

경량 비디오 코덱을 위한 3D 웨이블릿 코딩 기법

이 승 원[†]·김 성 민^{††}·박 성 호^{†††}·정 기 동^{††††}

요 약

비디오 압축에 사용되는 움직임 예측은 많은 계산과정을 요구하기 때문에 전체적인 부호기 복잡도를 높이는 단점을 지닌다. 이러한 부호기의 복잡도를 줄이기 위해 3D-WT과 같은 움직임 예측을 사용하지 않는 연구들이 소개되고 있다. 하지만, 기존의 3D-WT 기법들은 부호화를 위한 과도한 메모리 요구사항과 복호를 위한 수신 측의 지연시간이 가장 큰 단점으로 지적되었다. 본 논문에서는 수정된 Haar wavelet filter와 개선된 부호화 알고리즘을 통해서 메모리 사용량과 재생을 위한 지연시간을 최소화 하는 확장 가능한 3D-WT 기법인 FS(Fast playable and Scalable) 3D-WT를 소개한다. 3D-WT 중 가장 개선된 형태인 3D-V 기법과의 실험 결과 3D-V와 거의 비슷한 계산 처리 시간으로 높은 압축률과 수신 측에서의 짧은 지연시간을 보였다.

A 3D Wavelet Coding Scheme for Light-weight Video Codec

Seung-Won Lee[†]·Sung-Min Kim^{††}·Seong-Ho Park^{†††}·Ki-Dong Chung^{††††}

ABSTRACT

It is a weak point of the motion estimation technique for video compression that the predicted video encoding algorithm requires higher-order computational complexity. To reduce the computational complexity of encoding algorithms, researchers introduced techniques such as 3D-WT that don't require motion prediction. One of the weakest points of previous 3D-WT studies is that they require too much memory for encoding and too long delay for decoding. In this paper, we propose a technique called 'FS (Fast playable and Scalable) 3D-WT'. This technique uses a modified Haar wavelet transform algorithm and employs improved encoding algorithm for lower memory and shorter delay requirement. We have executed some tests to compare performance of FS 3D-WT and 3D-V. FS 3D-WT has exhibited the same high compression rate and the same short processing delay as 3D-V has.

키워드 : 웨이블릿 코딩(Wavelet Coding), 경량코덱(Light-Weight Codec), 비디오 인코딩(Video Encoding)

1. 서 론

유선환경에 비해 많은 제약 사항을 가지는 무선 환경에서 멀티미디어 서비스를 지원하기 위해서는 제약 조건을 극복하기 위한 여러 기술적인 지원이 요구된다. 대역폭과 관련된 제약 조건은 IMT2000과 같은 무선 통신 기술의 발전으로 점차 해소되고 있다. 그러나 무선 이동 통신환경에서와 같이 디바이스의 능력에 한계를 가진 경우 MPEG-4와 H.263과 같은 멀티미디어 포맷들의 코덱 계산 처리 복잡도와 그에 따른 배터리 수명시간의 단축은 아직까지 해결하기 힘든 부분이다. 이러한 코덱의 계산 복잡도에 관련된 제한 사항을 해결하기 위해 경량 코덱기술이 많이 연구되고 있고, 유선환경에서의 멀티미디어 서비스를 무선 이동 환경으로 옮길 수 있는 기술로 관심을 모으고 있다.

유선과 무선 이동 환경에서 비디오 데이터를 서비스하기 위해서는 여러 가지 압축 코딩 기술들이 필요하다. 멀티미디어 데이터의 경우 기존의 텍스트 데이터와 달리 대역폭의 저장 공간과 높은 대역폭을 요구한다. 예를 들면, 1024×768 이미지 크기, 16비트 칼라, 그리고 초당 30프레임으로 압축을 하지 않은 1시간 분량의 비디오 클립은 170GB 이상의 저장 공간과 48MB/s 이상의 대역폭을 요구한다. 그러나 이처럼 높은 대역폭을 이용해서 서비스하는 것은 네트워크의 막대한 자원낭비를 초래하게 된다. 따라서 비디오 데이터를 서비스하기 위해서 여러 가지 압축 코딩 기술이 사용되는데 MPEG 계열과 H.26x와 같은 동영상 압축 표준에서는 다음과 같은 기술이 사용된다. 먼저 각각의 프레임은 DCT를 이용하여 변환 후 양자화와 가변장 부호화 방식을 이용하여 압축하고, 시간적으로 연속되는 프레임들은 움직임 예측 기법을 이용하여 시간적인 중복성을 제거한다[1-5]. 그러나 DCT 변환 기법은 높은 압축률에서 화면의 질이 급격히 떨어지는 단점이 있고, 움직임 예측 기법

† 준 회원 : 부산대학교 대학원 전자계산학과

†† 준 회원 : 부산대학교 대학원 컴퓨터공학과

††† 정 회원 : 부산대학교 정보전산원 교수

†††† 종신회원 : 부산대학교 전자계산학과 교수

논문접수 : 2004년 1월 20일, 심사완료 : 2004년 2월 26일

은 많은 계산으로 인해 부호기 전체의 복잡도를 높이는 원인이 된다.

블록 단위로 처리하는 DCT 변환 기법은 블록과 블록이 독립성을 유지하며 변환되기 때문에 압축률이 높아질수록 주변의 블록들 사이의 값에 상당한 차이를 보이는 block-ing 현상이 발생한다[9]. 이에 반해, 이미지 전체를 하나의 입력 단위로 처리하는 웨이블릿 변환 기법은 높은 압축률에서도 DCT에 비해 좋은 화질을 유지할 수 있다[8]. 따라서 현재 이미지 코딩을 위해서 주로 이용되어 오던 웨이블릿 변환 기법이 비디오 코딩을 위한 대안으로 많은 연구가 진행되고 있으며, 이는 상대적으로 대역폭이 협소한 무선 환경의 서비스를 위해서도 적절한 선택이라고 볼 수 있다.

비디오 코딩 기법에서는 연속되는 프레임 간에 움직임 예측 기법을 사용하여 데이터를 압축한다. 전체 압축 과정에서 상당 부분의 압축이 시간적인 중복성을 제거하는 움직임 예측 기법을 통해서 이루어진다. 그러나 움직임을 예측하기 위해 투자되는 복잡한 계산과정은 부호기 전체의 복잡도를 높이는 단점으로 작용한다. 이를 극복하기 위해 시간적으로 연속적인 일정양의 프레임들을 하나의 입력단위로 처리해 움직임 예측을 하지 않고 코딩하는 3D-WT(Wavelet Transform) 기법에 대한 연구들이 소개되고 있다[18-22]. 이러한 3D-WT 기법들은 움직임 예측에 비해 계산 복잡도가 낮은 웨이블릿 변환을 사용하기 때문에 비디오 코딩의 전체적인 계산 복잡도를 낮추어 준다.

