개까지 늘리면서 실험하였다.

표 1은 자연 영상의 QP가 1, 2일 때, 기존 기법과 제안 기법의 결과를 나타낸다. 모든 결과는 2~11번 프레임의 결과를 평균 낸 것이고, 압축률의 향상률을 보기 위해 다음과 같은 식 (9)를 이용한다.

$$\text{향상률} = \frac{\text{압축률}_{\text{제안기법}} - \text{압축률}_{\text{기존기법}}}{\text{압축률}_{\text{기존기법}}} \times 100 \quad (9)$$

ICP만 적용했을 때 PSNR은 기존 기법과 거의 유사하지만 QP에 따라 압축률은 평균 21%, 23%의 향상을 보임을 알 수 있다. 대부분의 영상들은 RGB간의 상관관계가 높기 때문에 ICP의 예측으로 residue가 크게 낮아질 뿐만 아니라 80~90%가 ICP로 예측되고, 나머지 경우만 DV search로 예측되기 때문에 이처럼 큰 성능 향상을 보인다. 표 2는 CG 영상들의 기존 기법과 제안 기법의 결과를 나타낸 것으로 ICP를 적용했을 때 residue 감소가 더 크게 일어나 평균

표 1. 자연 영상의 성능 비교

Table 1. The performance comparison of natural images

Image	Algorithms	QP=1		QP=2	
		압축률	PSNR [dB]	압축률	PSNR [dB]
Bigships	기존 기법 [12]	2.307	49.92	2.825	45.21
	ICP 적용	2.768	49.93	3.409	45.25
	ICP+가변 블록	2.875	49.93	3.532	45.24
	제안 기법	2.901	49.93	3.572	45.24
City	기존 기법 [12]	2.165	49.92	2.612	45.19
	ICP 적용	2.817	49.92	3.397	45.21
	ICP+가변 블록	2.885	49.93	3.494	45.22
	제안 기법	2.900	49.93	3.518	45.22
Crew	기존 기법 [12]	2.942	49.99	3.768	45.37
	ICP 적용	3.452	50.00	4.483	45.43
	ICP+가변 블록	3.535	49.98	4.523	45.39
	제안 기법	3.562	49.98	4.580	45.40
Jets	기존 기법 [12]	2.956	49.99	3.870	45.45
	ICP 적용	3.742	50.03	5.020	45.54
	ICP+가변 블록	4.000	49.99	5.315	45.53
	제안 기법	4.055	49.99	5.409	45.55
Mobcal	기존 기법 [12]	1.793	49.93	2.117	45.18
	ICP 적용	2.078	49.95	2.484	45.20
	ICP+가변 블록	2.114	49.95	2.543	45.19
	제안 기법	2.130	49.94	2.564	45.19
Parkrun	기존 기법 [12]	1.587	49.95	1.849	45.19
	ICP 적용	2.040	49.97	2.411	45.21
	ICP+가변 블록	2.116	49.97	2.527	45.22
	제안 기법	2.120	49.97	2.532	45.22
Shields	기존 기법 [12]	2.011	49.97	2.411	45.24
	ICP 적용	2.276	50.02	2.752	45.27
	ICP+가변 블록	2.304	50.02	2.796	45.27
	제안 기법	2.311	50.02	2.806	45.27
Shuttle	기존 기법 [12]	3.843	50.20	5.805	46.01
	ICP 적용	4.683	50.17	7.208	46.00
	ICP+가변 블록	4.838	50.10	7.535	45.98
	제안 기법	4.915	50.11	7.719	46.02
Stockholm	기존 기법 [12]	2.048	49.92	2.467	45.17
	ICP 적용	2.401	49.92	2.901	45.18
	ICP+가변 블록	2.505	49.92	3.045	45.18
	제안 기법	2.518	49.92	3.062	45.17
Average	기존 기법 [12]	2.406	49.98	3.080	45.33
	ICP 적용	2.917	49.99	3.785	45.37
	ICP+가변 블록	3.019	49.98	3.923	45.36
	제안 기법	3.046	49.98	3.974	45.36

