
무손실 영상 압축을 위한 변형된 정수 변환들에 대한 성능 분석

김희경* · 유훈**

Performance analysis on modified integer transforms for lossless image compression

Hui-Gyeong Kim* · Hoon Yoo**

이 논문은 2011년도 상명대학교 연구비를 지원받았음

요 약

본 논문에서는 무손실 영상 압축에서 쓰이는 변형된 정수 변환들을 설명하고 이를 2차원으로 확장한 것에 대한 성능 분석을 제시한다. 변형된 정수 변환들은 2차원으로 확장하는 방법에 따라서 그 성능 및 복잡도가 다른 면이 있다. 따라서 본 논문에서는 분리 가능한 형태로 확장한 변형 H.264 정수 변환 및 리프팅 구조를 이용한 가역 정수 변환과 분리 가능하지 않은 형태로 확장한 JPEG XR의 PCT변환에 대한 성능 및 복잡도를 비교 분석하여 제시하고, 이에 관련된 실험 결과를 제공한다. 실험 결과는 리프팅 구조를 이용하여 변형된 가역 정수 변환을 분리 가능한 형태로 확장하는 방법이 압축 효율면에서 가장 우수함을 보여준다.

ABSTRACT

In this paper, we introduce modified integer transforms for lossless image compression and evaluate their performances for two-dimensional transforms. The two-dimensional extensions of the modified integer transforms show different performances in terms of coding efficiency and computational complexity. Thus, we measure performances for two-dimensional separable transforms and a two-dimensional non-separable transform. The separable modified integer transform used in H.264, the modified integer transform using the lifting scheme, and the non-separable transform in JPEG XR are evaluated in this paper. Also, experiments and their results are given. The experimental results indicate that the modified integer transform using the lifting scheme shows the best performance in terms of compression efficiency.

키워드

가역 정수 변환, 무손실 압축, JPEG XR, PCT, H.264/AVC

Key word

Reversible integer transform, lossless compression, JPEG XR, PCT, H.264/AVC

* 준회원 : 상명대학교 디지털미디어학부

접수일자 : 2012. 04. 09

** 종신회원 : 상명대학교 디지털미디어학부 (교신저자, hunie@smu.ac.kr)

심사완료일자 : 2012. 04. 27

I. 서 론

영상 압축의 두 가지 대표적인 변환에는 블록 변환과 웨이블릿 변환이 있다. 블록 기반의 변환은 입력 영상을 오버랩 되지 않는 작은 블록 단위로 나누어서 처리하는 기법이다[1]. 대표적인 예로써 H.264/ Advanced Video Coding(AVC)[2]와 JPEG extended range(JPEG XR)[3-5] 압축 방법을 들 수 있다.

이 때 실수 변환이 아닌 정수 변환을 하게 되면 곱셈 연산을 덧셈 연산과 쉬프트 연산으로 대체 할 수 있어 계산이 단순해져 속도가 빨라지고 메모리를 줄여주는 효과를 가져 온다. 이는 비용을 획기적으로 감소시킬 수 있을 뿐만 아니라 이러한 정수 변환은 무손실 압축이 가능한 가역 변환을 제공 한다는 장점을 가지고 있다. 따라서 이전 비디오 코딩에서는 8×8 Discrete Cosine Transform (DCT)를 사용하였지만, 이러한 실수 변환의 문제점을 극복하고 블록 크기로 인한 링 현상을 줄이기 위해 최근 4×4 변환 블록을 단위로 한 정수 변환을 많이 사용하고 있다.

