

계층적 깊이 영상의 깊이 정보 압축

윤승욱*, 김성열*, 김희순*, 윤국진**, 조숙희**, 안충현**, 호요성*

* 광주과학기술원 정보통신공학과

** 한국전자통신연구원 건파방송연구소

Compression of Depth Information in Layered Depth Image

Seung-Uk Yoon*, Sung-Yeol Kim*, Heesun Kim*,
Kug-Jin Yun**, Suk-Hee Cho**, Chung-Hyun Ahn**, and Yo-Sung Ho*

* Kwangju Institute of Science and Technology

E-mail: {suyoon, sykim75, hskim03, hoyo}@kjist.ac.kr

** Electronics and Telecommunications Research Institute

E-mail: {kgyun, shee, hyun}@etri.re.kr

요 약

본 논문에서는 계층적 깊이 영상(Layered Depth Image)의 깊이 정보 압축에 관해 연구하였다. 계층적 깊이 영상 데이터는 크게 8가지 구성요소로 분류할 수 있으며, 각각의 분류된 데이터에 대해 개별적인 압축 기법을 적용할 수 있다[1]. 특히, 기존의 방법을 개선해 깊이 데이터의 분산을 줄임으로써 데이터 변환을 수행할 때 이득을 얻을 수 있는 전처리 방법을 제안하였다. 제안된 알고리즘은 깊이 데이터를 이차원상의 점으로 보고 직선으로 연결한 후 직선과 각 점간의 최단거리를 이용하였다. 제안된 알고리즘을 통해 깊이 값의 분산을 줄이고, 결과적으로 데이터 변환의 효율을 증가시킬 수 있다.

Abstract

In this paper we propose an efficient pre-processing algorithm to compress depth information of Layered Depth Image(LDI). LDI data can be classified into 8 components: R, G, B, alpha, depth, distance, and splat index (Nx, Ny)[1]. We consider each depth value as a point in 2-D space. We compute the minimum distance between a straight line composed of previous two depth values and a current depth value. Then, we replace the current depth value with the minimum distance. The proposed algorithm make the variance of depth values small. Therefore, it increases the efficiency of transform and compression.

I. 서론

컴퓨터 그래픽스에 관한 연구가 시작된 이래로, 컴퓨터 그래픽스 기술을 이용해 사진과 같은 사실적인 영상을 얻는 일이 중요한 문제로 대두되어 왔다. 지난 수년간 많은 연구기관, 학교, 기업에서 삼차원 모델기반 렌더링 기법을 연구했으며, 그 결과 보다 현실감 있는 영상을 얻는데 유용한 모델링과 렌더링 도구들이 개발되었다. 그러나 이러한 도구를 이용해 복잡한 모델을 생성하기 위해서는 많은 노력과 숙련도 및 시간이 요구된다. 즉, 기존의 렌더링 기법은 모델링이 어렵고, 처리 시간이 많

이 걸린다는 단점을 가지고 있다. 이러한 문제점을 해결하기 위해 컴퓨터 그래픽스 분야에 새롭게 등장한 것이 바로 영상기반 렌더링 (Image Based Rendering) 기법이다. 영상기반 렌더링의 기본적인 아이디어는 여러 시점의 이차원 입력 영상을 이용하여 삼차원 공간의 임의 시점에서 본 영상을 생성하는 것이다. 이러한 접근 방식의 장점은 디스플레이 알고리즘이 적절한 자원을 요구한다는 점, 이차원 영상을 입력으로 사용하므로 생성하려는 영상의 복잡도와 무관하다는 점, 그리고 원하는 장면 생성을 위해 복잡한 삼차원 모델을 생성하는 것보다 영상이나 사진을 얻는 것이 훨씬 쉽다는 점이다. 따라서 상

