Daubechies D4 필터를 사용한 시간가변(time-varying) 볼륨 데이터의 압축

Compression of time-varying volume data using Daubechies D4 filter

허영주 Youngju Hur*, 이중연 Joongyoun Lee**, 구기범 Geebum Koo***
한국과학기술정보연구원(KISTI)*, **, ***

요약 The necessity of data compression scheme for volume data has been increased because of the increase of data capacity and the amount of network uses. Now we have various kinds of compression schemes, and we can choose one of them depending on the data types, application fields, the preferences, etc. However, the capacity of data which is produced by application scientists has been excessively increased, and the format of most scientific data is 3D volume. For 2D image or 3D moving pictures, many kinds of standards are established and widely used, but for 3D volume data, specially time-varying volume data, it is very difficult to find any applicable compression schemes.
In this paper, we present a compression scheme for encoding time-varying volume data. This scheme is aimed to encoding time-varying volume data for visualization. This scheme uses MPEG's I- and P-frame concept for raising compression ratio. Also, it transforms volume data using Daubechies D4 filter before encoding, so that the image quality is better than other wavelet-based compression schemes.
This encoding scheme encodes time-varying volume data composed of single precision floating-point data. In addition, this scheme provides the random reconstruction accessibility for an unit, and can be used for compressing large time-varying volume data using correlation between frames while preserving image qualities.

핵심어: Daubechies, encoding, time-varying volume data, wavelet

1. 서론

데이터 압축 기술은 사용되는 데이터의 용량이 증가하면서, 그리고 네트워크를 통한 데이터 전송량이 늘어나면서 점점 더 그 필요성이 증가하고 있다. 데이터 압축 기술은 여러 다양한 분야에서 활용되고 있으며, 특히 그래픽스 분야에서 사용되는 다양한 형태의 멀티미디어 데이터는 그 용량이 매우 방대하기 때문에 이런 방대한 데이터를 효과적으로 압축하기 위한 다양한 방법이 고안돼 왔다. 상대적으로 사용량이 많은 2차원 이미지나 3차원 동영상에는 JPEG이나 MPEG같은 표준이 제정, 활용되고 있다.

그러나 3차원 볼륨 데이터의 압축에 대한 연구는 그다지 활발하지 않았다. 일반적으로 볼륨 데이터는 여러 과학 분야에서 시뮬레이션의 결과로 산출된다. 볼륨 데이터는 컴퓨팅 기술의 발전에 힘입어 점점 더 용량이 방대해지는 추세에 있으며, 종종 컴퓨터 주기억장치(RAM)에 로딩하는 것이 불가능한 크기의 데이터가 산출되곤 한다. 이런 볼륨 데이터는 분류 방식에 따라 다양한 종류로 나눌 수 있다. 그 중에는 시간축에 대해 여러 개의 볼륨 데이터가 존재하는지 여부를 놓고 고정(static) 데이터와 시간 가변(time-varying) 데이터로 나누는 방법도 있다. 시간가변 볼륨 데이터는 시간에 따라 변하는 수치를 나타내는 데이터로, 용량이 큰 3차원 볼륨 데이터가 시간 단계(time step)만큼 여러개 나타나는 형태의 볼륨 데이터이며, 따라서 그 크기도 상당히 방대하다. 볼륨 데이터는 데이터의 종류의 응용목적에 적합한 압축 기술을 적용, 처리해야 하며, 따라서 시간가변 볼륨 데이터에는 기존의 압축기법과는 다른, 별도의 압축 기법이 적용돼야 한다.

본 논문에서는 시간가변 볼륨 데이터에 대한 손실(lossy) 압축기법을 제안한다. 이 기법은 시간가변 볼륨 데이터의 가

시화를 대상으로 하며, 기존 MPEG의 I-프레임과 P-프레임의 개념을 도입, 볼륨 데이터의 시간 단계간의 상호 연관성을 이용함으로써 압축률을 높였다. 또, Daubechies 웨이블릿 필터를 이용, 데이터를 변환하기 때문에 Haar 웨이블릿을 사용한 기존 압축 방식에 비해 동일한 압축률에서 상대적으로 우수한 복원 화질을 나타낸다. 이 압축방식은 부동소수점(single precision floating-point) 데이터로 구성된 시간가변 볼륨 데이터를 대상으로 하도록 설계됐으며, 렌더링 코드와의 연계를 고려, 블록 단위의 무작위 추출(random access)이 가능하다.