표 2. CG 영상의 성능 비교

Table 2. The performance comparison of CG images

Image	Algorithms	QP=1		QP=2	
		압축률	PSNR [dB]	압축률	PSNR [dB]
Web 1	기존 기법 [12]	5.878	52.48	6.960	48.20
	ICP+가변 블록	7.919	51.21	9.298	47.01
	ICP+VB 적용	8.829	52.14	10.719	47.18
Web 2	기존 기법 [12]	6.486	53.64	7.657	48.88
	ICP 적용	8.419	52.34	10.009	48.44
	ICP+가변 블록	9.495	52.93	11.646	47.70
Web 3	기존 기법 [12]	7.361	50.88	8.579	47.95
	ICP 적용	9.414	51.77	11.158	47.34
	ICP+가변 블록	10.668	50.24	13.329	47.43
Web 4	기존 기법 [12]	5.442	53.53	6.571	49.02
	ICP 적용	7.025	53.09	8.580	48.33
	ICP+가변 블록	7.639	53.08	9.500	48.11
Average	기존 기법 [12]	6.292	52.63	7.442	48.51
	ICP 적용	8.194	52.10	9.761	47.78
	ICP+가변 블록	9.158	52.10	11.299	47.61

30%, 31%의 압축률 향상을 보인다.

표 1의 기존 기법에 ICP와 가변 블록 예측을 적용한 경우, 기존 기법 대비 자연 영상은 25%, 27%의 압축률 향상을 보이고, CG 영상은 46%, 52%의 향상을 보인다. 가변 블록 예측을 적용하면서 코딩 블록 별로 블록 크기를 전송하는 2비트가 추가되었고, 평탄한 영역이 많아 큰 크기의 블록이 많이 선택될수록 코딩 효율은 높아진다. 자연 영상에 비해 CG 영상은 평탄한 영역이 많고 복잡하지 않아 가변 블록 예측 성능의 향상 폭이 훨씬 크다.

표 1의 자연 영상에 제안 기법을 적용한 경우, 템플릿은 500개를 사용하였다. 템플릿 기반 예측까지 모두 적용한 제안 기법은 기존 기법에 비해 27%, 29% 성능 향상을 보인다. 표 3은 CG 영상의 템플릿 개수에 따른 템플릿 기반 예측 성능을 나타낸다. CG 영상은 템플릿이 500개 일 때, 기존 기법 대비 56% 64%의 성능 향상을 보인다. CG 영상은 자연 영상보다 평균 residue는 주로 낮지만 특정 임계값 이상의 residue 값은 더 많기 때문에 템플릿을 이용해서 예측하게 되면 residue 값이 비슷비슷한 자연 영상에 비해 더

표 3. CG 영상의 템플릿 개수에 따른 성능 비교

Table 3. The performance depending on the number of templates in CG images

Image	Template 수	QP=1		QP=2	
		압축률	PSNR [dB]	압축률	PSNR [dB]
Web 1	500	9.471	51.79	11.572	47.28
	700	9.664	51.67	11.840	47.02
	1000	9.984	51.68	12.240	47.30
	1500	10.424	51.86	12.835	47.43
Web 2	500	10.239	52.43	12.705	48.14
	700	10.535	52.63	13.087	48.06
	1000	10.967	52.42	13.707	48.14
	1500	11.689	52.43	14.718	48.17
Web 3	500	11.516	52.55	14.449	47.84
	700	11.820	52.54	14.903	47.79
	1000	12.258	52.56	15.521	47.01
	1500	12.912	52.61	16.472	47.68
Web 4	500	8.145	53.28	10.214	48.34
	700	8.316	53.30	10.439	48.30
	1000	8.536	53.24	10.756	48.31
	1500	8.806	53.07	11.153	48.36
Average	500	9.843	52.51	12.235	47.90
	700	10.084	52.54	12.567	47.79
	1000	10.436	52.48	13.056	47.69
	1500	10.958	52.49	13.795	47.91

높은 효과를 볼 수 있다. 템플릿 개수를 500개, 700개, 1000개, 1500개 순으로 늘리면 QP가 1일 때 기존 기법 대비 56%, 60%, 66%, 74%의 성능 향상을 보이고, QP가 2일 때 64%, 69%, 75%, 85%의 성능 향상을 보인다.