대적으로 복잡도가 낮은 시스템을 통해 비슷한 화질의 영상을 생성해 낼 수 있다. 최근에는 이러한 영상기반 렌더링 기법 중에서 계층적 깊이 영상(Layered Depth Image)에 관한 연구가 많이 수행되고 있다. 계층적 깊이 영상은 여러 시점에서 생성된 깊이 영상을 합성하여 하나의 데이터 구조로 만든 것으로, 임의 시점의 영상을 간단한 워핑함수를 사용하여 쉽게 생성할 수 있다. 반면 한 장의 영상이 다시점 깊이 데이터를 포함하므로 용량이 매우 큰 단점이 있다. 현재 이러한 계층적 깊이 영상 데이터의 압축 및 부호화에 관한 연구는 세계적으로도 초보적인 실정이다. 따라서 본 논문에서는 계층적 깊이 영상 데이터의 깊이 정보 압축에 관해 연구하였다.

II. 영상기반 렌더링 기법

1. 영상기반 렌더링 기술동향

영상기반 렌더링은 최근에 영상 합성 기법을 위한 고전적인 기하학 정보 기반 기술에 대체기술로서 각광을 받고 있다. 영상기반 렌더링 기술은 크게 기하학 정보가 없는 렌더링, 암시적 기하학 정보가 있는 렌더링, 그리고 명확한 기하학 정보가 있는 렌더링으로 분류할 수 있다. 기하학 정보가 없는 렌더링 방법은 플레오틱 함수(Plenoptic function)의 특성을 이용하여 장면을 재현하는 것으로, 빛 정보와 텍스처 정보만을 이용한다. 라이트 필드(Light field) 렌더링과 루미그래프(Lumigraph) 렌더링 기법이 이 범주에 속한다. 암시적 기하학 정보가 있는 렌더링 방법은 새로운 시점에서의 장면을 재현하기 위해 입력 영상들의 위치적 관련성을 이용하는 것으로, 암시적이란 용어를 사용한 이유는 기하학적 정보가 직접적으로 이용되지 않는다는 의미에서이다. 삼차원 정보는 투영 기법 계산을 통해 얻어지고 이 정보는 새로운 장면을 위해 사용된다. 시점 모핑(View morphing) 기법이 이 범주에 속한다. 마지막으로, 명확한 기하학 정보가 없는 렌더링 방법은 직접적으로 깊이 정보와 같은 삼차원 모델 정보를 사용하며, 전통적인 텍스처 맵핑 방법이 이 범주에 속한다. 특히, 계층적 깊이 영상을 이용한 방법이 최근 많이 연구되고 있다. 그림 1은 다양한 영상기반 렌더링 기술을 나타낸다.

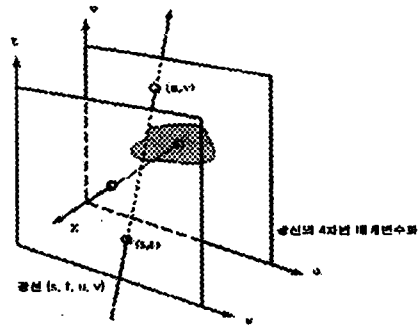
기하정보가 없는 렌더링 암시적 기하정보가 있는 렌더링 명시적 기하정보가 있는 렌더링

라이트 필드	루미그래프	계층적 깊이 영상	텍스처 맵 모델
영상 모자이크	시점 모핑	3차원 워핑	

[그림 1] 영상기반 렌더링 기술 분류

2. 루미그래프

루미그래프 표현은 사차원 플레오틱 함수를 이용하여 복잡한 모델의 완벽한 형상을 부호화하고, 모델의 복잡도와는 독립적인 속도로 객체를 표현할 수 있도록 한다 [2]. 플레오틱 함수는 위치와 방향을 표현하는 5가지 변수의 함수이다. 만약 공기가 투명하다고 가정하면 빈 공간을 통하는 광선의 광휘는 상수로 남는다. 만약 경계한 객체의 불투명한 외면에 남아있는 빛으로 관심을 더 제한한다면, 객체를 둘러싼 표면에만 플레오틱 함수의 값을 표현 하면 된다. 공간의 어떤 점으로부터 정육면체의 표면상의 어떤 공간을 통과하는 광선을 추적함으로써 광선이 지닌 광휘를 결정할 수 있다. 그러므로 객체에 기인하는 플레오틱 함수는 사차원으로 줄어들 수 있으며, 루미그래프에서는 이 사차원 함수를 (s, t, u, v) 의 매개변수로 표현한다. s, t, u, v 는 $s-t$ 평면과 $u-v$ 평면을 나타내는 매개변수이다.