앞서 언급한 3D-WT 기법들은 여러 장의 프레임들 하나의 단위로 처리함에 따라 발생하는 복호기 측에서의 지연과 여러 장의 프레임들 저장하기 위한 메모리상의 제약사항을 가진다. 본 논문에서는 개선된 부호화 알고리즘을 통해서 3D-WT 사용자 복호기 측에서 발생하는 지연시간과 부호기 측에서 발생하는 메모리의 요구사항을 최소로 하는 FS 3D-WT(Fast playable and Scalable 3D-Wavelet Transform) 기법을 제시하였다. 실험을 통한 결과에서 우리가 제시한 기법이 기존의 기법에 비해 압축률의 향상과 복호기 측에서의 재생 지연시간에 향상된 성능을 보임을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 소개한다. 3장에서는 제안하는 기법인 FS 3D-WT(Fast playable and Scalable 3D-Wavelet Transform) 기법을 소개하고, 4장에서 실험을 통해 기존의 기법과의 성능 비교를 보인다. 마지막으로 결론과 향후과제는 5장에서 주어진다.

2. 관련 연구

움직임 예측 기법은 웨이블릿 기법에 비해 DCT 기법에 좀 더 적합한 특징을 가지고 있어, 기존의 DCT를 사용하는 비디오 코딩 기법에서는 대부분 이 기법을 사용한다. 그

러나 오류의 전파와 계산의 복잡도와 같은 여러 가지 단점으로 인해 DCT 뿐만 아니라 웨이블릿을 사용하는 비디오 코딩 기법에서도 움직임 예측 기법을 사용하지 않고 비디오 데이터를 압축하는 방법에 대해서 연구가 되고 있다[18-22].

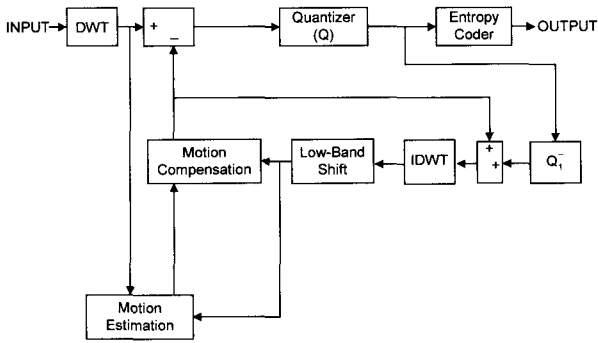
본 절에서는 웨이블릿을 사용한 비디오 코딩 기법 중에서 움직임 예측을 사용한 기법들과 움직임 예측을 사용하지 않고 3차원 영역에서 비디오 데이터를 코딩하는 기법인 3D-WT(3-Dimension Wavelet Transform) 기법들에 대해서 간단히 소개한다.

2.1 움직임 예측을 사용하는 기법

웨이블릿 기법이 비디오 코딩에서 적용될 때 웨이블릿 변환 기법이 가지는 특징을 최대한 살리기 위해 DCT를 사용한 기법들과는 다른 방법으로 움직임 예측을 하게 된다[12-17]. 기존의 MPEG-1, MPEG-2, MPEG-4, H.261, H.263과 같은 DCT를 변환 기법으로 사용하는 비디오 코딩 기법에서는 움직임 예측을 할 때 매크로 블록 단위로 하게 되며 공간 영역에서 움직임 예측을 한다[1-5]. 마찬가지로 웨이블릿을 사용하는 비디오 코딩에서도 공간 영역에서 움직임 예측을 통해 연속적인 프레임의 정보를 압축할 수 있다. 이럴 경우 DCT를 사용한 기존의 비디오 코딩 기법에서 단순하게 웨이블릿으로 바꾸어 적용한 것에 불과하고, 블록 단위로 움직임 예측을 하기 때문에 웨이블릿 방식의 코딩 효율을 떨어뜨릴 수 있다. 그리고 부호화와 복호화의 웨이블릿 계산의 부하로 인해 공간영역이 아닌 웨이블릿 영역에서의 움직임 예측이 필요하게 하였다.

웨이블릿 영역 상에서의 움직임 예측은 웨이블릿 변환이 가지는 shift variant 특성으로 인해 움직임 예측의 결과가 공간영역에 비해 나쁜 결과를 가져온다. Shift variant는 원래의 이미지가 1픽셀만큼 이동했을 경우 웨이블릿 변환된 계수의 값이 크게 변화되는 특성으로, 웨이블릿 영역 상에서 움직임 예측을 비효율적으로 만든다. [15]에서는 웨이블릿 영역 상에서의 움직임 예측과 관련된 문제를 해소하는 방법을 제시했다. Low-band shift method라는 방법을 사용하는데 (그림 1)과 같이 미리 웨이블릿 변환된 계수를 통해서 움직임 예측을 할 수 있도록 한다. 이 기법은 가로, 세로, 대각선 방향으로 공간영역 상에서 미리 픽셀 값을 shift를 해서 웨이블릿 변환을 한다. 따라서 새롭게 웨이블릿 변환되어 들어온 프레임은 기존의 여러 가지 변환된 프레임 중에서 가장 잘 맞는 프레임을 참조하므로, 앞에서 언급한 shift variant의 문제를 해결할 수 있다. 그러나 성능의 우수함에 비해 메모리 사용량이나 복잡도에 있어서는 다른 기법에 비해 떨어지는 단점을 가지고 있다. 그리고 움직임 예측이 블록 단위로 이루어지기 때문에, 움직임을 보강한 결과화면에서 blocking 현상이 나타날 수도 있다.

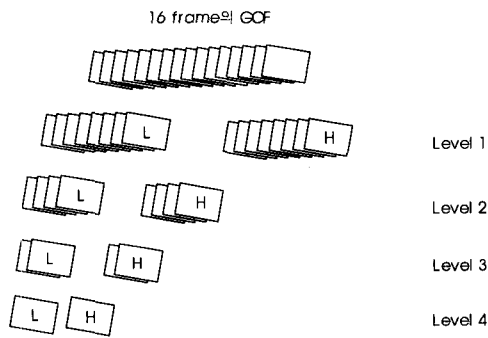
[14]에서는 [15]와 같은 기법을 이용하여 움직임 예측을 하므로, shift variant의 문제를 해결할 수 있다. 그리고 [15]와 같은 화면의 에지에서 발생할 수 있는 blocking 현상을 줄이기 위해, 블록과 블록을 overlapping 시켜 움직임 보상을 함으로서 blocking 현상을 줄일 수 있는 기법을 제안하였다.



(그림 1) Low-Band Shift 기법을 사용한 코딩 기법의 블록도

2.2 3D-WT기법

기존의 비디오 코딩 기법에서 사용된 움직임 예측은 부호기 내의 계산 복잡도를 높이는 직접적인 원인이 되었다. 최근에는 웨이블릿 부호화 기법을 바탕으로 부호기 내의 계산 복잡도를 줄이기 위해 움직임 예측을 사용하지 않는 기법이 연구되고 있다. 3차원 영역에서 웨이블릿을 통해서 코딩을 하는 3D-WT 기법은 이처럼 여러 장의 프레임들 하나의 입력단위로 처리한다.