기존의 H.264/AVC 표준에서 사용되는 변환은 결과의 비트 수가 크고 계산량이 많아 압축 성능 면에서 효율적이지 못하다. 이는 결과 비트 수가 커져서 변환 계수 범위가 커지면 엔트로피가 증가할 뿐만 아니라, 무손실 압축에서는 변환 계수의 범위가 압축 성능에 상당히 민감하게 작용하기 때문이다[7]. 따라서 앞선 연구[8]에서는 이러한 효율성 문제를 보완하기 위한 방법으로 H.264/AVC를 리프팅을 기반으로 무손실 영상 압축에서 적합한 형태로 변형하여 제시하고 그 성능의 우수함을 보인 바 있다. 하지만, 이 연구는 1차원에 제한된 것으로 2차원에 대한 연구가 필요하다.

본 논문에서는 무손실 영상 압축에서 쓰이는 기존의 1차원 기반으로 제시된 변형된 정수 변환 방법들을 살펴보고 이를 2차원으로 확장한다. 이 때 1차원 변환을 반복 사용해서 분리 가능(separable)한 형태로 구현하여 확장한 방법과 2차원적으로 분리 가능하지 않은(non-separable) 방법으로 설계된 JPEG XR의 Photo Core Transform(PCT)와의 성능 및 복잡도 비교 실험을 통하여 이들을 평가 분석한다.

본 논문의 구성은 다음과 같다. 2장에서 무손실 영상 압축을 위한 정수 변환들을 살펴보고 3장에서 성능 실

험에 대한 결과를 제시한다. 4장에서는 그 결과를 분석하고 5장에서 결론으로 끝을 맺는다.

II. 무손실 영상 압축을 위한 정수 변환들

2.1. S변환

2점 변환의 대표적인 연산으로 나비(butterfly) 연산이 있다. H.264/AVC 정수 변환의 고속 알고리즘은 나비 연산으로 구성되어 있는데 이 연산 중 하나가 2점 하다마드(Hadamard) 변환과 동일하다[8]. 이러한 2점 하다마드 변환을 리프팅 구조를 사용하여 효율적으로 구현한 것이 S변환(S-transform)[4]으로 그림 1에서 제시되었다.

여기서 변환계수 y_1 의 계수 범위(dynamic range)는 입력계수 x_0, x_1 보다 1비트 더 크다. 하지만 y_0 는 입력계수와 동일한 범위를 가지므로, y_0 의 계수 범위가 1비트 증가하는 하다마드 변환보다 범위가 작아짐을 확인할 수 있다. 계수가 1비트 작아지면 계수 범위가 반으로 감소하기 때문에 이는 무손실 압축에서 상당히 효율적임을 알 수 있다.

이 변환을 2차원으로 확장하기 위해서 2×2 블록의 4개의 픽셀들에 대해서 수직, 수평방향으로 나누어 2점 S변환을 분리 가능(separable)한 형태로 4번 반복 구현한다.

2×2 하다마드 변환은 픽셀들을 분리 가능하지 않은 형태로 변환한 것으로 이것과 2×2 블록에 대해 분리 가능하도록 변환한 S변환과의 변환 결과 비트를 비교해 보면, 두 변환은 각 픽셀에 대해서 비트량 할당이 다를 뿐 전체 비트량이 같아 결과적으로 동일한 변환임을 알 수 있다. 이와 같이 2×2 블록에서 두 변환 방식이 일치한다는 것을 고려한다면, 4×4 블록 단위의 무손실 정수 변환에서도 두 변환 방식이 성능의 유사함을 보이는 것은 당연하다.

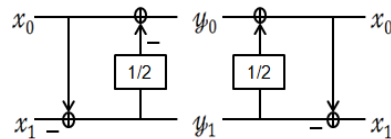


그림 1. S변환에 대한 리프팅 구현
Fig. 1 Lifting implementation of S-transform

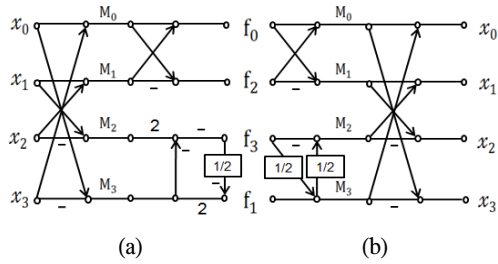


그림 2. 변형 H.264 정수 변환 (a) 변환 (b) 역변환
Fig. 2 Modified H.264 integer transform (a) forward transform (b) inverse transform