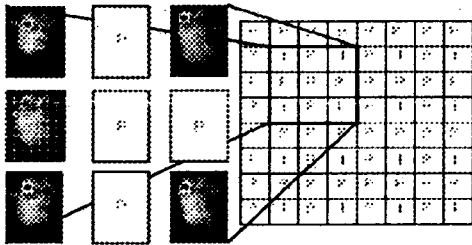


[그림 2] 광선의 사차원 매개변수화

3. 라이트 필드 렌더링

라이트 필드 렌더링은 일련의 이차원 영상을 사용하여 정적 삼차원 장면을 재현한다[3]. 각각의 영상은 라이트 필드라는 빛 분할 정보를 가지고 있으며, 복잡한 장면도 렌더링 속도에 영향을 미치지 않는다. 아무리 복잡도 높은 객체라 할지라도 이차원 영상에 담을 수 있기 때문에, 라이트 필드 렌더링에서는 장면 콘텐츠에 대한 어떠한 제약도 없다. 단지, 렌더링 화질은 사용 가능한 이차원 영상의 수에 따라 결정될 뿐이다. 하지만, 라이트 필드 렌더링은 실세계와 같은 렌더링 결과를 얻기 위해 수천장의 영상이 필요하다는 단점을 가지고 있다. 따라서 렌더링 하는 동안 로컬 기억장치에 해당 정보를 저장할 뿐만 아니라, 이 정보들을 전송하기 위해 영상 압축기술이 필수적이다. 라이트 필드 렌더링에서는 조명을 받은 물체 주위의 투명한 공간이 그 물체 표현에 반사되어 오는 빛으로 채워져 실세계와 같은 장면을 연출한다. 이러한 라이트 필드는 여러 시점으로부터 얻어진 이차원 영

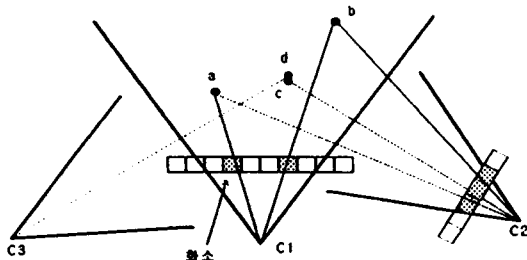
상들을 기록하여 얻어진다. 이러한 이차원 영상들을 규칙적인 격자로 이루어진 평면에 배치되게 되는데, 그림 3과 같이 이차원 영상들의 이차원 배열로서 라이트 필드를 표현한다.



