

템플릿 매칭 기반 적응적 갱신-예측 구조 웨이블릿 변환 기법

*박상재 **김성제 ***최윤식

연세대학교 전기전자공학부

*psj8308@yonsei.ac.kr

Adaptive Update-Predict Structure Based on Template Matching Method in Wavelet Transform

*Park, Sang-Jae **Kim, Sung-Jei ***Choe, Yoon-sik

Department of Electrical and Electronics Engineering Yonsei University

요약

웨이블릿 변환(Wavelet Transform)은 영상압축에 효율적인 방법으로 알려져 있으며 lifting scheme을 이용해서 쉽게 구현이 가능하다. 가장 널리 쓰이는 방법으로는 Daubechies 5/3 필터가 있고, 이를 바탕으로 하여 영상의 기하학적인 특성을 이용한 적응적 예측 방법이 많이 소개되었다. 본 논문에서는 적응적 예측을 위해 템플릿 매칭을 적용한 새로운 알고리즘을 제안하였다.

1. 서론

웨이블릿 변환은 주어진 신호를 다른 범위의 공간주파수들을 갖는 신호들의 집합(subband)으로 변환하는 방법이다. 이 방법은 DCT를 이용한 블록기반 압축 기술에 비해 블록화 현상이 발생하지 않으며 높은 압축률과 계층성(scalability)을 가진다. 이러한 장점들로 인해 웨이블릿 변환은 영상압축에 효율적인 변환방법으로 알려져 있으며 정지 영상 압축 표준인 JPEG2000에 채택되기도 하였다.[1][2]

Lifting scheme은 웨이블릿 변환을 설계함에 있어서 가장 널리 쓰이는 방법이다. 이 방법은 연산량이 적고, 변환 및 역변환 알고리즘이 단순하고, 완전복원이 가능하다는 특징을 가지고 있다.[3] Lifting scheme을 구현하기 위해서는 예측과 갱신 과정에서 적절한 필터 탭 계수를 설정하는 작업이 필요한데 일반적으로 가장 널리 쓰이는 방식은 Daubechies가 제안한 5/3 필터탭, 9/7 필터탭 등이다.[4] 이 방법은 주어진 영상을 수직 또는 수평의 한 방향으로만 필터링을 수행한다. 때문에 수직, 수평 방향 이외의 다른 방향을 고려하여 적응적 예측을 수행한다면 좀 더 좋은 압축 효율을 얻을 수 있다.

일반적인 lifting scheme은 분할(split), 예측(prediction), 갱신(update)의 순서로 이루어진다. 하지만 이 순서를 따를 경우 적응적 예측과정에서 방향선택에 관한 추가 정보를 보내야 한다. 만약 구조를 '분할-예측-갱신'이 아닌 '분할-갱신-예측'의 순서로 바꾼다면 예측 과정에서 갱신영역에 관한 정보를 이용할 수 있기 때문에 방향선택에 관한 추가 정보를 보낼 필요가 없다. 때문에 본 논문에서는 갱신-예측 필터를 사용해서 적응적 예측을 수행하는 방법을 제시할 것이다.

적응적 예측을 수행한 기존의 많은 작업들 가운데 Gerek과 Cetin은 영상의 기울기(마주하는 픽셀값들의 차)정보를 이용하여 방향성을 고

려한 적응적 예측방법을 제안하였다.[6] 하지만 이 방법에는 부호화와 복호화 사이에 방향선택에 대한 비동기화 문제가 발생하게 된다. 본 논문에서는 이러한 비동기화 문제를 해결하기 위해 부호화기 내부에 복호화기를 포함한 형태의 구조를 제안하였다. 이렇게 되면 부호화 과정에서 복원 영상을 얻을 수 있기 때문에 부/복호화과정에서의 비동기화 문제를 해결할 수 있고, 그 결과 좀 더 높은 압축률을 가질 수 있게 된다. 그리고 적응적 방향 예측을 위해서 템플릿 매칭(template matching) 방법을 사용하였다. 템플릿 매칭 방법은 현재 픽셀값을 부호화하기 위해 이전까지 복호화 된 픽셀값 들을 조사하여 현재 지점과 가장 유사한 지점을 찾는 방법이다. 이렇게 되면 부/복호화 과정에서 현재 부호화할 픽셀과 가장 유사한 픽셀값을 이용할 수 있기 때문에 적응적 예측에 있어서 강점을 가지게 된다.