2.2. 변형 H.264/AVC

기존의 H.264/AVC 표준에서 사용되는 정수 변환은 무손실 압축 성능을 충분히 고려하지 않은 방법으로 성능 면에서 문제를 가지고 있으므로 적합한 형태로의 변형이 요구된다. 그림 2에서 변형된 H.264/AVC 정수 변환의 흐름도가 제시되었다. 이 변환은 H.264/AVC를 가역(reversible) 정수 변환으로 변형시킨 것으로 변환 후 다시 역변환 한 결과는 입력 변수의 정확히 4배가 나오는데 곱셈 연산을 사용하지 않고 쉬프트 연산만 사용하여 원본 값으로의 복원이 가능하다. 따라서 무손실을 보장할 뿐만 아니라 변환 계수의 범위를 줄여 그 효율성을 높였다[7]. 1차원에서의 이 변형된 정수 변환을 2차원으로 확장하기 위하여 S변환을 확장한 것과 같은 방식으로 수직, 수평 방향에 대해 분리 가능한 방식으로 4×4 블록에 대해서 반복해서 구현한다. 이 때 4점 변환은 가로 혹은 세로 방향의 연속적인 4개의 픽셀이 입력이 된다. 변형 H.264 정수 변환이 1차원의 압축에서 효율적이지 못했던 점에 근거하여[8], 2차원에서도 성능 면에서 다소 문제점이 존재한다는 것을 알 수 있다.

2.3. JPEG XR에서의 정수 변환

JPEG XR에서 쓰이는 기본 변환 중의 하나는 Photo Core Transform(PCT)이다[4]. 이 변환은 입력 영상을 4×4 블록 단위로 나눈 뒤 각 블록에 대해서 2차원적으로 처리하는 방식이다. 변환의 입력은 연속적인 선형의 4개의 픽셀이 아닌 2×2 블록 모양의 픽셀로, 이 4개의 픽셀을 1차원 데이터처럼 나열하여 변환해 주는 방식을 사용한다. 변환 전후의 계수 배치 순서는 그림 3(a)에 나타난 것과 같다[9]. 그림 3(b)에서 알 수 있듯이 PCT 변환은 두 단계로 이루어져 있다.

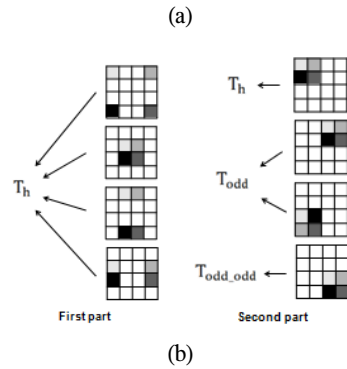
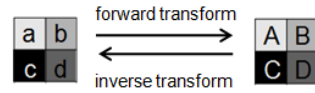


그림 3. PCT 변환 (a) 변환과 역변환 연산 (b) 변환 계수의 두 단계 과정
Fig. 3 PCT transform (a) forward transform and inverse transform operation (b) two processes of PCT

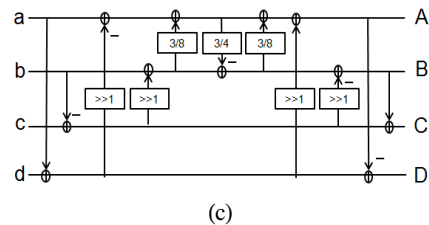
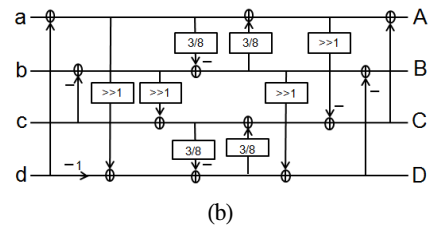
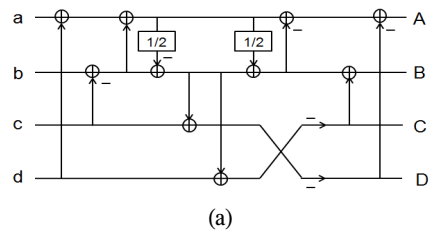