[그림 3] 라이트 필드의 데이터 구조

4. 계층적 깊이 영상

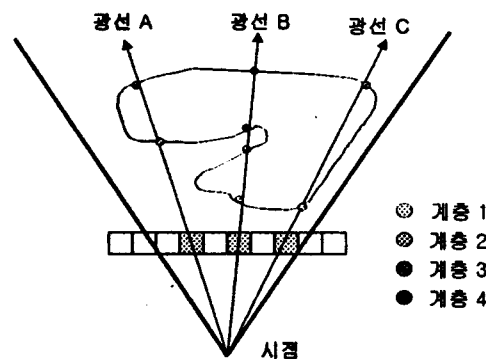
계층적 깊이 영상은 복잡한 기하학적 도형의 객체를 표현하고 렌더링 하기 위한 방법이다. 삼각형 메쉬를 사용해서 모델을 표현하는 방식과 달리, 계층적 깊이 영상은 단일 카메라 위치에서 보이는 화소의 배열을 사용해서 객체를 표현한다[1]. 각 계층적 깊이 영상 화소는 컬러 정보, 화소와 카메라사이의 거리를 나타내는 깊이 정보, 계층적 깊이 영상의 렌더링을 지원하는 몇 가지 다른 특성 정보로 표현된다. 계층적 깊이 영상은 각 시점 위치에 따른 다수의 계층으로 구성되어 있어서, 새로운 카메라 위치에서 객체나 장면에 대한 효과적인 렌더링이 가능하다. 렌더링의 복잡도는 계층적 깊이 영상에 포함되어 있는 화소의 개수와 연관이 있다. 즉, 화면의 복잡도가 아닌 시점 위치의 해상도와 관련된다. 따라서 사실적인 장면을 재현하기 위해 많은 다각형을 필요로 하는 삼차원 영상도 계층적 깊이 영상 기법을 사용하면 빠르게 렌더링할 수 있다. 계층적 깊이 영상 데이터를 획득하는 방법에는 여러 방식이 있다. 여기서는 단순히 다수의 깊이 정보를 가지고 있는 영상으로부터 계층적 깊이 영상을 생성하는 방식에 대해서 알아본다. N개의 깊이 정보를 가지고 있는 영상은 공통의 카메라 시점으로 워핑(Warping)을 통해서 계층적 깊이 영상 영상을 구성하게 된다.



[그림 4] 계층적 깊이 영상 획득

그림 4에서와 같이 공통 시점 C_1 에서 보았을 때의 계층적 깊이 영상 구조는 C_2 와 C_3 위치의 깊이 정보를 가지고 있는 영상을 워핑(Warping)하여 생성된다. 만약 입력 카메라 시점(C_2 와 C_3)에서 계층적 깊이 영상 카메라 시점으로 워핑을 하는 동안 두 개 이상의 화소값이 같은 계층으로 맵핑되면, 그 두개의 화소 위치에서 각 화소의 깊이 정보를 비교하게 된다. 만약 그 값이 초기에 설정한 기준 값보다 크게 되면 각각의 화소에 대해 그 화소 위치에 다른 계층이 더해진다. 그렇지 않을 경우, 두개의 화소값의 평균값이 하나의 계층을 만들고 그 계층의 깊이 값을 대표하게 된다.

계층적 깊이 영상은 시각 위치에 따라 가능한 다수의 계층과 더불어 단일 카메라 위치에서 보이는 화소의 배열로 이루어져 있다. 그림 5에 나타난 것처럼 광선들이 카메라 위치 P에서 투과된다. 광선은 객체와 다수의 점에서 교차하게 되는데, 그 점들은 뒤에서 앞으로 배치한다. 모든 광선 중에서 첫번째 교차점은 첫번째 계층을 이루게 된다. 두 번째 교차점은 두 번째 계층을 이루게 되고 나머지 계층도 이와 같은 방식으로 이루어지게 된다. 각 시각 위치에 따라 교차점의 개수는 계층 수(NOL)로 나타낸다. 예로써 광선 PA와 PC에서 광선에 교차점은 두 개가 있어서 두 계층을 이루게 된다. 비슷한 방식으로 광선 PB에 대해서는 광선이 객체와 4개의 교차점에서 만나게 되어 결과적으로 4개의 계층이 필요하게 된다. 원래의 카메라 위치 P에서는 첫번째 계층에 해당하는 화소만이 보이게 된다. 그러나, 뒤에 있는 계층의 화소들은 원래의 카메라 위치로부터 시선이 다른 곳으로 움직이게 되어 새로운 시각 위치에서는 뒤쪽에 있던 화소들이 보이게 된다.