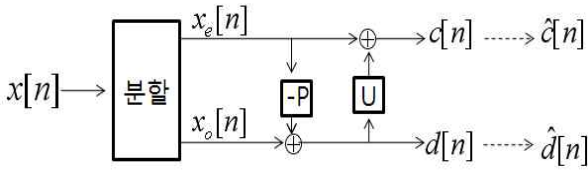
본 논문의 진행 순서는 다음과 같다. 먼저 2장에서는 Gerek과 Cetin이 제안한 갱신-예측 구조에 기반 한 적응적 예측 방법에 대해서 설명하고, 3장에서는 템플릿 매칭을 이용한 본 논문의 제안 알고리즘에 대해 설명한다. 그리고 4장에서는 Gerek과 Cetin의 알고리즘과 제안 알고리즘을 비교한 실험결과를 소개하고 5장에서는 결론을 제시 한다.

2. 적응적 예측

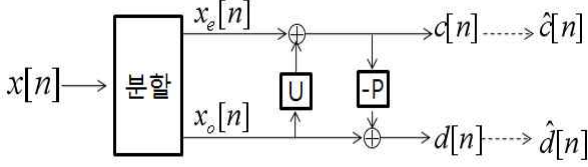
가. 갱신-예측 구조

일반적인 lifting scheme은 '분할(split) - 예측(predict) - 갱신(update)'의 흐름을 따른다. 전체적인 과정은 다음과 같다.

원본 영상의 픽셀값을 $x[n]$ 이라고 하고, $x[n]$ 을 기수(odd)와 우수



<그림 1> 예측-갱신 구조



<그림 2> 갱신-예측 구조

(even)에 따라 (1)식과 같이 $x_e[n, m], x_o[n, m]$ 으로 나눈다.

$$x_e[n, m] = x[2n, m], x_o[n, m] = x[2n + 1, m] \quad (1)$$

그리고 결과인 $x_e[n], x_o[n]$ 을 이용해 예측(고주파 성분에 해당)과 갱신(저주파 성분에 해당)과정을 수행하게 된다.

영상은 2차원 정보이기 때문에 실제로 필터링을 할 때는 수평방향으로 필터링을 수행한 후 수직방향으로 또 한 번 필터링을 수행해야 한다. 하지만 편의상 앞으로의 예에서는 수평 방향의 필터링에 대해서만 고려하여 설명하도록 하겠다. 널리 쓰이는 Daubechies 5/3 필터의 경우 예측 영역인 $d[n, m]$ 과 갱신 영역인 $c[n, m]$ 값을 $x_e[n, m]$ 과 $x_o[n, m]$ 을 통해 다음과 같이 구할 수 있다.

$$d[n, m] = x_o[n, m] - \frac{x_e[n, m] + x_e[n + 1, m]}{2} \quad (2)$$

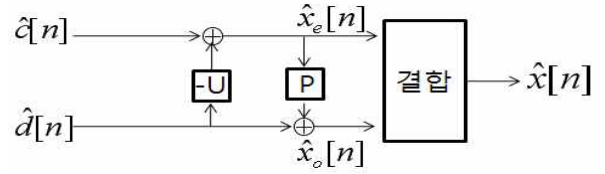
$$c[n, m] = x_e[n, m] + \frac{d[n, m] + d[n + 1, m]}{4} \quad (3)$$

이 과정이 위의 <그림 1>에 해당하는 과정이다. 그리고 아래 (4),(5)와 같은 역과정을 통해 쉽게 복원이 가능하다.

$$\hat{x}_e[n, m] = \hat{c}[n, m] - \frac{\hat{d}[n, m] + \hat{d}[n + 1, m]}{4} \quad (4)$$

$$\hat{x}_o[n, m] = \hat{d}[n, m] + \frac{\hat{x}_e[n, m] + \hat{x}_e[n + 1, m]}{2} \quad (5)$$