그림 4. PCT의 기본 연산 (a) T_h (b) T_{odd} (c) T_{odd_odd}
Fig. 4 Basic operations of PCT (a) T_h (b) T_{odd} (c) T_{odd_odd}

첫 번째 단계에서는 영상의 각 4×4 블록을 4개의 변환 계수에 대해 4번씩 T_h 연산을 하는데 여기서 T_h 연산은 2차원 2×2 하다마드 변환을 의미한다. 두 번째 단계에서는 $T_h, T_{odd}, T_{odd_odd}$ 연산을 순서대로 각각 1회, 2회, 1회씩 수행 한다. 각 변환 연산에 해당하는 신호 흐름도는 그림 4에 제시된 것과 같다. 여기서 JPEG XR의 모든 변환 연산은 리프팅 구조를 사용하여 구현되었다[4].

PCT 변환이 하다마드 변환을 기반으로 한다는 점을 근거로 성능 면에서 분리 가능하지 않은 2차원 변환과 유사함을 가진다는 것을 알 수 있다. 따라서 PCT 변환이 분리 가능한 형태로 확장한 S변환에 비해 복잡성의 문제점을 가진다.

2.4. 리프팅 구조를 이용한 효율적인 정수 변환

기준에 가역 변환용으로 제시되었던 변형 H.264/AVC 방법은 변환 계수의 범위가 커서 무손실 압축에서 효율적이지 못하였다.

따라서 앞선 연구[8]에서는 이러한 문제점을 해결하기 위해 고속 연산을 수행 하도록 리프팅 구조를 기반으로 설계한 변환인 Modified Reversible Integer Transform(MRIT)을 제안하였다. 여기서 제안된 변환의 파라미터 계수에서 2/5는 5로 나누는 요소가 변환의 복잡도를 증가시키므로 이를 간단한 쉬프트 연산과 덧셈 연산으로 구현이 가능하도록 1/2로 근사시켜 최적화한 뒤 그 성능을 실험으로 보였다. 그림 5에서 무손실 압축에 효율적이도록 최종적으로 제시한 변환에 대한 흐름도를 보여준다.

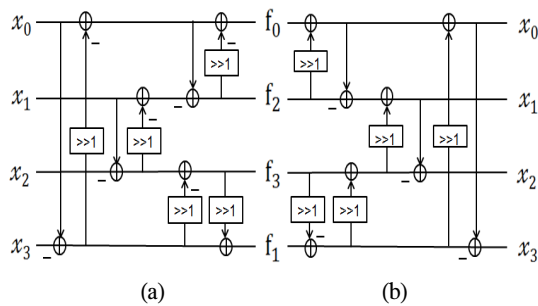


그림 5. 리프팅 구조를 이용한 효율적인 정수 변환 (a) 변환 (b) 역변환

Fig. 5 Efficient integer transform using lifting structure (a) forward transform (b) inverse transform

이 4점 변환을 변형 H.264/AVC를 2차원으로 확장할 때와 동일한 방법으로 분리 가능한 방식으로 4×4 블록의 16개의 픽셀에 대해서 구현한다.

III. 실험 및 결과

본 논문에서는 1차원에서 제시된 정수 변환 방법인 변형 H.264/AVC와 MRIT 변환을 2차원으로 확장하고 2차원적으로 설계된 JPEG XR에서의 PCT변환 방법과의 비교를 통해 그 성능을 분석하기 위한 실험을 수행한다. 실험 검증을 위해 성능과 복잡도의 두 가지 측면에서 진행한다. 실험에서 사용된 영상은 Lena 영상과 Baboon 영상을 시계방향으로 90도 회전 시킨 512×512 크기의 흑백 영상이다.