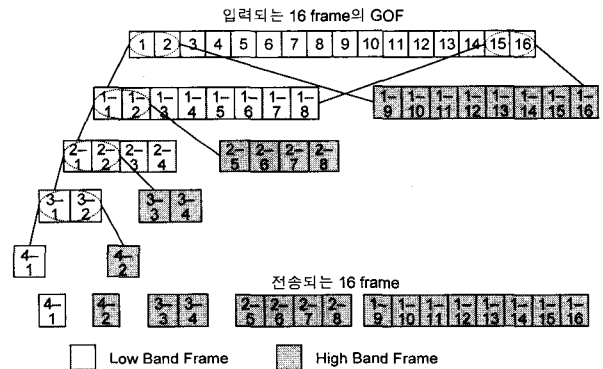


(그림 2) 16프레임의 시간적 분해과정

전통적으로 소개된 3D-WT 기법은 전체 비디오를 하나의 시퀀스로 처리한다. 즉, 긴 시간의 비디오 데이터에 대해서 하나의 입력단위로 처리해야 하므로 이러한 기법은 개념적인 소개일 뿐 실제로 적용하지 못하였다. 이런 기법의 단점을 극복하고 실현 가능성을 위해 전체 시퀀스의 길이에 상관없이 특정 크기의 GOF(Group of Frame) 단위로 입력받는 개념이 도입되었다. (그림 2)는 16개의 프레임으로 구성된 GOF 단위로 웨이블릿 변환을 하는 과정을 나타낸다.

(그림 2)에서 시간적으로 연속된 16개의 프레임은 하나의 GOF를 구성한다. 각각의 프레임들에서 같은 위치의 픽셀 값들은 웨이블릿 변환을 통해서 시간적으로 분해되고 4번의 분해과정을 통해 1개의 저대역 프레임과 15개의 고대역 프레임으로 나누어진다. 이처럼 시간적인 분해를 통해서 얻어진 고대역 프레임들은 전체적으로 작은 값들로 이루어져 있으므로, 높은 압축 효율을 얻을 수 있다.

반면 (그림 2)를 통해서 알 수 있듯이 연속적인 프레임들을 통해 시간적인 분해를 하기 위해서는 우선 GOF의 크기에 해당하는 16장의 프레임을 입력받아 처리를 해야 한다. 따라서 부호기 측에서 부호화를 위해서는 최소한 GOF 크기의 메모리를 필요로 하게 되므로, 메모리에 대한 요구량이 많아지는 단점을 가지게 된다. 그리고 (그림 3)에서 보는 것처럼 전송된 비디오 데이터를 복호기 측에서 복호하기 위해서는 긴 시간의 지연이 발생함을 알 수 있다. (그림 3)에서는 보듯이 복호기 측에서는 부호기 측에서 전송된 데이터의 첫 번째 프레임을 복호하기 위해 최소 9 프레임은 수신해야만 재생이 가능하다. 즉, 복호기 측에서 재생을 위해서 필요한 지연시간은 GOF 크기의 1/2 이상으로 길어짐을 알 수 있다.

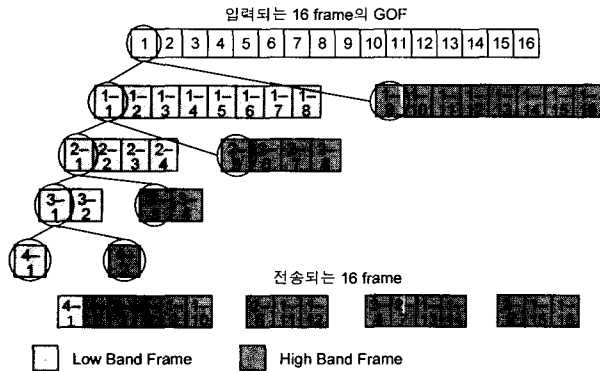


(그림 3) 시간적인 분해후의 전송 순서

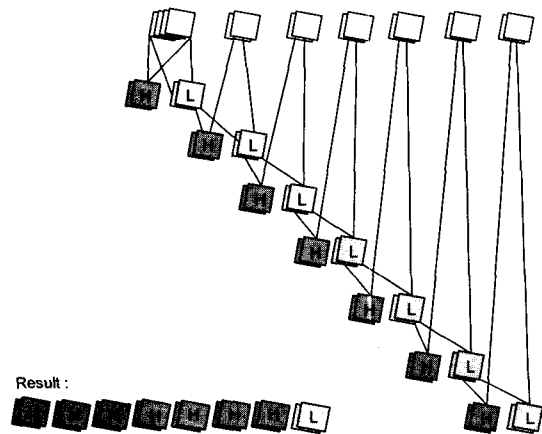
(그림 4)를 보면 (그림 3)에서 설명하는 복호기 측의 지연시간을 줄이기 위해 시간적인 분해를 통해 만들어진 프레임들의 전송 순서를 재조정한다. 그러나 부호기 측에서의 처리를 위한 메모리 요구사항은 여전히 단점으로 존재한다.

[19]에서는 부호기 측에서 여전히 남아있는 메모리 요구사항에 대한 단점을 해결하는 방법을 제시하였다. (그림 5)는 부호기 측에서 메모리 요구사항을 줄이고, 전송을 위한 지연시간을 줄이는 새로운 부호화 알고리즘을 설명하고 있다. (그림 5)의 알고리즘에서는 GOF를 구성하는 전체 프레임들을 입력으로 받아들이지 않아도 된다. 따라서 부호기 측에서 요구하는 메모리에 대한 사용량을 줄일 수 있다. 또한 각각의 분해 과정이 끝날 때 마다 바로 복호기 측으로 전송할 수 있다는 장점을 지닌다. 그러나 분해 과정의 특징상 가장 중요한 정보를 담고 있는 저대역 프레임이 GOF의 가

장 마지막 단계에서 생성이 되므로, 복호기 측에서 재생을 위해서는 맨 마지막에 전송되는 저대역 프레임에 기다려야 하는 단점이 존재한다. 따라서 부호기 측에서의 메모리에 대한 요구사항은 해결할 수 있지만, 복호기 측에서의 재생을 위한 지연시간에서는 여전히 문제점을 안고 있다.



(그림 4) 복호기 측의 지연시간을 줄이기 위한 프레임 전송순서 재조정



(그림 5) [19]에서 제시한 부호화 알고리즘

3. FS 3D-WT

앞선 관련 연구에서 살펴본 3D-WT 기법들은 다음과 같은 단점을 지니고 있다. 첫째, 입력되는 데이터가 프레임의 그룹이기 때문에 부호화 단계에 많은 메모리가 필요하게 된다. 둘째, 부호화 후 전송해서 복호기 측에서 재생하기까지는 긴 지연 시간을 가지게 된다. [19]에서는 이런 단점 중 부호기 측의 메모리 문제에 대한 해결책을 제시했으나 여전히 비디오 재생의 지연 문제는 단점으로 남아 있다. 본 논문에서는 앞서 언급한 단점을 개선한 경량코덱에서 사용할 수 있는 3D-WT 비디오 코딩 기법인 FS 3D-WT 기법을 제안한다. 성능 평가를 위해서 기존의 기법과 FS 3D-WT 기법을 비교한 결과, 기존의 기법에 비해 압축률과 재생 지연시간에 있어 성능 개선을 보였다.