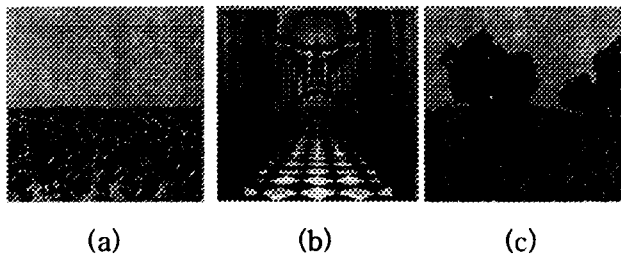


[그림 5] 계층적 깊이 영상 구조

위도 신호와 컬러 신호만으로 이루어진 일반적인 영상과 다르게 각 계층적 깊이 영상은 63비트의 정보를 포함하고 있다. 우선 R, G, B, 알파 채널 각각에 대해 8비트를 할당하고, 20비트를 깊이 값에 할당한다. 또한, 11비트는 스플랫 테이블(Splat Table)에 할당한다.

III. 기존의 계층적 깊이 영상 압축 방법

고해상도의 계층적 깊이 영상은 방대한 양의 데이터를 포함한다. 예를 들어, Cathedral 계층적 깊이 영상은 총 14MB의 데이터를 포함하고 있으며, 해상도는 1024 x 1024이고 1588812개의 깊이 값을 가지고 있다. 그림 6은 여러 계층적 깊이 영상을 보여준다. 사실적인 장면을 묘사하기 위해서는 더 높은 샘플링을 필요로 하게 되는데, 그 결과 데이터양은 더 많이 늘어난다. 표1은 영상의 해상도와 그 영상이 포함하고 있는 깊이 정보의 총 데이터 양을 보여준다[1].

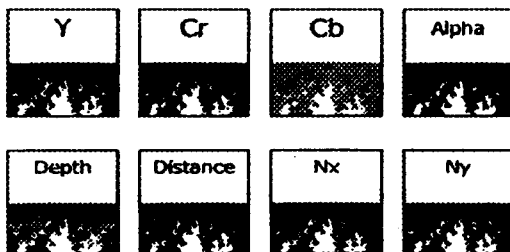


[그림 6] LDI: (a) Sunflowers, (b) Stream, (c) Cathedral

계층적 깊이 영상	해상도	계층의 개수	화소의 개수	용량
Sunflowers	1024 x 1024	6	463,821	5.55MB
Stream	1024 x 1024	12	1,125,690	10.86MB
Cathedral	1024 x 1024	24	1,588,812	14.10MB

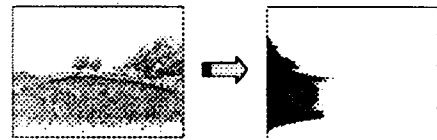
[표 1] 해상도에 따른 계층적 깊이 영상의 용량

계층적 깊이 영상은 세가지의 핵심적인 사항을 가지고 있다. 먼저 복합 계층구조를 가지고 있다. 두 번째는 계층상에서 후미의 계층은 낮은 화소 밀도를 가지고 있다는 점이다. 마지막으로, 각각의 화소는 컬러 정보, 깊이 정보, 스플랫 테이블 인덱스 등을 포함해서 다수의 특징 값을 가지고 있다. 계층적 깊이 영상 데이터는 R, G, B, Alpha, Depth, Distance, X norm, Y norm 이렇게 8개의 구성요소로 분리될 수 있으며, 이를 그림 7에 나타내었다. R, G, B 구성요소는 Y, Cr, Cb 구성요소로 변환된다. 압축된 계층적 깊이 영상 비트열은 NOL 비트열과 각 계층의 구성요소 이미지의 각각의 비트열과 합쳐져서 형성된다[1].



[그림 7] 계층적 깊이 영상 데이터의 구성요소 분리

계층적 깊이 영상 구성요소 영상은 낮은 화소 밀도를 가지고 사각형 영역 형태를 갖지 않는다. 계층이 더 후미에 있을수록 화소의 밀도는 더 낮아진다. 계층적 깊이 영상에서 구성요소 영상은 MPEG-4에서 연속적인 경계와 함께 고정적인 화소 밀도를 갖는 영상 또는 객체보다 더 낮은 화소 밀도를 갖는다. 따라서 각 계층의 데이터를 한 곳으로 모으기 위한 전처리 단계가 필요하다. 여기서 사용되는 데이터 모으기 (aggregation) 연산은 단순히 수평방향으로 데이터를 모은다. 같은 수평선상의 모든 화소들은 수평선상의 왼쪽으로 밀려지게 된다. 단순한 데이터를 모으는 연산을 그림 8에 나타내었다[1].