2. 관련 연구

일반적으로는 인코딩에 앞서 데이터를 다른 영역으로 변환해서 데이터의 분포 범위를 좁히거나 유효 데이터의 개수를 줄이면 압축률을 높일 수 있다. 즉, 데이터의 중복도가 높아지거나 분포 범위가 줄어들면 보다 높은 압축률을 얻을 수 있는 것이다. 따라서 인코딩에 앞선 데이터 변환은 압축에 있어서 필수적인 기술이다. 이런 용도로 사용되는 데이터 변환 기법은 매우 다양하며 DTC(Discrete Cosine Transform) 변환이나 웨이블릿 변환 기법이 많이 사용된다. 압축에 많이 사용되는 웨이블릿은 Haar 웨이블릿[12]인데, 이 함수는 비교적 간단한 연산으로 압축 효과를 높일 수 있다는 장점이 있으며 연산이 빠르기 때문에 복원속도가 중요시되는 실시간 응용분야에 많이 활용된다.

웨이블릿 변환에는 Haar 웨이블릿 외에도 여러 다양한 변환이 존재한다. 그 중 Daubechies 웨이블릿[4][6]은 Haar 웨이블릿보다 복원 데이터의 손실율이 적다. Daubechies 웨이블릿은 2개에서 20개의 계수를 사용한 다양한 데이터 변환 방식을 제공하는데, 그중 4개의 계수를 사용한 D4 방식은 계산이 비교적 간단하면서도 복원 화질의 손상도가 낮기 때문에 많이 사용된다.

2D 이미지 압축에서는 데이터 변환에 웨이블릿을 많이 사용한다. [15]에서는 웨이블릿 계수의 계층 구조를 이용한 인코딩 방식을 소개했으며 [14]에서는 데이터의 중요도에 따라 인코딩의 우선순위를 달리하는 EBCOT이라는 압축 방식을 소개했다. EBCOT은 JPEG2000에서 표준으로 채택, 사용되고 있다.

2D 이미지 압축과 함께 가장 많이 사용되는 멀티미디어 형식인 동영상을 압축하는데 가장 많이 사용하는 방식은 MPEG[9][13]이다. MPEG은 각 프레임간의 상호 연관성과 인간의 인지 시스템의 허점을 이용해서 데이터를 압축한다. [10]과 [11]에서는 프레임간 상호 연관성을 이용하는 MPEG의 특성을 볼륨 데이터에 적용함으로써 시간가변 볼륨 데이터를 효과적으로 압축하는 방법을 제안했다. 여기에서는 MPEG의 I-프레임과 P-프레임 개념을 이용해서 볼륨 데이터를 압축한다.

[7]에서는 볼륨 데이터를 애니메이트해서 여러개의 프레임을 생성한 뒤, 이 프레임에 Haar, Daubechies D4, CDF 등의 여러 다양한 웨이블릿 필터를 적용하고 LZH나 산술(arithmetic)코딩과 같은 기존 인코딩 방식으로 압축했다. [8]에서는 각각의 시간 단계별 볼륨 데이터의 상호 연관성을 octree 형태의 자료구조에 저장, 압축하는 방법을 제시했다. [6]에서는 vector quantization을 이용, 시간가변 볼륨 데이터를 압축하는 방법을 제안했다.