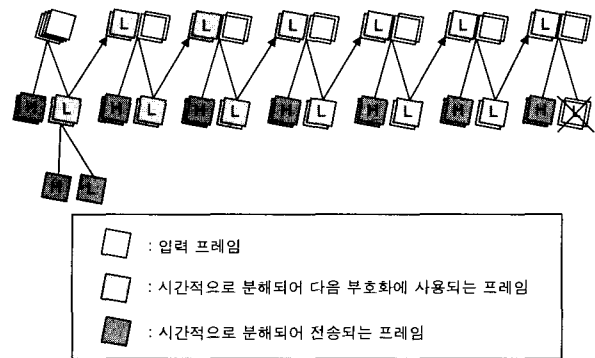
또한 기존의 기법들은 GOF를 구성하는 프레임 수가 커

짐에 따라 압축률 향상을 가져오는 데 반해, 부호기 측의 메모리 사용량이나 복호기 측의 재생 지연시간도 늘어나는 단점을 가진다. 그러나 FS 3D-WT 기법은 복호기 측에서 발생되었던 재생을 위한 지연시간과 부호기 측에서의 메모리 사용량을 4개의 프레임으로 고정시켰기 때문에 GOF를 구성할 때 다른 기법들에 비해 큰 확장성을 가지고 있다.

3.1 Fast playable and Scalable(FS) 3D-WT

3.1.1 부호화 알고리즘

제안하는 기법인 FS 3D-WT 기법은 메모리와 지연시간에 관한 효율성을 제공할 뿐만 아니라, GOF 크기를 결정할 때도 성능에 있어 영향을 크게 받지 않는 확장성을 지니고 있다. (그림 6)에서는 본 논문에서 제안하는 기법에 대한 부호화 알고리즘의 개요를 보이고 있다.

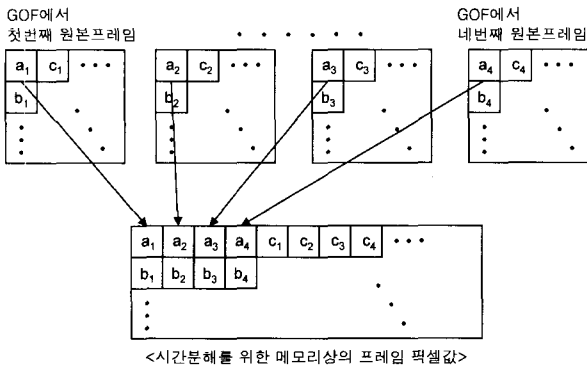


(그림 6) 메모리 요구사항과 지연시간을 줄이는 새로운 알고리즘

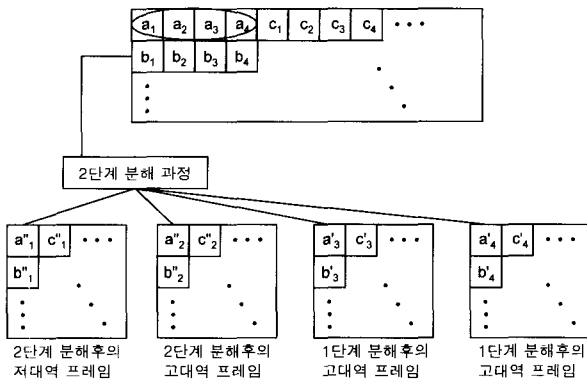
제안하는 기법은 기존의 움직임 예측기법을 적용했을 경우와 달리 여러 장의 프레임을 입력 받아 부호화를 처리하게 되는데, 기존에 제시되었던 3D-WT 기법들과 같은 방법으로 특정 크기의 GOF를 입력으로 받아들인다. 본 논문에서 제시한 FS 3D-WT 기법은 부호화 알고리즘의 특성상 압축률을 제외하고 GOF의 크기에 따른 성능의 차이가 없다. 그러나 설명의 편의를 위해 16개의 프레임을 하나의 GOF로 설정하였다.

기존의 기법들은 GOF의 모든 프레임을 받아들인 후 연속된 프레임들을 부호화한다. 반면, FS 3D-WT 기법에서는 입력되는 GOF의 크기에 상관없이 4개의 프레임을 중심으로 이루어진다. FS 3D-WT의 부호화 과정은 다음과 같다. (그림 6)과 같이 GOF에서 처음으로 입력되는 4개의 원본 프레임은 먼저 2단계 분해되어 1개의 저대역 프레임과 3개의 고대역 프레임으로 생성된다. 이때, 실제적인 분해를 위한 메모리상의 픽셀 값들은 (그림 7)과 같다. 메모리상의 픽셀 값들은 2단계 분해과정을 통해 (그림 8)과 같이 저대역과 고대역 프레임을 생성한다. 이때 분해된 4개의 프레임은 첫 번째 단계에서의 전송할 프레임이 된다. 복호기 측에

서는 전송 받은 이 4개의 변환 프레임을 이용하여 GOF의 첫 4개의 원본 프레임을 즉시 복호할 수 있다. 따라서 기존의 기법보다 재생을 위한 지연이 감소하게 된다.



(그림 7) GOF에서 첫부호화 과정을 위한 메모리의 모습



(그림 8) GOF의 첫 부호화에서 저·고대역 프레임 생성과정

GOF에서 첫 부호화 단계이후 과정은 다음과 같다. 최초로 입력된 4개의 프레임을 2단계 분해·전송한 후 이어진 2개의 원본 프레임을 입력으로 받아들인다. 부호화 단위인 4개의 프레임 중에 나머지 2개의 프레임은 앞서 입력되어 1단계 분해에서 얻어진 저대역 프레임이다. (그림 6)에서 보듯이 이전 단계에서 분해된 저대역 2개의 프레임과 새롭게 입력되는 2개의 원본 프레임을 합한 4개의 프레임을 부호화 단위로 하여 1단계의 분해과정을 거친다. 여기에서 주의할 점은 GOF의 첫 부호화 단계 이후의 과정들은 1단계의 분해과정만을 거친다는 것이다.

즉, 압축률을 높이기 위해서 GOF에서 처음으로 입력되는 4개의 원본 프레임만 2단계 분해과정을 거치게 된다. 나머지 단계에서는 고대역 프레임만을 전송하고 저대역 프레임은 다음 단계의 부호화 과정을 위해서 사용되기 때문에, 2단계의 분해과정이 필요 없게 된다. 이 과정은 GOF에 해당하는 모든 프레임들이 입력될 때까지 진행되며, 맨 마지막 단계에서 발생하는 저대역 프레임은 전송하지 않아도 복호기 측에서는 복호가 가능하게 된다. (그림 9)는 제안한 기법의 부호화 알고리즘을 나타내고 있다.

```

for ( i = 0 ; i < GOF/2 - 1 ; i++){
    if( i == 0 ){ // GOF의 첫번째 부호화 과정
        for ( j = 0 ; j < 4 ; j++){ // 첫 4프레임
            read_Y, read_U, read_V; // 입력 file로 부터
            Y,U,V 읽음
            store frame_buf [0] frame_buf [3];
            /* 4프레임을 메모리에 저장 */
        }
        level1_temporal_decomposition(frame_buf [0] frame_
        buf [3]) ;
        /* 1단계 시간분해 */
        /* frame_buf[0-1] : low band frame, frame_buf
        [2-3] : high band frame */
        level2_temporal_decomposition(frame_buf [0] frame_
        buf [1]) ;
        /* 2단계 시간분해 */
        /* frame_buf[0] : low band frame, frame_buf[1] :
        high band frame */
        send_data(frame_buf[0] frame_buf[3]);
    }
    else{
        for ( j = 0 ; j < 2 ; j++){ // 두개의 프레임씩 읽음
            read_Y, read_U, read_V; // 입력 file로부터
            Y,U,V 읽음
            store frame_buf [2] frame_buf [3];
            level1_temporal_decomposition(frame_buf [0]
            frame_buf [3]) ;
            /* 1단계 시간분해 */
            send_data(frame_buf[2] frame_buf[3]);
            /* frame_buf[0-1]은 low band frame으로 다음
            부호화에 사용되며, high band frame은!
            frame_buf[2-3]을 전송함 */
        }
    }
}
    
```

