

적응적 비트 할당과 DFSVQ에 기반한 3차원 합성영상의 부호화

정희원 박진우*, 송근원*, 박영식*, 남재열*, 하영호*

3D Synthetic Image Compression Based on Adaptive Bit Allocation and DFSVQ

Jin-Woo Park*, Kun-Woen Song*, Young-Sik Park*, Jae-Yeal Nam*,
Yeong-Ho Ha* *Regular Members*

요약

본 논문에서는 3차원 물체의 연결성 정보 및 기하학 정보의 부호화를 위한 새로운 기법을 제안하였다. 연결성 정보의 부호화에서는 3차원 물체를 분해하여 정점 신장나무와 삼각형 신장나무를 구한 뒤, 각 신장나무의 런 길이가 문턱값보다 크면 이진수로 표현하고 그렇지 않으면 비트열로 표현하여 적응적 비트 할당을 통해 과잉 비트를 감소시켰다. 이때 문턱값은 최대길이 런의 이진숫자의 길이이다.

기하학 정보의 부호화에서는 많은 코드워드를 가지며 적은 비트로 부호화를 할 수 있고 국부적인 상관성에 적합한 특성을 가진 유한상태 동적 벡터 양자화를 이용하여 기하학 정보를 부호화하였다. 제안된 기하학 정보의 부호화 방법은 기존의 방법보다 적은 오차와 개선된 화질을 보여준다.

ABSTRACT

In this paper, new methods are proposed for connectivity and geometry compression of 3D object. In connectivity compression, after obtaining the vertex and triangle spanning trees by decomposing 3D object, bits are adaptively allocated to each run of two spanning trees. The threshold which is the length of the binary number of the maximum run length. If a certain run length exceeds threshold, it is represented by binary number, Otherwise it is represented by bitstream. Therefore, compression efficiency is enhanced through an adaptive bit allocation to each run of two spanning trees.

In geometry compression, DFSVQ (Dynamic Finite State Vector Quantization), which has many codewords and local correlation, is proposed to encode the geometry information. It produces less distortion and gives better visual quality than conventional methods.

I. 서론

인터넷의 사용이 활성화됨에 따라 여러가지 멀티미디어 서비스가 점차 증가하고 있다. 또한 인터넷에서의 3차원 가상공간에 대한 관심이 높아감에 따라 3차원 합성영상 서비스의 요구가 점차 증가하고 있다. 그러나 3차원 합성영상은 데이터의 양이 방대하여 저장 또는 전송시에 많은 문제점이 발생되므로

부호화가 필수적으로 요구되어진다. 3차원 물체는 일반적으로 다각형 메쉬(polygonal mesh)로 표현되어진다. 다각형 메쉬로 이루어진 3차원 물체의 데이터는 기하학 정보 및 연결성 정보로 구성되며, 기하학 정보는 3차원 좌표상에서 합성영상에 대한 정점의 위치를 나타내고 연결성 정보는 각 점들간의 연결상태를 표현한다. 이들에 대한 부호화는 각 정보에 대해 독립적으로 행하여진다.

*경북대학교 전자전기공학부, 논문번호:98010-0106, 접수일자:1998년 1월 6일
*본 논문은 97년도 정보통신부 대학기초연구사업의 연구비에 의해 수행된 결과임