하지만 이 구조에서는 예측과정 후 갱신과정을 수행하기 때문에 적응적 예측과정에 어려움이 있다. 왜냐하면 적응적 예측을 수행하기 위해서는 참고로 할 영상이 필요한데, 예측 과정을 먼저 수행하게 되면 (즉, $d[n, m]$ 값을 먼저 구하게 되면) 이 값을 구하는 과정에서 이용할 수 있는 값들은 x_e 와 x_o 값들이다. 하지만 부호화 과정에서는 이 값들을 이용할 수 없기 때문에 만약 이 값들을 통해 적응적 예측을 수행한다면 적응적 예측에 관한 추가 정보를 전송해줘야 하기 때문에 비트율이 늘어나게 된다. 이런 문제점을 해결하기 위해 본 논문에서는 예측-갱신 방식 대신 갱신-예측 방식을 이용한다. 이렇게 되면 예측 과정에서 앞서 구한 갱신 영역을 이용할 수 있기 때문에 추가



<그림 3> 역 갱신-예측 구조

정보 전송 없이 적응적 예측이 가능해진다. <그림2>은 갱신-예측 과정을 나타내고 있으며 <그림3>는 복원과정에서 사용하게 되는 역 갱신-예측구조를 나타내고 있다.

위의 그림에서 $\hat{c}[n], \hat{d}[n]$ 은 $c[n], d[n]$ 을 양자화 했음을 의미 한다. 때문에 <그림3>에서 보듯이, 복원 과정에서 이용하는 값은 $c[n], d[n]$ 이 아닌 $\hat{c}[n], \hat{d}[n]$ 을 이용하게 된다.

나. 기울기 정보를 이용한 적응적 예측

갱신-예측 방법을 사용하게 되면 $c[n, m]$ 이 먼저, $d[n, m]$ 이 나중에 생성된다. 이렇게 되면 $d[n, m]$ 을 얻기 위해 적응적 예측을 할 때 참조 영상으로 이미 생성된 $c[n, m]$ 을 이용할 수 있게 된다.

$c[n, m], d[n, m]$ 값을 얻기 위해 본 논문에서 사용한 필터는 다음과 같다.

$$c[n, m] = \frac{x_e[n, m] + x_o[n, m]}{2} \quad (6)$$

$$d[n, m] = x_o[n, m] - \left(-\frac{c[n-1, m]}{8} + c[n, m] + \frac{c[n+1, m]}{8} \right) \quad (7)$$

본 논문에서 언급하는 '적응적 예측'이라는 용어는 필터링과정에서 수직, 수평 방향만이 아닌 45도, 135도의 대각선 방향도 고려하여 필터링한다는 의미이다.

이와 같은 방식의 적응적 예측에 있어서 중요한 점은 3가지 방향 중 최적의 방향을 선택하는 알고리즘을 찾아야 한다는 것이다. 만약 방향 선택을 제대로 하지 못한다면 일반적인 수직, 수평 방향의 필터링 보다 좋지 않은 결과를 보일 수도 있다.

3방향에 대한 필터링 $P_i[n, m]$ 을 다음과 같이 정의하자.

$$P_1[n, m] = \left(-\frac{c[n-1, m]}{8} + c[n, m] + \frac{c[n+1, m]}{8} \right) \quad (8)$$

$$P_2[n, m] = \left(-\frac{c[n-1, m-1]}{8} + c[n, m] + \frac{c[n+1, m+1]}{8} \right) \quad (9)$$

$$P_3[n, m] = \left(-\frac{c[n-1, m+1]}{8} + c[n, m] + \frac{c[n+1, m-1]}{8} \right) \quad (10)$$

3가지 방향을 모두 고려했을 때 최적의 방향은 아래의 (11)식에 의해서 구할 수 있다.

$$d[n, m] = \min(x_o[n, m] - P_i[n, m]) \quad (i=1,2,3) \quad (11)$$

$d[n, m]$ 값이 작은 값에 모여 있을수록 압축 효율이 좋아지기 때문에

최소의 $d[n,m]$ 을 가지는 방향이 최적의 방향으로 결정된다. 하지만 모든 지점에서 최적의 방향을 선택하게 되면 복호화 과정을 위해 부호화 과정에서 선택된 방향 정보를 전송해야 한다는 문제점이 있다. 때문에 부/복호화 과정에서 방향 선택에 관한 추가 정보를 전송하지 않으면서 이상적인 방향에 가까운 방향선택 알고리즘을 찾는다면 성능 향상을 가져올 수 있을 것이라는 사실을 알 수 있다.