(그림 9) FS 3D-WT의 부호화 알고리즘

3.1.2 웨이블릿 필터(Wavelet Filter)

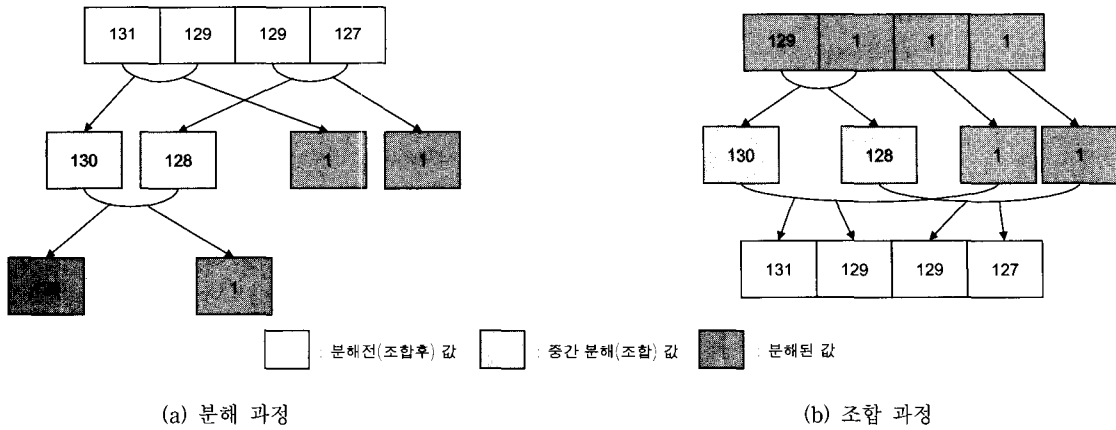
프레임들간의 시간적 상관관계를 변환 압축하기 위해서는 웨이블릿 필터가 필요하다. FS 3D-WT 기법에서는 프레임들을 변환하기 위해서 개선된 Haar wavelet을 사용하였다. Haar wavelet은 입력된 두 수에 대해서 아래의 식으로 계산되어 분해한다.

- 입력되는 수 : a, b,
- 분해되어 만들어진 웨이블릿 계수 : α , β
- Low pass filter : $(a+b) / 2 = \alpha$
- High pass filter : $(a-b) / 2 = \beta$

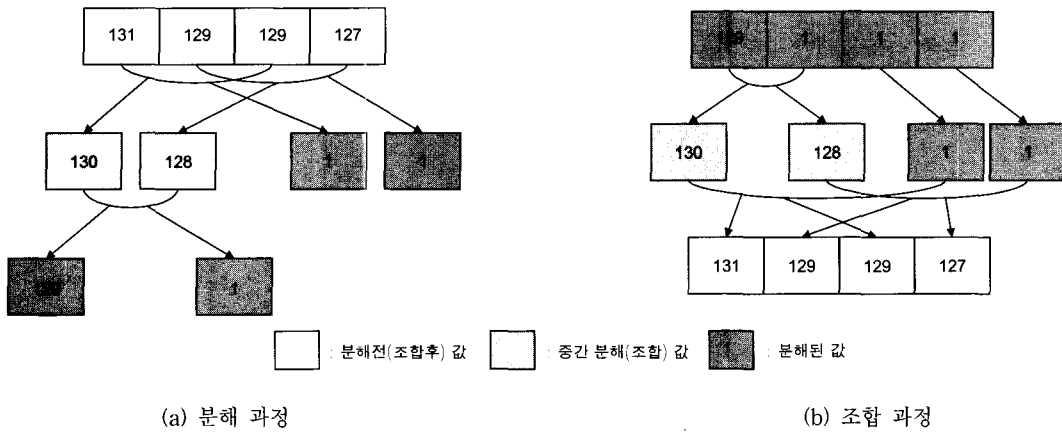
조합과정은 아래와 같다.

- $\alpha + \beta = a$
- $\alpha - \beta = b$

따라서 4개의 프레임을 대상으로 Haar wavelet 을 이용할 경우 (그림 10)(a)와 같이 분해된다. 이 때 4개의 값은 각각의 프레임에서 특정 픽셀의 값들이다. 분해된 값의 조합과정은 (그림 10)(b)에 나타나 있다.



(그림 10) Haar wavelet을 통한 값의 분해과정과 조합과정



(그림 11) 수정된 필터를 통한 값의 분해와 조합과정

본 논문에서 제안하는 알고리즘에서 사용하는 웨이블릿 필터는 위에서 설명된 Haar wavelet을 수정해서 사용하였다. (그림 10)에서 알 수 있듯이 Haar wavelet을 통해서 본 논문에서 제안하는 알고리즘을 만족할 수 없다. 본 논문이 4개의 프레임을 중심으로 부호화하기 때문에 <표 1>과 같이 수정된 필터를 이용하였다.

<표 1> 수정된 Haar wavelet filter

	Level 1	Level 2
Low pass filter	$[1/2, 0, 1/2, 0]$ $[0, 1/2, 0, 1/2]$	$[1/2, 1/2]$
High pass filter	$[1/2, 0, -1/2, 0]$ $[0, 1/2, 0, -1/2]$	$[1/2, -1/2]$

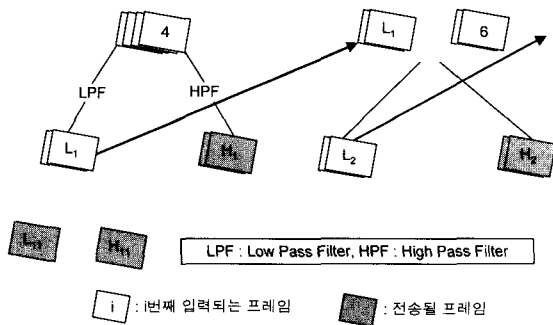
<표 1>에서 제시되어 있는 필터를 사용하여 값을 분해하고 조합하는 과정은 (그림 11)에 나타나 있다. 앞서 언급했듯이 본 논문에서 제시하는 부호화 알고리즘은 이전 부호화 단계의 저대역 2개의 프레임과 새로 입력되는 2개의 프레임을 통해 시간적인 분해를 하고 고대역 2개의 프레임을 전송한다. 그리고 수신 측에서는 이전 복호 단계 저대역 2개의 프레임과 새로 전송되어 온 고대역 2개의 프레임을

통해 원래의 값으로 조합되는 2개의 프레임과 다음단계에서 사용할 저대역 2개의 프레임을 얻게 된다. (그림 11)을 통해 알 수 있듯이 수정된 필터를 이용하게 되면 이 과정이 가능하게 된다.