시간가변 볼륨 데이터에 상대적 개념인 고정 볼륨 데이터에 대한 인코딩 기법은 [1]과 [2]에서 제안한 바 있다. 이 방식은 [3]의 제로비트 인코딩 방식을 응용, Haar 웨이블릿 대신 Daubechies D4 웨이블릿 필터를 사용하고 압축 대상을 부동 소수점 데이터로 확장했다. 본 논문에서는 이 인코딩 방식을 확장, 부동 소수점 데이터로 구성된 시간가변 볼륨 데이터에 적용 가능한 인코딩 기법을 제안한다. 이 압축 기법은 Daubechies D4 필터를 사용한, 시간가변 볼륨 데이터에 특화된 인코딩 기법이며 [10]이나 [11]같이 MPEG의 프레임간 연관성을 이용해서 데이터를 인코딩한다. 이 기법은 Haar 필터를 사용한 기법에 비해 더 나은 화질을 제공하며 블록 단위의 무작위 추출을 지원하기 때문에 볼륨 렌더링 코드에 직접 삽입해서 사용하는 것이 가능하다.

3. 볼륨 데이터 인코딩 및 디코딩

시간가변 볼륨 데이터를 구성하는 볼륨 데이터간 상관관계는 그림 1과 같다. 각 볼륨 데이터는 MPEG에서와 같이 I-볼륨과 P-볼륨으로 구분할 수 있는데, I-볼륨은 다른 볼륨을 참조하지 않고 자체적으로 복원이 가능한 볼륨을 나타내고 P-볼륨은 다른 볼륨을 참조해야만 복원이 가능한 볼륨을 나타낸다. 그림 1에서는 volume 1이 I-볼륨이고 다른 볼륨은 모두 P-볼륨이다.

볼륨 데이터를 변환하면 변환 과정에서 본래 데이터와 오차가 발생하게 된다. 이 오차를 무시한 채 무조건 원래 볼륨과 다른 볼륨과의 상관관계를 구하게 되면 P-볼륨이 거듭해나갈수록 데이터에 쌓이는 오차는 점점 더 커진다. 이런 현상을 방지하기 위해 그림1과 같은 복잡한 과정이 필요한 것이다.

모든 볼륨에 대해서는 블록화 과정과 웨이블릿 변환, 그리고 quantization 과정을 모두 거친, 인코딩 직전의 볼륨 데이터를 다시 de-quantization 과정과 웨이블릿 복원 과정을 거쳐 복원한 뒤, 그 복원된 볼륨 데이터를 다른 볼륨 데이터와의 상관관계를 구하기 위한 참조 볼륨으로 사용한다. I-볼륨의 경우에는 본래 볼륨 데이터를 이 과정을 거친 뒤, 다른 P-볼륨을 생성하기 위한 참조 데이터로 사용한다. P-볼륨의 경우에는 조금 더

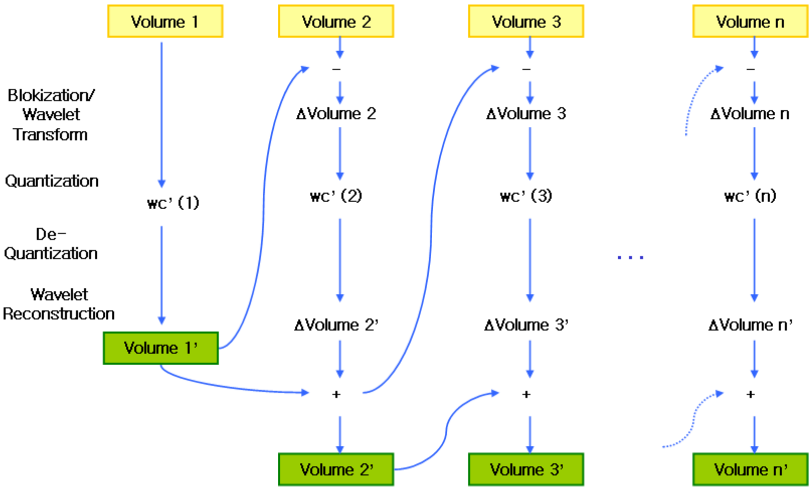

그림 1. 시간가변 데이터를 구성하는 각 프레임의 상관관계

복잡한 과정이 필요하다. 우선 현재 압축하고자 하는 볼륨의 현재 데이터와 복원한 이전 볼륨간의 차를 구한 뒤, 이 차이값을 변환, quantization을 수행한다. 이렇게 quantization까지 끝난 데이터는 바로 인코딩을 수행할 수 있다. 그러나 이 데이터를 다음 P-볼륨 생성을 위한 참조 데이터로 이용하려면 한차례 복원 과정을 거친 뒤, 이전 프레임의 볼륨값에 더해서 사용한다. 이런 식으로 각 프레임을 구성하면 오차를 최소화할 수 있다.