먼저, 성능을 평가하기 위해 영상을 4×4 블록의 16개 픽셀 단위로 나누어 변환을 수행한 뒤 같은 위치에 있는 변환 계수(transform coefficient)들을 모은다. 대역별로 모아진 16개의 변환 계수들에 대한 1차 엔트로피를 구하여 각 변환 간의 성능을 비교 평가 한다. 각 블록에 대한 변환 계수의 배치 순서는 그림 6에 제시되었다.

또한 신호 흐름도의 연산량을 계산하여 이론적인 복잡도를 추정하고, 소프트웨어의 구현을 통해 실제 연산 속도를 분석한다. 소프트웨어의 PCT 구현시 $T_h, T_{odd}, T_{odd_odd}$ 연산에 관한 함수는 [4]의 흐름도를 바탕으로 [9]의 코드를 참조하여 최적화시켜 구현하였다.

먼저, 변환 계수 값에 대한 1차 엔트로피 값을 통해 변환 성능을 분석할 수 있는데 표 1은 Lena와 Baboon 영상에 대한 실험 결과를 보여 준다.

| | | | |
|----------|----------|----------|----------|
| f_0 | f_1 | f_2 | f_3 |
| f_4 | f_5 | f_6 | f_7 |
| f_8 | f_9 | f_{10} | f_{11} |
| f_{12} | f_{13} | f_{14} | f_{15} |

| | | | |
|----------|----------|----------|----------|
| f_0 | f_2 | f_1 | f_3 |
| f_8 | f_{10} | f_9 | f_{11} |
| f_4 | f_6 | f_5 | f_7 |
| f_{12} | f_{14} | f_{13} | f_{15} |

(a) (b)

그림 6. 변환 계수 배치 (a) H.264 and MRIT (b) PCT
Fig. 6 Transform coefficient ordering (a) H.264 and MRIT (b) PCT

표 1. 엔트로피 실험 결과
Table. 1 Experimental results for entropy

| | Lena | | | Baboon | | |
|-----|--------|-------|-------|--------|-------|-------|
| | H.264 | PCT | MRIT | H.264 | PCT | MRIT |
| f0 | 11.252 | 9.362 | 7.388 | 11.005 | 9.106 | 7.133 |
| f1 | 7.850 | 6.213 | 4.919 | 9.592 | 6.958 | 6.673 |
| f2 | 6.390 | 4.999 | 3.438 | 8.805 | 6.216 | 5.864 |
| f3 | 6.544 | 4.416 | 3.576 | 8.920 | 6.451 | 5.986 |
| f4 | 8.551 | 5.503 | 5.633 | 9.323 | 7.226 | 6.390 |
| f5 | 7.646 | 3.397 | 5.694 | 9.324 | 5.112 | 7.382 |
| f6 | 6.637 | 3.498 | 4.664 | 8.658 | 5.518 | 6.704 |
| f7 | 6.679 | 3.991 | 4.698 | 8.914 | 5.373 | 6.961 |
| f8 | 6.970 | 4.414 | 4.029 | 8.184 | 6.854 | 5.226 |
| f9 | 6.774 | 3.598 | 4.802 | 8.376 | 5.175 | 6.409 |
| f10 | 5.998 | 4.019 | 4.031 | 7.843 | 5.869 | 5.870 |
| f11 | 6.197 | 4.455 | 4.214 | 8.220 | 5.560 | 6.249 |
| f12 | 7.083 | 4.562 | 4.126 | 8.103 | 6.153 | 5.144 |
| f13 | 6.941 | 3.710 | 4.962 | 8.303 | 5.991 | 6.337 |
| f14 | 6.290 | 3.930 | 4.315 | 7.869 | 6.414 | 5.896 |
| f15 | 6.670 | 4.993 | 4.687 | 8.365 | 6.657 | 6.401 |
| ave | 7.155 | 4.691 | 4.698 | 8.738 | 6.290 | 6.289 |