그림 9는 제안 기법과 기존 기법 및 JPEG-LS의 준무손실 모드 코딩 성능을 비교한 것이다. 제안 기법은 템플릿 개수 500개를 적용하였다. 그림 9 (a), (b), (c)는 자연 영상으로 제안 기법은 기존 기법에 비해 상당히 향상되었고, JPEG-LS 결과와 비슷한 수준을 유지한다. (d), (e), (f)는 CG 영상으로 기존 기법과 JPEG-LS는 비슷한 성능을 보이며 제안 기법은 이들 기법보다 훨씬 더 높은 성능을 보이는 것을 알 수 있다.

표 4는 자연 영상의 제안 기법 및 JPEG-LS의 수행 시간을 나타낸 것이고, 표 5는 CG 영상에 대한 수행 시간을 나

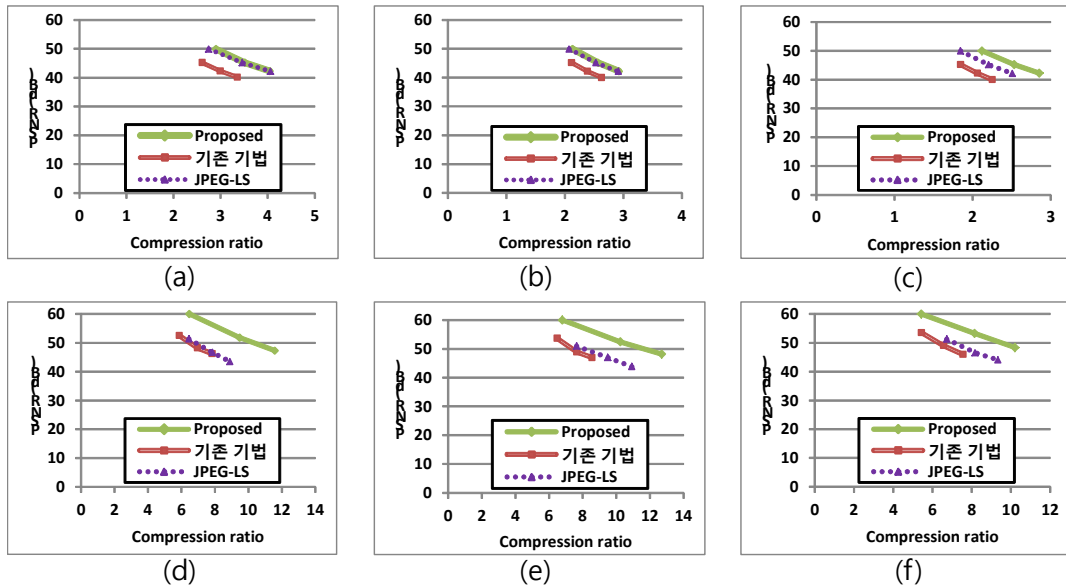


그림 9. JPEG-LS와의 성능 비교 (a) City, (b) Mobcal, (c) Parkrun, (d) Web 1, (e) Web 2 (f) Web 4
 Fig. 9. The performance comparison to JPEG-LS (a) City, (b) Mobcal, (c) Parkrun, (d) Web 1, (e) Web 2 (f) Web 4