2.1 볼륨 데이터의 인코딩 과정

그림 1에서는 시간가변 볼륨 데이터를 구성하는 모든 볼륨 데이터간의 상관관계를 보여줬다. 그림 2에서는 시간가변 볼륨데이터를 구성하는 각각의 볼륨 데이터를 인코딩하고 디코딩하는 과정을 볼 수 있다.

볼륨 데이터는 볼륨의 종류 즉 I-볼륨인지, 혹은 P-볼륨인지 여부에 따라 전처리 과정이 달라진다. I-볼륨은 자체 데이터를 그대로 변환해서 인코딩하지만 P-볼륨인 경우에는 이전 볼륨과의 상호관계를 먼저 구한 뒤, 변환과 인코딩 과정을 거치기 때문이다. 이와 같이 P-볼륨의 경우에는 이전 볼륨과의 상호 관계를 구하는 과정이 전처리 과정으로 추가된다.

전처리가 끝난 (I-볼륨의 경우에는 전처리 과정이 없다.) 볼륨 데이터는 블록 단위로 저장된다. 블록의 크기는 8 x 8 x 8, 16 x 16 x 16 등으로 다양하게 설정할 수 있는데, 여기에서는 실험에 사용할 볼륨 데이터의 해상도와 D4 웨이블릿 필터의 특성을 고려, 8 x 8 x 8 크기의 블록을 사용했다.

이렇게 블록 단위로 저장된 볼륨 데이터는 3차원 웨이블릿 변환 과정을 거치게 된다. 웨이블릿 변환은 2단계로 이뤄지게 되며, 경우에 따라서는 변환된 계수중 비교적 중요도가 낮은 계수를 절삭(truncation)함으로써 압축률을 높이기도