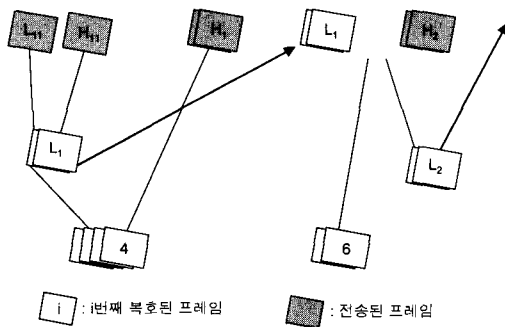
(그림 12)에서는 입력되어 들어오는 프레임들에 대해서 GOF의 크기에 상관없이 초기의 6개의 프레임에 대해서 부호화 하여 전송하고, 전송된 프레임을 복호화 하는 과정을 보여준다. GOF에 포함되는 이후의 프레임들은 (그림 12)에서 5, 6번째 프레임과 같은 방식으로 진행된다.

(그림 12)(a)를 통해서도 알 수 있듯이, 실제 전송되는 데이터와 시간적인 분해를 하기 위한 프레임을 다르게 둬으로써 연속적인 분해를 위한 메모리 사용을 줄이고 복호를 위해 긴 지연시간을 요구하지 않고서도 가능하게 하였다. 그리고 제안하는 기법은 4개의 프레임을 작업단위로 가지므로, 메모리 요구사항은 GOF의 크기에 상관없이 4개의 프레임으로 유지된다. 기존의 기법들은 GOF 크기에 따라서 메모리 요구사항이 달라지나, 본 논문에서는 부호화 단위로 4개의 프레임을 사용하기 때문에 GOF 크기에 상관없이 일정하게 메모리 사용량을 유지할 수 있다. 그러므로 압축률 조절을 위해 GOF 크기를 조절할 수 있는 확장성을

제공한다. 또한, 수신 측에서 재생을 위한 지연시간도 GOF의 크기에 상관없이 항상 4프레임만 수신하면 가능하기 때문에 일정하게 유지시킬 수 있다.



(a) 부호화 과정



(b) 복호화 과정

(그림 12) GOF의 최초 6개의 프레임에 대한 부호화와 복호화의 예

4. 실험 및 성능 평가

본 장에서는 여러 가지 측정 인자를 이용하여 본 논문에서 제안한 FS 3D-WT 기법과 기존에 제시되어 있는 기법을 실험하였다. 본 장의 나머지 부분에서는 실험 측정을 위한 여러 가지 파라미터를 이용한 실험결과를 통해서 FS 3D-WT 기법이 가지는 효율성에 대해서 알아보았다.

4.1 실험 환경

시간적으로 연속되어 입력되는 프레임들을 무선 이동 디바이스에서 서비스하기 위해서는 경량코덱의 사용이 필수적인 사항이라 볼 수 있다. 경량코덱의 평가 항목으로는 부호화를 위한 처리 시간, 데이터 압축률, 부호기 측에서의 메모리 요구량, 수신 측에서 야기되는 지연시간 등이 있다. 본 장에서는 이러한 항목들을 3D-WT 중 가장 개선된 형태인 [19]에서 제시된 3D-V 기법과 비교하여 성능을 평가하였다. 성능 평가를 위한 파라미터는 <표 2>에 나타나 있다.

<표 2> 실험 환경

CPU	Pentium - IV 2.4G
Memory	256MB
파일 포맷	qcif(176×144)
비디오 종류	foreman, carphone
측정 파라미터	부호화 처리시간, 압축율 메모리 사용량, 수신측 지연시간

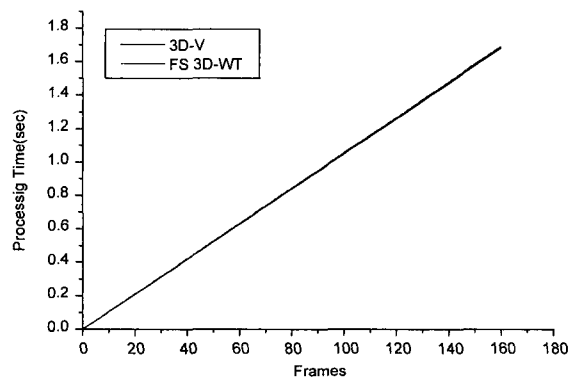
4.2 성능 평가

본 절에서는 논문에서 제안하는 FS-3DWT 기법의 부호화 처리 시간, 압축율, 부호기 측에서의 메모리 사용량, 수신측에서의 지연시간 측정을 통해 [19]에서 제안한 3D-V 기법과 성능을 평가·비교하였다.

4.2.1 부호화 처리 시간

<표 2>의 실험환경에 따라, 160개의 프레임을 입력으로 받아 시간적인 분해를 하는 부호화 처리 시간을 측정하였다. 두 기법들이 가지는 알고리즘의 특성상 3D-V 기법이 본 논문에서 제안한 FS 3D-WT 기법에 비해 빠른 처리 시간을 가질 것으로 예상되지만, 그 차이와 다른 파라미터를 통한 실험 결과와의 관계를 통해 기법의 효율성을 평가하기 위하여 본 실험은 이루어졌다. 각각의 기법은 움직임이 상대적으로 많은 foreman.qcif 파일을 통해서 부호화 처리 시간을 측정하였으며, 부호화한 프레임은 1-160프레임이다.

(그림 13)을 통해서 알 수 있듯이, 3D-V기법과 제안하는 기법과의 계산 처리 시간은 3D-V기법이 0.5% 정도 빠르게 나타났다. 160프레임을 대상으로 진행된 실행시간은 1.68초와 1.69초로 실제로 많은 프레임을 부호화하는데 있어서 큰 차이 없이 처리할 수 있음을 확인할 수 있었다.



(그림 13) 부호화 처리시간 비교

4.2.2 시간 분해후의 압축률

연속적인 프레임을 움직임 예측 없이 압축하게 되는 3D-WT 기법들은 시간 분해를 통해서 데이터를 압축하게 되는데, 부호화 알고리즘에 따라 시간 분해 후의 압축률은 차이가 날 수 있다. 본 절에서는 3D-V 기법과의 압축률을 비

교하여 성능을 평가하고 있다. 움직임이 많은 foreman 비디오와 상대적으로 적은 carphone 비디오에 대해서 실험을 하였으며, <표 3>에서는 실험을 통해서 나타난 결과 값을 나타내고 있다.

<표 3> FS 3D-WT와 3D-V 기법의 압축률 비교

	FS 3D-WT	3D-V
Foreman 17~32	74.3%	73.7%
Foreman 161~176	75.1%	74.3%
Carphone 1~16	80.8%	78.3%
Carphone 145~160	88%	83.9%

(그림 14)에서는 <표 3>에서 나와 있는 압축률의 결과를 그래프로 나타내고 있다. 그림을 통해서도 알 수 있듯이, FS 3D-WT 기법은 두가지의 경우 모두에 있어 3D-V 기법보다 성능이 향상되었다. 그리고 움직임이 상대적으로 작은 carphone 비디오에서는 더 향상된 성능을 보였음을 확인할 수 있었다.