Gerek과 Cetin은 이러한 방향선택 알고리즘으로 기울기 값(마주보는 픽셀의 값의 차)을 이용하였다. 마주보는 픽셀값의 차이가 작다면 그 방향을 예지로 생각할 수 있고, 그 방향으로 필터링을 수행 한다면 좀 더 좋은 예측값을 얻을 수 있다고 생각한 것이다. 즉, 아래와 같은 3방향의 기울기 값

$$\left| \frac{c[n-1,m]}{8} - \frac{c[n+1,m]}{8} \right| \quad (12)$$

$$\left| \frac{c[n-1,m-1]}{8} - \frac{c[n+1,m+1]}{8} \right| \quad (13)$$

$$\left| \frac{c[n-1,m+1]}{8} - \frac{c[n+1,m-1]}{8} \right| \quad (14)$$

중에서 최소값을 가지는 방향을 선택한 다음 그 방향으로 (11)식을 적용하여 $d[n,m]$ 을 구하는 것이다.

예를 들어 (12),(13),(14)의 값 중에서 (13)이 최소값으로 결정되었다면(135도 방향), $d[n,m]$ 은 $x_o[n,m] - P_2[n,m]$ 즉, 아래의 (15)식을 통해 구하게 되는 것이다.

$$d[n,m] = x_o[n,m] - \left(-\frac{c[n-1,m-1]}{8} + c[n,m] + \frac{c[n+1,m+1]}{8} \right) \quad (15)$$

하지만 이 방법에서는 복호화 과정에 $c[n,m]$ 값을 참조 할 수 없다는 문제점이 있다. 왜냐하면 부호화 과정 후 전송되는 값은 $c[n,m]$ 이 아닌 양자화 된 $\hat{c}[n,m]$ 을 전송하기 때문이다. 때문에 복호화과정에서는 $\hat{c}[n,m]$ 을 참조하게 되어 부호화과정과 복호화과정 사이에 방향선택에 대한 비동기화 문제가 발생하게 된다. 또한 기울기 값이 작지만 예지방향이 아닌 부분도 존재하기 때문에 기울기 정보를 이용하는 방법 외에 다른 방법이 필요하다. 본 논문에서는 방향선택에 관한 알고리즘으로 템플릿 매칭을 제안 하였다.

3. 제안 알고리즘

본 논문에서는 적응적 예측을 수행하기 위해 템플릿 매칭을 이용하여 새로운 방향선택 알고리즘을 제안한다. 그리고 템플릿 매칭을 위해서 복호화기를 포함한 부호화기 구조를 설계 하였다. 이 구조를 사용하게 되면 앞서 언급한 비동기화 문제역시 해결할 수 있다.

가. 복호화기를 포함한 부호화기

복호화기를 포함한 부호화기를 만들기 위해서는 먼저 비동기화 문제를 해결해야 한다. 앞서 언급한 것처럼 적응적 예측을 수행하는 과정에서 양자화로 인해 부/복호화과정의 방향선택이 달라지면서 오차가 발생하게 된다. 이 문제를 해결하는 방법은 부/복호화 과정에서 참조하는 값을 똑같은 것으로 선택하는 것이다. 만일 부호화 과정에서 참조하는 값이 $c[n,m]$ 이 아니라 양자화 된 $\hat{c}[n,m]$ 을 이용한다면 복호화 과정에서도 같은 값을 참조할 수 있기 때문에 똑같은 방향을 선택할 수 있고 비동기화 문제를 해결할 수 있다. 주의하여야 할 점은 부호화 과정에서 '방향선택'은 $\hat{c}[n,m]$ 의 정보를 이용해 결정하지만 실제로 $d[n,m]$ 값을 구할 때는 앞서 선택된 방향대로 $c[n,m]$ 의 값을 이용해야 한다는 점이다. 복호화기를 포함한 부호화기를 사용하면 부호화 과정에서 부호화를 수행한 지점까지의 복원된 영상을 얻을 수 있다.