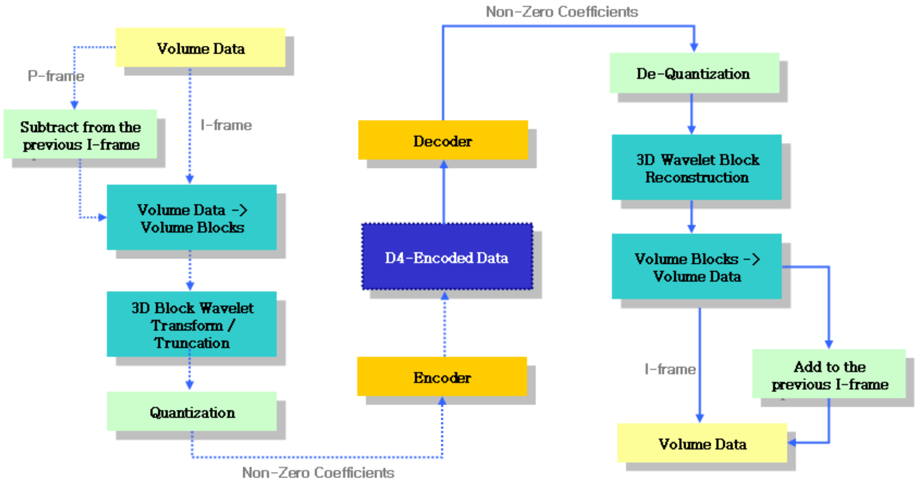

그림 2. 볼륨 데이터의 인코딩 및 디코딩

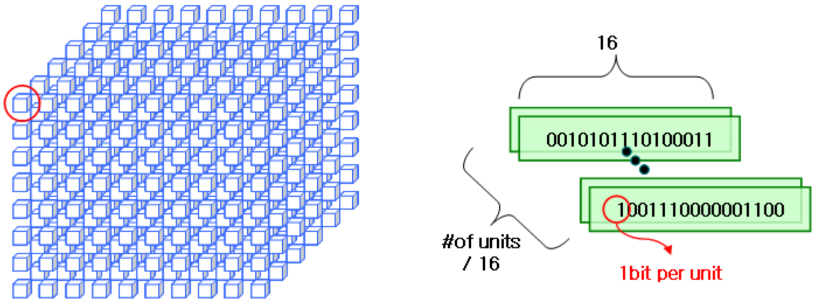

그림 3. 유닛 플래그 테이블

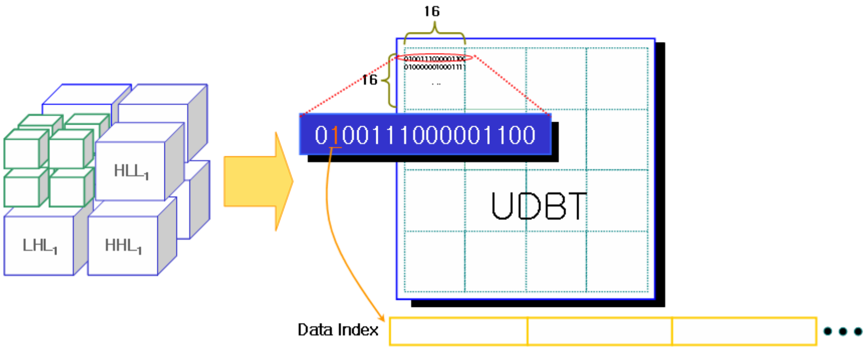

그림 4 인코딩된 블록 데이터

한다. 이렇게 계수를 절삭하는 방법과 절삭되는 계수의 비율에 따라 압축률은 차이를 보인다.

웨이블릿 변환이 끝난 계수는 quantization 과정을 거치면서 한번 더 압축이 이뤄진다. 이 quantization 과정을 통해 4-바이트 크기의 부동 소수점 데이터는 2-바이트 크기의 quantization 인덱스로 변환되며, 이 과정에서 압축률은 크게 늘어난다. Quantization 방식에는 모든 구간을 동일한 크기로 나눠서 인덱스를 부여하는 uniform quantization과 데이터 분포도가 높은 부분을 더 좁은 구간으로 나누는 non-uniform quantization이 존재하는데, 여기에서는 uniform quantization 방식을 사용했다. 이 과정이 끝난 인덱스는 인코딩 과정을 거쳐 최종 압축 데이터가 생성된다. 인코딩 과정은 뒤에서 자세히 설명하기로 한다.

2.2 인코딩 데이터의 구조

본 논문에서 제안한 인코딩 방식에서는 유효 블록의 위치 저장에 그림 3과 같은 유닛 플래그 테이블(Unit Flag Table)을 사용한다.

유닛 플래그 테이블은 유효 블록의 위치를 나타내는데 사용되는 자료구조로, 디코딩시에는 의미 있는 블록의 위치를 찾아가는 용도로 사용된다. 블록 하나는 1비트로 표시되며, 의미 있는 데이터를 포함하는 블록은 1로, 그렇지 않은 블록은 0으로 표시된다. 즉, 이 테이블에 0으로 설정된 비트에 위치한 블록 내의 웨이블릿 계수는 모두 0이기 때문에 이 블록에서는 블록 내 계수를 검색하거나 복원을 위한 계산 작업을 따로 수행할 필요가 없다.

유효 블록에 존재하는 계수는 그림 4와 같은 자료구조로 저장한다. 즉, 그림 4의 자료구조는 유닛 플래그 테이블에서 유효한 블록으로 표시된 블록에 대해서만 존재한다.

UDBT(Unit Data Bit Table)는 블록 내에서 유효 계수의 위치를 표시하는 자료 구조로, 해당 위치의 웨이블릿 계수가 유효한지 여부를 나타낸다. 즉, 디코딩하고자 하는 웨이블릿 계수 위치에 해당하는 비트가 0이면 그 계수는 0이기 때문에 더 이상 다른 과정을 수행할 필요가 없다. 그러나 1이라면 해당 위치에 계수가 존재한다는 의미이기 때문에 블록 내에서 유효 계수로써의 순서를 센 뒤, 이 값을 가지고 데이터 인덱스에서 quantization 인덱스를 검색해서 데이터 복원에 활용하면 된다. 즉, 유효블록 내에 존재하는 유효 데이터의 위치는 UDBT에, 그리고 실제 quantization 인덱스는 2바이트 크기의 데이터 인덱스에 저장되는 것이다.