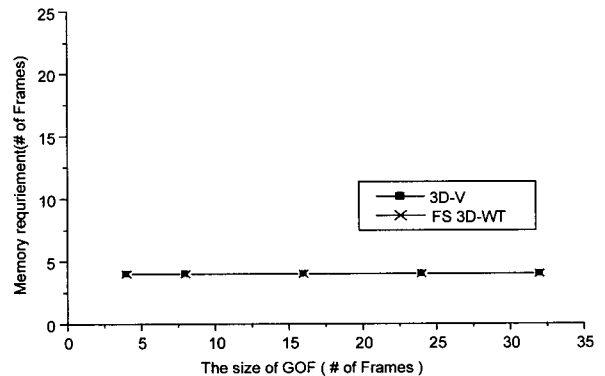
실험에 사용된 프레임은 움직임에 따른 부호화 결과의 확실성을 방지하기 위하여 비디오의 임의의 프레임 구간을 선별하여 얻어진 16개의 프레임을 통해 실험하였다. 그리고 실험의 파라미터로 사용된 압축률은 <표 3>에서와 같이 시간 분해를 통해서 압축된 데이터의 양이 시간 분해 전에 입력된 데이터의 어느 정도를 차지하는 지를 이용하여 나타내었다.

4.2.3 부호기 측에서의 메모리 사용량

3D-WT 기법은 움직임 예측 없이 여러 장의 프레임을 입력으로 받아서 시간적인 분해를 하기 때문에, 기존의 부호기가 필요로 하는 메모리보다 많은 양을 요구하게 된다.

GOF 단위의 모든 프레임을 받던 이전의 기법들이 GOF의 크기가 커짐에 따라 메모리 사용량이 늘어나는 반면, 본 논문에서 제안하는 기법은 3D-V기법과 마찬가지로 4개의 프레임을 작업단위로 처리하기 때문에 GOF의 크기에 상관없이 요구하는 메모리 사용량은 (그림 15)와 같이 항상 일

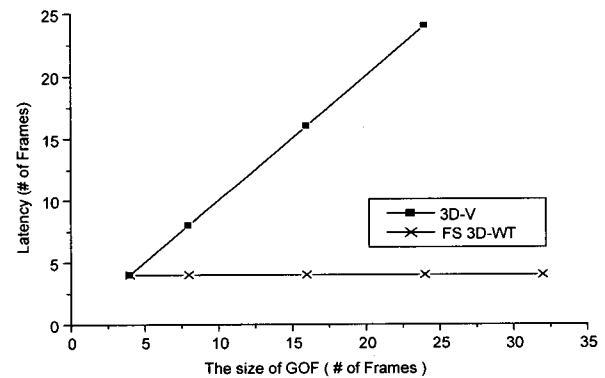
정하게 4프레임으로 유지됨을 알 수 있다.



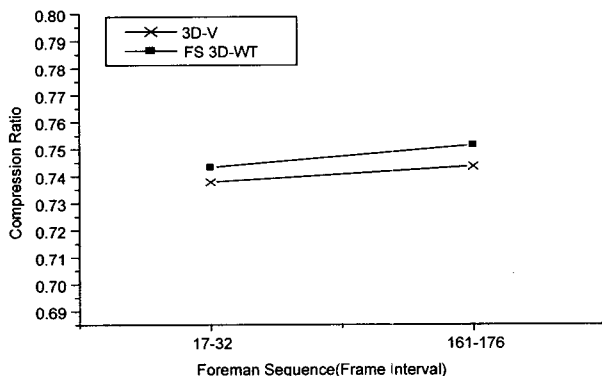
(그림 15) GOF 크기에 따른 메모리 요구사항 비교

4.2.4 수신 측에서의 지연시간

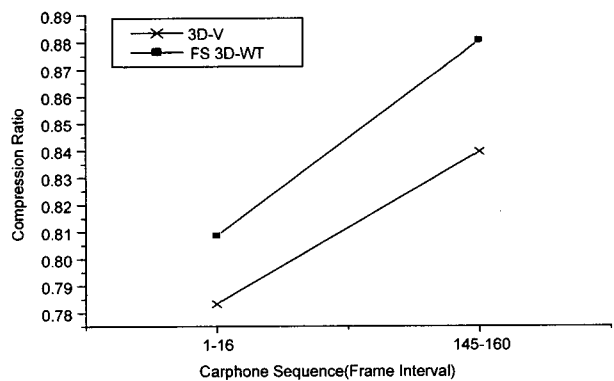
앞에서도 언급했듯이 3D-WT 기법은 GOF 단위로 부호화 작업이 이루어지기 때문에, 수신 측에서의 지연시간도 GOF가 모두 부호화 되어 전송될 때까지 기다려야만 하는 단점을 가지고 있었다. 하지만, 본 논문에서 제안된 기법은 (그림 16)에서와 같이 효율적인 부호화 알고리즘을 통해서 부호화할 때의 메모리 사용량뿐만 아니라, 수신 측에서의 지연시간도 부호화 작업단위인 4프레임으로 고정되어 사용



(그림 16) GOF 크기에 따른 수신 측에서의 지연시간 비교



(a) Foreman Sequence의 압축률 비교



(b) Caphone Sequence의 압축률 비교

(그림 14) 비디오의 움직임 특성에 따른 압축률 비교

될 수 있다. 한편, 3D-V 기법이 메모리 사용에서는 GOF의 크기에 상관없이 4프레임으로 고정적인 사용량을 보이지만 수신 측에서 야기되는 지연시간은 GOF에 따라 증가함을 볼 수 있다. 본 논문에서 제안한 FS 3D-WT 기법은 3D-V 기법과 거의 비슷한 계산 처리 시간으로 높은 압축률과 수신 측에서의 짧은 지연시간을 가능하게 했다. 또한, 부호기 측에서의 메모리 사용량과 수신 측에서의 지연시간에서는 GOF 크기에 상관없이 확장이 가능한 구조를 가지고 있어, 통신환경에 따라 압축률을 보다 더 효율적으로 조절할 수 있다는 장점을 가지고 있다.

5. 결론 및 향후과제

현재 일반화되어 있는 비디오 코딩 기법은 한 프레임씩을 입력으로 해서 DCT나 웨이블릿을 통해 변환시킨 후, 시간에 따라 연속적인 프레임들을 움직임 예측을 통해서 압축하여 부호화한다. 하지만, 움직임 예측이 가지는 복잡한 계산과정은 부호기의 전체적인 복잡도를 높이는 큰 단점으로 지적되어 왔고, 이는 무선 디바이스에서의 서비스를 위한 경량코덱으로의 적용가능성을 떨어뜨린다. 이러한 이유에서 움직임 예측 없이 연속적인 프레임을 부호화 할 수 있는 3D-WT 기법이 소개되었다. 하지만 3D-WT 기법도 부호기 측에서의 메모리 사용량이나 수신 측에서의 지연시간 등 여러 가지 문제점이 제시되었고 이를 해결하기 위한 방안들이 연구되었다.

본 논문에서는 메모리 요구사항과 수신 측의 지연시간에 대한 3D-WT 기법들의 단점을 개선할 수 있는 FS(Fast playable and Scalable) 3D-WT 기법을 제안하였다. 그리고 이 기법을 통하여 무선 디바이스에서 사용할 수 있는 경량코덱에 대한 적용 가능성을 높였다. 부호화할 때의 메모리 요구사항, 수신 측에서 재생을 위해 필요한 지연시간, 부호화 처리 시간, 그리고 압축률을 실험을 통하여 기존의 기법과 비교·논의하였다. 실험 결과 FS 3D-WT 기법은 기존의 기법들이 가지던 단점을 보완하여 좀 더 향상된 능력을 보여주고 있다. 특히 부호화할 때의 GOF 크기에 확장성을 제공함으로써 서비스의 효율성을 높일 수 있다.