연결성 정보의 부호화를 위해서 Michael Deering[1]은 일반화된 삼각형 메쉬(generalized triangle mesh)를 제안하여 3차원 물체를 구성하는 삼각형에서 꼭지점의 중복을 피하여 데이터를 감소시켰다. Gabriel Taubin등[2]은 3차원 물체를 분해하여 삼각형 정점을 한번만 통과하는 정점 신장나무(vertex spanning tree)와 잘려진 삼각형 메쉬의 쌍대그래프인 삼각형 신장나무(triangle spanning tree)로 표현했다. 정점 신장나무는 런(run)의 길이와 런의 상태에 따른 2개의 상태비트(state bit)인 잎비트(leaf bit)와 끝비트(last bit)로 부호화되고, 삼각형 신장나무는 런의 길이와 상태비트인 잎비트와 진행비트(marching bit)로 부호화되었다. 이 방법에서는 각 나무가지의 길이를 고려하지 않고 각 나무가지에 같은 양의 비트를 할당하여 부호화하기 때문에 가지길이의 변화가 크면 많은 잉여정보를 발생시킨다. Frank Bossen[3]은 3차원 물체를 분해하여 두 개의 신장나무로 표현한 다음, 신장나무를 부호화함에 있어서 정점 신장나무의 경우에는 깊이 우선 탐색을 통하여 근 정점에서 멀어지면 각 가지에 '1' 비트를 할당하고 가까와 지면 '0' 비트를 할당하여 각 정점당 2비트를 할당하였다. 삼각형 신장나무의 경우에는 분해된 삼각형 메쉬안에서 삼각형들 간의 연결상태에 따라서 '0' 또는 '1' 비트를 할당하였다. 이와 같은 방법은 가지의 길이가 긴 경우 많은 잉여비트를 발생시킨다.

본 논문에서 제안한 연결성 정보의 부호화 기법은 삼각형 신장나무와 정점 신장나무의 런의 길이에 따라 부호화하는 방법을 달리하여 적응적으로 비트를 할당함으로써 잉여비트의 발생을 줄인다. 즉 런의 길이가 문턱 값 이상이면 최대길이의 런을 표현하기 위해 필요한 길이의 이진수로 표현하고, 그렇지 않은 경우에는 런의 길이만큼 '0'의 열과 마지막의 '1'인 비트를 추가하여 표현한다. 이때 문턱값은 최대길이의 런을 표현하기 위한 이진수의 길이로 한다. 그리고 삼각형 신장나무도 정점 신장나무에서와 같은 방법으로 부호화를 한다.

본 논문에서는 두개의 신장나무 런의 길이를 표현하기 위하여 Taubin의 방법에서 하나의 상태비트를 추가하였지만 문턱값에 따라 적응적으로 비트를 할

당하여 기존의 방법보다 삼각형 신장나무에서는 2,500-10,000비트를 감소시켰고 정점 신장나무에 있어서는 200-12,000비트 감소를 나타내었다.

기하학 정보의 부호화에 있어서 Michael Deering은 부동소수점의 96비트로 표현되는 3차원 정점들을 양자화하여 데이터를 줄였다. Gabriel Taubin은 정점신장나무를 깊이 우선 탐색(depth-first-searching)으로 각각의 대응되는 각 점의 좌표값을 저장하고 정점들을 DPCM을 이용하여 부호화하였다. 그리고 최[5]는 깊이 우선 탐색에 의하여 저장된 정점들을 예측 잔여 벡터 양자화(predictive residual vector quantization)로 부호화하였다.

본 논문에서 제안한 기하학 정보 부호화기법은 정점 신장나무를 깊이 우선 탐색으로 탐색한 순서로 정렬된 점들에 대하여 유한상태 동적 벡터 양자화(dynamic finite state vector quantization)[6-7]를 적용하여 기하학 정보를 부호화하였다. 유한상태 동적 벡터 양자화는 기존의 2차원 영상 부호화에서 영상의 국부적인 상관성을 이용하여 여러개의 한정된 상태로 나누어 입력의 국부적인 특성을 예측된 특정상태에서 부호화하고 이러한 상태를 동적으로 생성시키는 방법이다. 3차원 기하학 정보는 위치간의 국부적 상관성이 크고 3차원 벡터로 표현되기 때문에 적은 비트로 많은 코드워드를 발생시키는 유한상태 동적 벡터 양자화를 이용하여 부호화 함으로써 왜곡을 줄이고 부호화율을 높일수 있다.

II. 3차원 물체의 부호화

인터넷에서 3차원 물체를 표현하기 위해 VRML(Virtual Reality Modeling Language)을 사용하고 있다. VRML은 SILICON GRAPHICS의 openInventor를 단순화하여 인터넷상에서 3차원 그래픽 공간을 구성하도록 제작된 그래픽언어이다. 본 논문에서는 3차원 물체의 기본적인 표현방식을 VRML의 데이터 형태에 맞추었다. VRML은 3차원 공간을 구성하는 노드(node)라는 기본단위로 기술하고 있다. 3차원 물체를 구성하기 위하여 사용하기 위한 여러가지 종류의 노드 중에서 Coordinate3노드는 하나의 점에 대한

3차원 좌표값과 순서가 매겨지고 IndexedFaceSet노드에서는 삼각형을 표현하기 위한 3개의 점에 대한 좌표값과 끝에는 '-1'이 추가되어 표현되고 삼각형에 대하여 순서가 매겨진다.