2.3 디코딩

인코딩된 볼륨 데이터를 디코딩하는 과정은 그림 2의 후반부에서 볼 수 있다.

디코딩은 블록 단위로 수행할 수 있으며, 인코딩과 마찬가지로 I-볼륨과 P-볼륨의 처리에 약간의 차이가 있다. 볼륨 데이터를 복원하려면 I-볼륨이건 P-볼륨이건 일단 데이터를 디코딩해서 웨이블릿 계수를 채워놓아야 한다. 디코딩 과정에서는 유닛 플래그 테이블을 사용, 유효 블록에 대해서

만 디코딩 과정을 수행하면 된다. 유효 블록을 복원하려면 블록 내에서 UDBT를 참조, 유효 계수의 위치를 파악하고, 데이터 인덱스 스트림에서 해당 유효 계수에 대해 실제 계수를 검색할 수 있는 quantization 인덱스를 검색하면 된다. 이렇게 검색된 인덱스는 quantization 테이블을 참조, 다시 웨이블릿 계수로 복원되고, 이렇게 채워진 블록에 대해 3차원 웨이블릿 복원 수식을 적용하면 한 블록에 대한 복원 과정이 모두 완료된다.

블록화된 데이터를 모두 복원하고 나면 이 블록 데이터는 한데 합쳐져 하나의 볼륨 데이터를 구성한다. I-볼륨의 경우에는 본래 데이터가 복원되므로 별도의 복원 과정이 필요없이 바로 사용할 수 있지만, P-볼륨의 경우에는 이전 볼륨과의 차이값이 복원되기 때문에 복원된 이전 볼륨의 데이터값에 차이값을 더해서 본래 데이터를 구해야 한다. 이런 식으로 복원된 볼륨 데이터는 볼륨 렌더링을 과정을 통해 이미지화하는 것이 가능하다.

4. 결과

본 논문에서 제시한 방법을 적용해서 인코딩한 결과는 표 1에서 볼 수 있다. 실험에는 3,600 x 2,400 x 40 해상도를 가진 볼륨 데이터를 사용했다. 이 데이터는 대양의 온도 (ocean temperature) 변화를 나타내는 시간가변 볼륨 데이터로 부동 소수점 데이터로 구성돼 있다. 시간가변 볼륨 데이터를 구성하는 전체 시간 단계는 12단계이며 각각의 시간 단계를 나타내는 볼륨 데이터의 크기는 1.3G다. 따라서 전체 데이터 크기는 1.3 x 12 = 15.6G가 된다.

표 1에서 볼 수 있듯이 각각의 시간 단계는 평균 약 50%의 압축률을 보인다. 또, I-볼륨보다는 P-볼륨의 압축률이 상대적으로 약간 높은 현상을 볼 수 있으며, SNR과 PSNR 수치를 비교했을 때, 화질 저하는 거의 없는 것으로 결론지을 수 있다.

실험에 사용한 데이터는 볼륨 전체가 조밀하게 차있는 경우에 해당되며, 만약 상대적으로 빈공간이 많은 데이터를 사용할 경우에는 압축률은 훨씬 더 높아질 것이다.

표 1의 결과는 절삭 과정을 생략하고 웨이블릿 계수를 모두 사용한 경우, 즉 100%의 계수를 사용한 경우의 결과를 나타내며, 절삭 과정을 추가하면 압축률은 훨씬 높아질 것이다. 또, uniform quantization 대신 non-uniform quantization 기법을 사용하면 압축률을 높이는 동시에 보다 나은 화질을 얻는 것도 가능하다.