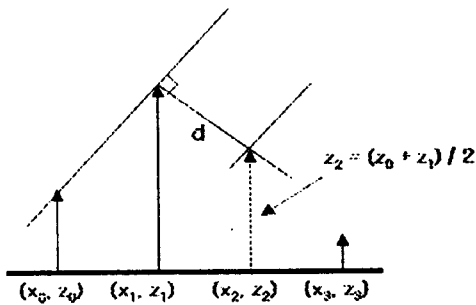
[그림 8] 데이터 모으기 (Data Aggregation)

데이터 한데 모으기 연산은 수평 객체 세그먼트를 길게 하고 변환 효율을 향상시킨다. 비록 수직방향의 배열에 영향을 미치는 단점이 있지만, 단점에 비해 더 많은 장점을 제공한다. 계층적 깊이 영상 구성요소를 압축하기 위해서 이미 존재하는 부호화 도구들을 사용할 수 있다. 여기서는 VOW (Video Object Wavelet) 코덱을 사용하였다. 구성요소 영상을 정확한 웨이블릿 계수로 변환하기 위해 VOW의 SAWT(Shape Adaptive Wavelet Transform)이 사용된다. 웨이블릿 계수는 양자화 되고 부분 비트평면 부호기와 함께 엔트로피 부호화된다.

IV. 제안된 깊이 정보 전처리 알고리즘

본 논문에서는 계층적 깊이 영상의 깊이 정보 압축에 직접적인 영향을 미치는 웨이블릿 변환의 효율을 높이기 위해 새로운 전처리 방법을 제안하였다. 계층적 깊이 영상은 화소에 대해 컬러 값, 깊이 값, 스플랫 인덱스 값을 포함하고 있다. 이 때 어떤 화소는 깊이 값을 가지지만, 어떤 화소는 깊이 값을 가지고 있지 않다. 특히 계층의 수가 늘어나 계층의 후미로 갈수록 이러한 현상이 두드러진다. 즉 계층이 늘어날수록 화소 및 깊이 값의 분포가 최소화되는 특성이 나타난다. 기존 알고리즘은 이러한 특성을 이용하기 위해 웨이블릿 변환을 수행하기 전 데이터 모으기 기법을 사용하였다. 본 논문에서는 깊이 정보의 압축 효율을 높이기 위해, 추가적으로 전처리 과정을 수행한다. 제안된 알고리즘은 다음과 같다. 화소에 대한 x, y 좌표 값과 깊이를 나타내는 z 값을 살펴보면, y 축을 고정하였을 때 (x, z) 값이 변하는 것을 확인 할 수 있다. 따라서 일차원인 깊이 정보를 이차원