표 4. 자연 영상의 연산 시간 결과
 Table 4. The computing time of natural images

Image	Time (sec)				
	Full search	Fast template matching	ICP and fixed block size	Mode decision	JPEG-LS
Bigships	2.885	2.131	0.423	0.993	0.875
City	3.607	2.278	0.469	1.052	0.885
Crew	1.943	1.848	0.339	0.917	0.794
Jets	1.996	1.827	0.339	0.897	0.814
Mobcal	5.218	3.211	0.670	1.249	0.929
Parkrun	6.602	3.465	0.822	1.337	0.932
Shields	4.527	2.545	0.520	1.133	0.9
Shuttle	1.779	1.698	0.335	0.856	0.592
Stockholm	3.193	2.262	0.461	1.047	0.91
Average	3.528	2.363	0.486	1.053	0.848

표 5. CG 영상의 연산 시간 결과
 Table 5. The computing time of CG images

Image	Time (sec)				
	Full search	Fast template matching	ICP and fixed block size	Mode decision	JPEG-LS
Web 1	2.682	1.638	0.377	0.785	0.333
Web 2	2.464	1.614	0.378	0.771	0.284
Web 3	2.348	1.561	0.354	0.755	0.266
Web 4	2.284	1.644	0.367	0.791	0.370
Average	2.444	1.614	0.369	0.776	0.313

타낸 것이다. QP가 1일 때, 위에서 사용한 10개 프레임에 대해 6번 반복하여 총 60 프레임의 평균 시간을 측정하였다. 표 4, 5를 보면 템플릿 매칭 고속화를 적용함으로써 full search에 비해 30% 이상의 시간을 단축시킬 수 있었고, 모드 결정을 통해 추가로 절반 이상의 시간을 더 단축시킬 수 있었다. 표 4, 5의 네 번째 column은 항상 템플릿 매칭을 포함한 ICP를 사용하고, 가변 블록 크기가 아닌 8×1 블록만을 고정하여 사용한 예측 방법이다. 이 경우, JPEG-LS와 유사한 연산 속도를 보임을 알 수 있다.

V. 결론

본 논문은 낮은 복잡도를 가진 이미지 코더의 예측 기법을 제안하여 기존 기법과 비슷한 PSNR을 유지하면서 압축율을 향상시키는 기법을 제안하였다. RGB 간의 상관관계를 고려하여 생성한 선형 모델을 적용하면, 기존 기법보다 평균 약 20%~30%의 성능 향상을 보인다. 또한, 다양한 블록 크기를 이용해 불필요한 예측 모드 비트량을 줄이면, 기존 기법 대비 평균 약 25~50%의 성능 향상을 볼 수 있다. 마지막으로 템플릿을 이용해 성능이 좋지 않은 복잡한 텍스처 영역의 G 성분 예측 성능을 개선시켜 압축율을 향상시킬 수 있다. 실험 결과를 통해 제안 기법이 자연 영상의 경우 기존 기법 대비 약 30%의 성능 향상을 보이고, CG 영상의 경우 60~70%의 성능 향상을 보임을 알 수 있다. 모드 결정을 적용하여 좀 더 낮은 연산량을 갖는 이미지 코더를 구현할 수 있다. 제안 기법은 video CODEC같은 방대한 메모리 대역폭을 요구하는 멀티미디어 SoC에서 CODEC과 외부 메모리 간 인터페이스 솔루션으로 사용될 수 있다.

참고 문헌 (References)

[1] M. Sugawara, M. Kanazawa, K. Mitani, H. Shimamoto, T. Yamashita and F. Okano, "Ultrahigh-Definition Video System with 4000