하지만, 실제적인 서비스를 위해 전체 비디오 코딩 과정을 통한 압축률과 에러에 대한 강인성 측면에서 검증되어야 할 부분들이 남아 있다. 따라서 향후에는 움직임 예측을 통한 비디오 코딩 기법과의 데이터 압축률 비교를 통해 실제 적용 가능성을 알아볼 것이다. 그리고 시간적인 분해를 통해 압축된 프레임 정보가 네트워크를 통해서 전송될 때 발생하는 에러에 대해 어느 정도의 강인성을 지니고 있는지 조사하고 에러에 대해서도 강인성을 지닐 수 있도록 보완해야 할 것이다.

참 고 문 헌

[1] Video codec for audiovisual services at p×64kbit/s, CCITT Recommendation H.261, 1990.

[2] Video Coding for Low Bit Rate Communication, ITU-T Recommendation H.263, Jan., 1998.

[3] coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s MPEG1, ISO/IEC 11172-2, 1994.

[4] "MPEG-2 video," ITU-T Recommendation H.262-ISO/IEC 13818-2, Jan., 1995.

[5] Coding of audio-visual Objects, Part 2 Visual, Amendment 4 : Streaming Video Profile, ISO/IEC 14496-2/FPDAM4, July, 2000.

[6] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform," IEEE trans. Image Processiong, Vol.5, pp.205-220, Apr., 1992.

[7] S. Mallat, "A Theory for multiresolution signal decomposition : The wavelet representation," IEEE Trans. Pattern Anal. Machine Intell., Vol.11, pp.674-693, July, 1989.

[8] Zixiang Xiong, Ramchandran K., Orchard M.T. and Ya-Qin Zhang, "A comsparative study of DCT- and wavelet-based image coding," IEEE Trans. Circuits Syst. Video Technol., Vol.6, pp.692-695, Aug., 1999.

[9] Wing-Kuen Ling and Bing Zeng "A novel method for blocking effect reduction in DCT-coded images," Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999. IEEE International Symposium on, Vol.4, pp.46-48, Jul., 1999.

[10] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Trans. Signal Processing, Vol.41, pp.3445-3463, Dec., 1993.

[11] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits Syst. Video Technol., Vol.6, pp.243-250, June, 1996.

[12] Aksay, A., Bozdagi, G., Akhan, M. B., Temizel, A. and Duhamel, P. B. D., "Motion wavelet compression," Time-scale and Time-Frequency Analysis and Applications, IEE Seminar on, 2000.

[13] Ya-Qin Zhang, Sohail Zafar, "Motion-Compensated Wavelet Transform Coding for Color Video Compression," IEEE Trans. On Circuit and Systems for Video Technology, Vol. 2, No.3, Sep., 1992.

[14] Ming Fai Fu, Au, O. C. and Wing Cheong Chan, "Novel motion compensation for wavelet video coding using overlapping," Acoustics, Speech, and Signal Processing, IEEE International Conference on, Vol.4 pp.3421-3424, 2002.

[15] H.W. Park and H. S. Kim, "Motion estimation using low-band-shift method for wavelet-based moving-picture coding," Image Proc, IEEE Trans., Vol.9, pp.577-587, Apr., 2000.

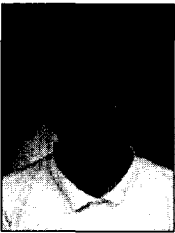
[16] S. Kim, T. Aboulnasr and S. Panchanathan, "Adaptive multiresolution motion estimation techniques for wavelet-based video coding," Proc. SPIE Visual communications Image Processing, Vol.3309, pp.965-974, Jan., 1998.

[17] S. Kim, S. Ree, J. G. Jeon and K. T. Park, "Interframe cod-

[1] Video codec for audiovisual services at p×64kbit/s, CCITT

ing using two-stage variable block-size multiresolution motion estimation and wavelet decomposition," IEEE Trans. Circuits Syst. Video Technol. Vol.8, pp.399-410, Aug., 1998.

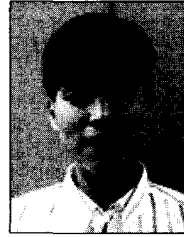
- [18] Salvarlo J. and Sousa L., "Video Coding by using the 3D Zero-Tree approach in the Wavelet transform domain," Digital Signal Processing, 14th International Conference on, Vol.2, pp.683-687, 2002.
- [19] E. Moyano, F. J. Quiles, A. Garrido, L. Orozco-Barbosa and J. Duato, "Efficient 3D Wavelet Transform Decomposition For Video Compression," Digital and Computational Video, 2001. Proceedings. Second International Workshop on, pp. 118-125, 2001.
- [20] C. I. Podilchuk, N. S. Jayant and N. Farvardin, "Three-dimensional subband coding of video," IEEE Trans. Image Processing, Vol.4, pp.125-139, Feb., 1995.
- [21] David Taubman, Avidh Zakhor, "multirate 3-D Subband coding of Video," IEEE Trans. on Image Processing, Vol.3, No.5, Sep., 1994.
- [22] Beong-Jo Kim, Zixiang Xiong, William A. Pearlman, "Low Bit-Rate Scalable Video Coding with 3-D Set Partitioning in Hierarchical Trees(3-D SPIHT)," IEEE Trans. On Circuits and system for Video Technology, Vol.10, No.8, Dec., 2000.



이 승 원

e-mail : swlee@pusan.ac.kr
 1997년 부산대학교 전자계산학과(학사)
 1999년 부산대학교 대학원 전자계산학과
 (이학석사)
 1999년~현재 부산대학교 대학원 전자
 계산학과(박사과정)

관심분야 : 트랜스코딩, 유비쿼터스 컴퓨팅, 멀티미디어 스트리밍



김 성 민

e-mail : morethannow@pusan.ac.kr
 2001년 부산대학교 전자계산학과(학사)
 2003년 부산대학교 전자계산학과 석사
 (이학석사)
 2003년~현재 부산대학교 컴퓨터공학과
 박사과정

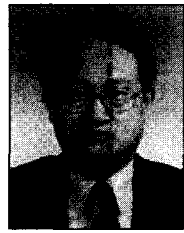
관심분야 : 트랜스코딩, 유비쿼터스 컴퓨팅, 멀티미디어 스트리밍



박 성 호

e-mail : shpark@pusan.ac.kr
 1996년 부산대학교 전자계산학과(학사)
 1998년 부산대학교 대학원 전자계산학과
 (이학석사)
 2002년 부산대학교 대학원 전자계산학과
 (이학박사)

2002년~현재 부산대학교 정보전산원 조교수
 관심분야 : 트랜스코딩, 멀티미디어 캐싱, 이동통신



정 기 동

e-mail : kdchung@melon.cs.pusan.ac.kr
 1973년 서울대학교 졸업(학사)
 1975년 서울대학교 대학원(석사)
 1986년 서울대학교 대학원 계산통계학과
 (이학박사)
 1990년~1991년 MIT, South Carolina
 대학 교환 교수

1995년~1997년 부산대학교 전자계산소 소장
 1978년~현재 부산대학교 전자계산학과 교수
 관심분야 : 병렬처리, 멀티미디어 시스템, 멀티미디어 통신