나. 템플릿 매칭

복호화기를 포함한 부호화기를 설계한 이유는 템플릿 매칭을 이용하기 위해서이다. 템플릿 매칭은 예측하고자 하는 픽셀 주변의 정보를 이용하여 이전의 복호화 된 영역에서 가장 비슷한 부분을 찾아 예측하는 방법이다.

이전 논문에서는 템플릿 매칭을 이용하여 현재 부호화할 $x_o[n,m]$ 과 가장 비슷하다고 계산되는 픽셀을 템플릿 매칭을 통해 찾았다.[7] 이때 템플릿 매칭을 통해 찾은 픽셀의 좌표를 p,q라 하고 $T(\cdot)$ 을 템플릿 매칭을 수행하는 과정으로 정의 하면 다음과 같이 나타낼 수 있다.

$$\hat{x}_o[p,q] = T(x_o[n,m]) \quad (16)$$

그리고 방향 선택을 위한 식은 다음과 같다.

$$|\hat{x}_o[p,q] - \hat{P}_i[p,q]| \quad (i=1,2,3) \quad (17)$$

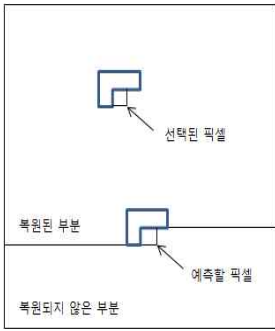
즉 위의 식에서 최소의 값을 가지는 방향을 선택한 후 그 방향에 따라 $d[n,m]=x_o[n,m] - P_i[n,m]$ 값을 계산하였다.

본 논문에서는 템플릿 매칭을 복원 영상에 적용하는 것이 아니라 갱신영역에 적용하는 방법을 통해 성능 향상을 이루었다.

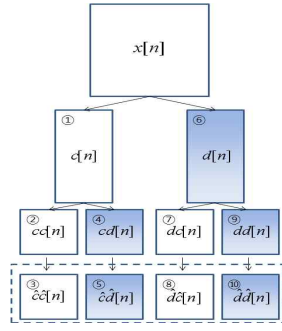
제안하는 알고리즘의 과정은 다음과 같다.

먼저 부호화할 픽셀 $x_o[n,m]$ 의 위치에 해당하는 $\hat{c}[n,m]$ 의 주위 4개의 픽셀과 가장 유사한 값을 가지는 영역을 이전의 갱신 영역에서 찾는다. 여기서 $c[n,m]$ 이 아닌 $\hat{c}[n,m]$ 을 참조하는 이유는 복호화 과정에서 동기화를 이루기 위해서이다. 즉, (16)식과는 다르게 $\hat{c}[p,q] = T(\hat{c}[n,m])$ 이 된다. 그리고 갱신영역에서 p,q 지점에 해당하는 x_o 영역에서의 지점을 찾는다. 그리고 (17)번의 식을 이용해서 최소값을 갖는 방향을 선택한 다음, 그 방향에 따라 $d[n,m]=x_o[n,m] - P_i[n,m]$ 값을 계산한다.

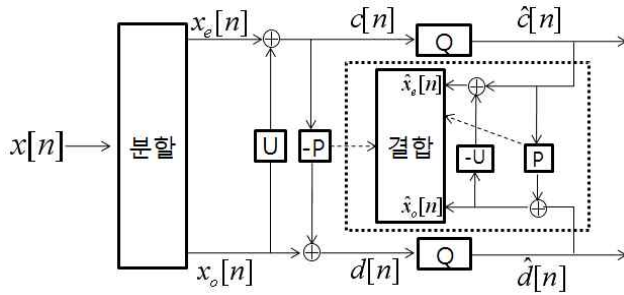
영상은 2차원 정보이기 때문에 수직, 수평방향으로 한 번씩의 필터링을 수행하여야한다 <그림 5>는 제안 알고리즘에 따라 2번의 필터링



<그림 4> 템플릿 매칭



<그림 5> 제안알고리즘 구조



<그림 6> 제안 알고리즘 블록도

과 양자화에 의해 생성되는 4개 대역의 생성순서를 나타낸다.