- Scanning Lines," SMPTE Motion Imaging Journal, vol. 112, no. 10/11, pp. 339-346, Oct. 2003.
- [2] M. Sugawara, K. Masaoka, M. Emoto, Y. Matsuo and Y. Nojiri, "Research on Human Factors in Ultrahigh-Definition Television (UHDTV) to Determine Its Specifications," SMPTE Motion Imaging Journal, vol. 117, no. 3, pp. 23-29, Apr. 2008.
- [3] C. Chrysafis and A. Ortega, "Line-Based, Reduced Memory, Wavelet Image Compression," IEEE Trans. Image Processing, vol. 9, no. 3, pp. 378-389, Mar. 2000.
- [4] E. Ordentlich, D. Taubman, M. Weinberger and G. Seroussi, "Memory Efficient Scalable Line-based Image Coding," Proc. IEEE Data Compression Conf., pp. 218-227, Mar. 1999.
- [5] N. Memon and X. Wu, "Recent Developments in Context-Based Predictive Techniques for Lossless Image Compression," Comput. J., vol. 40, no. 2/3, pp. 127-136, May 1997.
- [6] M. J. Weinberger, G. Seroussi and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Trans. Image Processing, vol. 9, no. 8, pp. 1309-1324, Aug. 2000.
- [7] ISO/IEC, "Information Technology - Lossless and Near-lossless Compression of Continuous-Tone Still Images - Baseline," ITU-T Recommendation T.87, 14495-1, 1998.
- [8] X. Wu and N. Memon, "Context-Based, Adaptive, Lossless Image Coding," IEEE Trans. Communications, vol. 45, no. 4, pp. 437-444, Apr. 1997.
- [9] A. Avramovic and B. Reljin, "Gradient Edge Detection Predictor for Image Lossless Compression," Proceedings of ELMAR, pp. 131-134, Sep. 2010.
- [10] J. Kim and C.M. Kyung, "A Lossless Embedded Compression Using Significant Bit Truncation for HD Video Coding," IEEE Trans. Circuits Syst. Video Technol., vol. 20, No.6, pp. 848-860, Jun. 2010.
- [11] E. Wige, G. Yammine, P. Amon, A. Hutter and A. Kaup, "Pixel-Based Averaging Predictor for HEVC Lossless Coding," in Proc. ICIP, pp. 1806-1810, Sep. 2013.
- [12] Y. G. Lee, B. C. Song, N. H. Kim, T. H. Kim and W. H. Joo, "Low-complexity near-lossless image coder for efficient bus traffic in very large size multimedia SoC," in Proc. ICIP, pp. 2329-2332, Nov. 2009.
- [13] Y. G. Lee and B. C. Song, "An Intra-Frame Rate Control Algorithm for Ultralow Delay H.264/Advanced Video Coding (AVC)," IEEE Trans. Circuits Syst. Video Technol., vol. 19, pp. 747-752, May 2009.
- [14] B. C. Song, Y. G. Lee and N. H. Kim, "Block Adaptive Inter-Color Compensation Algorithm for RGB 4:4:4 Video Coding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, pp. 1447-1451, Oct. 2008.
- [15] B. C. Song, M. J. Kim and J. B. Ra, "A fast multiresolution feature matching algorithm for exhaustive search in large image databases," IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 5, pp. 673-678, May 2001.

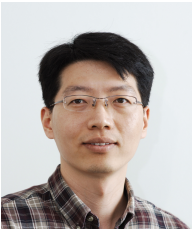
저 자 소 개

손 지 덕



- 2012년 : 인하대학교 전자공학과 학사 졸업
- 2014년 : 인하대학교 전자공학과 석사 졸업
- 주관심분야 : 영상 신호처리

송 병 철



- 1994년 2월 : 한국과학기술원 전기 및 전자공학과 졸업 (공학사)
- 1996년 2월 : 한국과학기술원 전기 및 전자공학과 졸업 (공학석사)
- 2001년 2월 : 한국과학기술원 전기 및 전자공학과 졸업 (공학박사)
- 2001년 3월 ~ 2008년 2월 : 삼성전자 디지털미디어연구소 책임연구원
- 2008년 3월 ~ 현재 : 인하대학교 전자공학과 부교수
- 주관심분야 : 영상 신호처리, 영상 시스템/SoC