H.264/AVC 변환에서 변환 계수 값의 평균 엔트로피가 Lena 영상의 경우 7.155dB이고, Baboon 영상의 경우 8.738dB로 PCT와 MRIT에 비해 대략 2.5dB 정도 높은 반면, PCT와 MRIT의 두 변환은 거의 동일함 확인할 수 있다.

표 2. 연산 복잡도
Table. 2 Complexity of Operation.

| | H.264 | | PCT | | MRIT | |
|-------|---------|---------|---------|---------|---------|---------|
| | forward | inverse | forward | inverse | forward | inverse |
| add | 64 | 64 | 96 | 96 | 64 | 64 |
| shift | 24 | 16 | 44 | 44 | 40 | 40 |

제시된 표 2는 하나의 4×4 블록에 대해 각 변환과 그 역변환에 대한 연산량을 보여준다. 표의 수치는 [4]의 신호 흐름도에 따른 이론적인 연산 개수를 나타낸다. 복잡한 곱셈연산은 간단한 덧셈 연산과 쉬프트 연산으로 구현 가능하므로, 연산량 계산 시 신호 흐름도의 1/2에서 사용된 곱셈연산은 1개의 쉬프트 연산으로, 3/8과 3/4에

서는 1개의 덧셈 연산과 2개의 쉬프트 연산으로 변환하여 계산한다.

제안된 MRIT의 변환과 역변환에서의 덧셈 연산량은 각각 64개로 H.264/AVC 변환과 동일함을 보이지만, 쉬프트 연산량은 MRIT가 변환과 역변환의 두 경우 모두 40개로, 변환 시 24개가 필요하고 역변환에서 16개가 필요한 H.264/AVC에 비해 조금 많음을 알 수 있다. 따라서 전체 연산량 면에서 MRIT가 H.264/AVC에 비해 약간 비효율적으로 보이나, 앞서 제시한 표 1과 함께 고려해 보면 엔트로피의 향상에 비해 연산의 증가량이 미미함을 확인할 수 있다. 한편 PCT의 변환과 역변환은 모두 덧셈 연산 96개와 쉬프트 연산 44개로 다른 두 연산에 비해 확연히 복잡함을 보인다.

시간 복잡도는 소프트웨어로 구현했을 때의 연산 시간을 측정하여 구한다. 본 논문에서의 실험 환경은 쿼드 코어 CPU 2.66 GHz의 하드웨어 환경에서 구현하였고 변환간의 속도를 비교하기 위해서 각 변환을 100회 반복하여 측정한다. H.264/AVC와 MRIT는 변환의 평균 연산 수행 소요 시간이 각각 417ms와 498ms로, 694ms이 소요된 PCT에 비해 훨씬 우월함을 보인다. 또한 역변환의 결과도 이와 유사한 경향을 보이는데 H.264/AVC와 MRIT가 각각 426ms와 501ms로, 707ms이 걸린 PCT에 비해 상당히 효과적임을 알 수 있다.

IV. 성능 분석 및 토의

본 논문에서는 2×2 블록 단위의 변환에서 1차원의 2점 변환을 분리 가능한 형태로 구현한 S변환과 분리 가능하지 않은 형태로 구현한 2×2 하다마드 변환간의 성능의 유사함을 보이고, 이를 4×4 블록 단위로 확장하여 변환 간의 성능을 비교하였다.