표 1. Ocean Temperature 데이터의 인코딩 결과

Volume Number	Ocean Temperature (3,600 x 2,400 x 40 = 1.3G)			
	RMSE	SNR	PSNR	Data Size(MB)
00 (I-볼륨)	0.532	60.91	66.70	610M
01	0.072	59.83	65.54	702M
02	0.072	59.83	65.55	402M
03	0.072	59.80	65.53	402M
04 (I-볼륨)	0.064	60.77	66.52	610M
05	0.072	59.72	65.48	702M
06	0.073	59.68	65.53	402M
07	0.073	59.68	65.53	402M
08 (I-볼륨)	0.064	60.80	66.76	610M
09	0.015	73.67	80.75	702M
10	0.016	72.70	79.56	402M
11	0.017	72.07	78.73	402M

4. 결론

본 논문에서는 Daubechies D4 웨이블릿 필터를 사용해서 시간가변 볼륨 데이터를 압축하는 인코딩 기법을 제안했다. 이 방식은 4바이트 크기의 부동소수점으로 구성된 시간가변 볼륨 데이터를 대상으로 하는 압축기법으로, D4 필터를 사용해서 데이터를 변환하고 MPEG의 I-프레임 P-프레임 개념을 사용, 볼륨 데이터간의 상호관계를 이용해서 압축률을 높인다.

향후에는 절삭 기법과 non-uniform quantization 기법을 추가, 압축률을 높임으로써 보다 효율적인 인코딩 기법으로 개선할 예정이다.

참고문헌

[1] 허 영주, 박 상훈, "Daubechies 웨이블릿 변환을 이용한 볼륨 데이터 압축", 제 24회 한국정보처리학회 추계학술발표대회, 제 12권 2호, 2006년 11월.

[2] 허 영주, 박 상훈, "Daubechies 웨이블릿 필터를 이용한 볼륨 데이터 인코딩", 정보처리학회논문지 제 13-A권 제 7호, 2006년 12월.

[3] C.Bajaj, I.Ihm, S.Park, "3D RGB compression for interactive applications", ACM Transactions on Graphics, Vol.20, pp.10-38, 2001.

[4] I.Daubechies, D4 Wavelet Transform, http://www.bearcave.com/misl/misl_tech/wavelet/daubechies/.

[5] I.Daubechies, Ten Lectures on Wavelets, SIAM, 1992.

[6] N.Fout, K.Ma, J.Ahrens, "Time-varying, multivariate volume data reduction", ACM Symposium on Applied computing", pp.1224-1230, 2005.

[7] S.Guthe, W.Straber, "Real-time decompression and visualization of animated volume data", Conference on Visualization, pp.349-256, October 2001.

[8] K.Ma, D.Smith, M.Shih, H.Shen, "Efficient encoding and rendering of time-varying volume data", ICASE Technical Report 98-22, NASA/CR-1998-208424, ICASE, Hampton, Virginia, 1998.

[9] D.Le Gall, "MPEG: A video compression standard for multimedia applications", Communications of the ACM, Vol.34, Issue 4, pp.46-58, 1991.

[10] B.Sohn, C.Bajaj, V.Siddavanahalli, "Feature based volumetric video compression for interactive playback", Proceedings of 2002 Symposium on Volume Visualization and Graphics, pp.89-96, 2002.

[11] B.Sohn, C.Bajaj, V.Siddavanahalli, "Volumetric video compression for interactive playback", Computer Vision and Image Understanding, Special Issue on Model-based and image-based 3D scene representation for interactive visualization, 96(3), pp.435-452, 2004.

[12] E.Stollnitz, T.DeRose, D.Salesin, "Wavelets for Computer Graphics: Theory and Applications", Morgan Kaufmann Publishers, 1996.

[13] P.Symes, Digital Video Compression, McGraw-Hill, 2004.

[14] D. Taubman, "High Performance Scalable Image Compression with EBCOT", IEEE Transactions on Image Processing, Vol.9, no.7, July 2000.

[15[C.Valens, EZW encoding, http://perso.wanadoo.fr/polyvalens/clemens/ezw/ezw.html

[16] C.Wang, J.Gao, H.Shen, "A multiresolution volume rendering framework for large scale time-varying data visualization", Proceedings of International Workshop in Volume Graphics, pp.11-19, 2005.