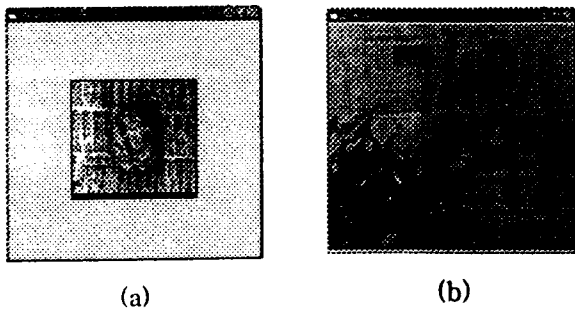
공간상의 한 점으로 해석할 수 있게 된다. 본 논문에서는 원래의 깊이 값을 그대로 사용해 데이터 모으기를 하는 대신, 깊이 값의 분산을 줄이기 위해 직선과 한 점의 거리를 계산하였다. 즉, 이전의 두 깊이 값을 연결하는 직선과 현재 깊이 값 사이의 최단 거리를 계산하여 현재 화소의 깊이 값을 계산된 최단 거리로 대체한다. 그 후에 데이터 모으기와 웨이블릿 변환을 수행한다. 이 때, 계층적 깊이 영상의 모든 화소에 대해 계층별로 깊이 값이 존재하지 않는 경우도 존재하므로 이를 고려해 주어야 한다. 즉, 이전 화소의 깊이 값이 없을 경우에는 이전의 두 점을 연결하는 직선을 구할 수가 없다. 이를 해결하기 위해 깊이 값이 없는 화소에 대해서는 이전 두 점의 평균값으로 깊이 값을 삽입해 준다. 그리고 이 평균값을 현재 화소의 깊이 값으로 사용하여 최단 거리를 계산한다. 이를 그림 9에 나타내었다. 최종적으로 데이터 모으기 연산을 수행하기 전에는 임시로 삽입된 평균값을 제거하여 준다.



[그림 9] 비어있는 깊이 값 및 최단 거리 계산

V. 실험결과

본 논문에서는 계층적 깊이 영상의 깊이 정보를 압축하기 위해, 화소의 깊이 값을 제안된 알고리즘을 통해 계산한 최단 거리로 대체하였다. 실험 데이터는 그림 10에 나타낸 두 장의 계층적 깊이 영상을 사용하였다.



[그림 10] 계층적 깊이 영상 실험 데이터: (a) Ball, (b) Flower

표2는 전처리 후 분산과 원본 계층적 깊이 영상의 깊이 값 표준편차를 비교한 결과이다. 표1을 보면 전처리 후 깊이 값의 분산이 원본에 비해 매우 작음을 알 수 있다.

	Ball		Flower	
	원본	전처리 후 영상	원본	전처리 후 영상
계층 1	51.98	27.06	1815.07	364.46
계층 2	118.14	49.81	3080.90	609.16
계층 3	160.31	74.16	3749.86	769.95

[표 2] 깊이 값의 표준편차 비교

표3은 웨이블릿 변환을 수행한 후 가변길이 부호화를 거친 데이터량을 보여준다. 제안한 알고리즘을 적용한 후 데이터량이 감소함을 확인할 수 있다.

	Ball		Flower	
	원본	전처리 후 영상	원본	전처리 후 영상
깊이정보 데이터량	78.52	67.54	415.66	363.78

[표 3] 깊이 정보의 데이터량 비교 [단위: KB]

VI. 결 론

본 논문에서는 계층적 깊이 영상의 깊이 정보를 압축하기 위한 전처리 알고리즘을 제안하였다. 계층적 깊이 영상에서는 계층이 늘어날수록 깊이 값의 밀도가 낮아지는 특성이 있어 데이터 모으기를 통해 변환 효율을 향상시킬 수 있다. 여기에 추가적으로 일차원 깊이 정보를 이차원 공간상의 점으로 해석하여 최단 거리를 계산하였다. 현재 깊이 값을 이 값으로 대체시킴으로써 계층별 깊이 값의 분산을 줄일 수 있었고, 결과적으로 변환 효율을 증가시켜 깊이 값을 효과적으로 압축할 수 있었다.

감사의 글 (Acknowledgements)

본 연구는 광주과학기술원(K-JIST)과 광주과학기술원 실감방송 연구센터를 통한 대학IT연구센터(ITRC), 그리고 교육부 두뇌한국21(BK21) 정보기술사업단의 지원에 의한 것입니다.

참 고 문 헌

- [1] J. Duan and J. Li, "Compression of the layered depth image", IEEE Trans. on Image Processing, vol.12, pp.365-372, March 2003.
- [2] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph", Proc. ACM SIGGRAPH96, pp. 43-54, Aug. 1996.
- [3] M. Levoy and P. Hanrahan, "Light field rendering", Proc. ACM SIGGRAPH 96, pp. 31-42, Aug. 1996.