변환 계수 값에 대한 1차 엔트로피 값을 통해 변환의 성능을 비교해 보았을 때 H.264/AVC가 다른 두 변환에 비해 엔트로피가 높아 성능 면에서 효율적이지 못함을 알 수 있다. 또한 PCT의 결과와 MRIT변환 결과의 엔트로피가 거의 동일한 것은 두 변환이 유사한 성능을 가지고 있음을 실험적으로 보여준다. 즉, 분리 가능한 형태의 변환과 분리 가능하지 않은 형태의 변환의 성능이 4×4 블록 단위의 2차원 무손실 정수 변환에서도 큰 차이가 없음을 확인할 수 있다.

따라서 이러한 유사한 성능의 변환을 평가하기 위해서는 복잡도 측면에서 비교해 보아야 한다. 동일한 성능을 보인 두 변환이 전체 연산량 면에서는 PCT변환이 MRIT변환에 비해 상당히 복잡함을 보였다. 또한, 연산 시간 면에서도 다소 비효율적임을 확인 할 수 있다. 따라서 성능과 복잡도 측면을 모두 고려해 보았을 때 2차원 변환으로 확장시킬 때 분리 가능한 형태로 수행한 MRIT 변환 방법이 가장 우수함을 알 수 있다.

V. 결 론

본 논문에서는 무손실용 정수 변환 방법으로 변형 H.264/AVC 정수 변환 및 JPEG XR의 PCT 변환, 리프팅을 이용한 효율적인 가역 정수 변환을 소개하였다. 또한 각 변환을 2차원으로 확장하여 변환 간의 성능 및 복잡도 분석 실험을 통해서 이를 비교 평가하였다. 실험 결과로 본 논문에서 제안된 방법 중 리프팅 구조를 이용한 효율적인 정수 변환 방법을 수직, 수평 방향에 대해 반복적으로 분리 가능한 형태로 구현한 방법이 가장 우수함을 보였다. 결론적으로, 분리 가능하지 않은(non-separable) 형태로 확장한 변환보다 분리 가능한(separable) 형태로 2차원으로 확장한 변환 방법이 추천된다.

참고문헌

- [1] Tu, C. and T.D Tran, "Context-based entropy coding of block transform coefficients for image compression," IEEE Trans. Signal Processing, Vol. 11, No. 11, Nov. 2002.
- [2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circ. Syst. Video Tech., Vol. 13, No. 7, July 2003.
- [3] Chia-Ho Pan, Ching-Yen Chien, Wei-Min Chao, Sheng-Chieh Huang, and Liang-Gee Chen, "Architecture design of full HD JPEG XR encoder for digital photography applications," IEEE Trans. Consumer Electronics, Vol. 54, No. 3, Aug. 2008.

- [4] C. Tu, S. Srinivasan, G. J. Sullivan, S. Regunathan, and H. S. Malvar, "Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR / HD Photo," in Proc. SPIE, Aug. 2008.
- [5] F. Dufaux, G. Sullivan, and T. Ebrahimi "The JPEG XR image coding standard," IEEE Signal Processing Magazine, vol. 26, No. 6, p. 195-199, 204-204, Nov. 2009.
- [6] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," IEEE Trans. Circ. Syst. Video Tech., Vol 13, No. 7, July 2003.
- [7] A. Hallapuro, M. Karczewicz, and H. Malvar, "Low complexity transform and quantization - part II: extentions," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 JVT B-039, Jan. 2002.
- [8] 유훈, "무손실 압축을 위한 H.264/AVC 정수 변환의 변형," 한국해양정보통신학회 논문지, 제 14권 10호, pp.2255-2260, 2010.
- [9] Microsoft Corp., "HD photo device porting kit, "http://www.microsoft.com/whdc/xps/hdphotodpk.mspx, 2006.

1

저자소개

김희경(Hui-Gyeong Kim)



2012년 현재 상명대학교
디지털미디어 학·석사
연계 과정
※ 관심분야: 영상처리, 영상압축

유 훈(Hoon Yoo)

한국정보통신학회 논문지
제15권 제5호 참조