그리고 한번 필터링을 수행할 때의 제안하는 알고리즘의 블록도는 <그림 6>와 같다. <그림 6>에서 예측과 역예측 과정에서 점선으로 표현한 부분은 템플릿 매칭을 통해 얻어진 값을 참조해서 구한다는 의미이다.

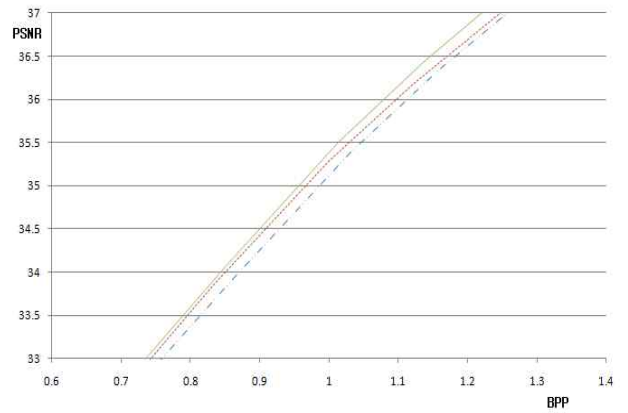
4 실험 결과

제안 알고리즘은 barbara영상을 사용하여 실험하였다. 또한 템플릿 매칭에서 정확도를 높이기 위해 템플릿 매칭 하는 지점을 5개를 선택하여 각각의 지점에 해당하는 $\hat{c}[p,q]$ 값들과 $\hat{x}_o[p,q]$ 의 평균값을 계산하여 방향을 선택하였다. 또한 템플릿 매칭을 수행할 탐색 영역은 20으로 설정하였다. 적응적 예측을 수행할 수 있는 부분은 <그림 5>의 4,6,9번 과정이다. 본 논문에서는 6번 과정에서만 템플릿 매칭을 이용한 적응적 예측을 수행하였고, 나머지 부분에 대해서는 적응적 예측을 수행하지 않았다. 실험결과는 <그림 7>과 같다.

가장 아래의 그래프는 Gerek과 Cetin이 제안한 방법이고 중간 그래프는 [7]논문의 결과이고 가장 위의 그래프는 제안 알고리즘의 그래프이다. Gerek과 Cetin의 알고리즘에 비해 0.2 ~ 0.3 dB정도 더 좋은 성능을 보이는 것을 확인 할 수 있다.

5. 결론

본 논문에서는 갱신-예측 구조에서 템플릿 매칭을 사용하여 적응적



<그림 7> 실험결과

예측을 수행하는 알고리즘을 제안하였다. 그리고 템플릿 매칭을 이용하기 위해서 복호화기를 포함한 부호화기 구조를 설계하였다. 그리고 템플릿 매칭을 복원 영상이 아닌 갱신 영역에 수행하는 알고리즘을 제안하였다. 그 결과 픽셀의 기울기 정보를 이용한 Gerek과 Cetin의 알고리즘에 비해 최대 약 0.3dB 정도의 성능향상이 있음을 확인 할 수 있다.

[참 고 문 헌]

- [1] M.Vetterli and J.Kovacevic "Wavelets and Subband Coding", Prentice Hall, 1995
- [2] D.Taubman and M. Marcellin, "JPEG2000 : Image Compression Fundamentals, Standards, and Practice". Norwell, MA:Kluwer, 2001.
- [3] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proc. SPIE Wavelet Applications in Signal and Image Processing III*, vol. 2569, pp. 68 - 79, 1995.
- [4] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247 - 269, 1998.
- [5] O. Gerek and E. Cetin, "Linear/nonlinear adaptive polyphase subband decomposition structures for image compression," in *IEE E Int. Conf.Acoustics, Speech, Signal Processing*, vol. 3, pp. 1345 - 1348, 1998.
- [6] O. Gerek and A. E. Cetin, "A 2-D orientation-adaptive prediction filter in lifting structures for image coding," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 106-111, 2006.
- [7] 박상재, 김성제, 최윤식 "템플릿 매칭을 이용한 적응적 갱신-예측 구조 웨이블릿", 정보및 제어 학회 2009년논문집, pp319-320, 2009년