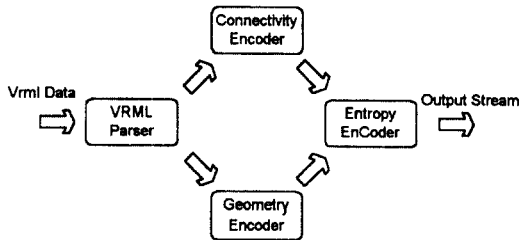


그림 1. VRML 부호화기
Fig. 1. VRML encoder.

그림 1에서와 같이 VRML 데이터가 들어오면 VRML 분석기를 통해 기하학 정보와 연결성 정보로 분류되어진다. 분류된 연결성 정보는 연결성 정보 부호화기로 보내어지고 기하학 정보는 기하학 정보 부호화기로 보내어진다. 각각의 부호화기에서 부호화된 데이터는 엔트로피 부호화기로 보내어 부호화되고 전송된다.

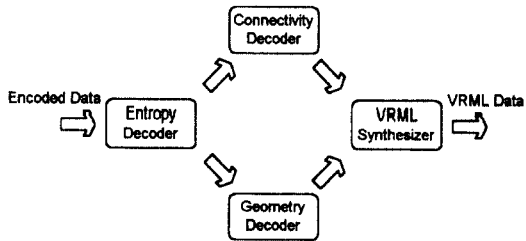


그림 2. VRML 복호화기
Fig. 2. VRML decoder.

그림 2는 복호화기의 구조를 보여준다. 부호화된 데이터를 입력으로 받아서 엔트로피 복호화기에서 복호화를 한 다음, 연결성 정보 비트열은 연결성 정보 복호화기로 보내지고 기하학 정보 비트열은 기하학 정보 복호화기로 보내져 복호화한다. 각각의 복호화기로부터 복호화된 데이터들은 VRML 합성기로 보내어져서 VRML화일을 구성한다.

1. 연결성 정보의 부호화

3차원 물체를 표현하는 삼각형 메쉬에서 하나의 삼각형은 3개의 정점으로 구성되어 있다. 여기서 하나의 변을 공유하는 삼각형열에서 한개의 정점을 추가함으로써 삼각형을 표현할 수 있다. Michael Deering은 삼각형 메쉬를 표현하기 위하여 하나의 삼각형에 연속되는 정점을 추가하여 또 다른 삼각형을 표현함으로써 삼각형 메쉬의 데이터양을 감소시켰다. 이는 이웃하는 삼각형이 하나의 변을 공유하는 것을 이용하여 2개의 정점을 생략하여 표현되는 점을 감소시켰다. 이러한 표현 방법을 일반화된 삼각형 줄 (generalized triangle strip)이라고 한다. 일반화된 삼각형 줄에서 또 다른 중복되는 점들이 나타나므로 뒤에 사용되는 점들의 경우에는 스택에 저장하여 다시 사용될 때 스택에서 복원하여 삼각형을 구성하였다. 이를 일반화된 삼각형 메쉬라고 한다.

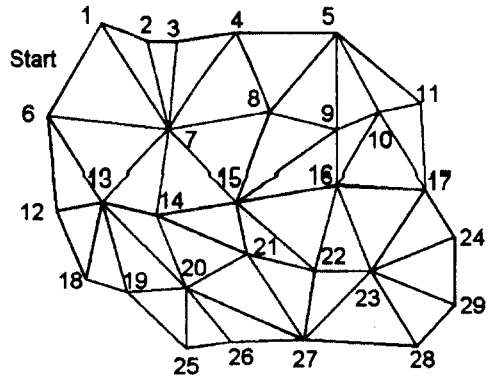


그림 3. 일반화된 삼각형 메쉬
Fig. 3. Generalized triangle mesh.

그림 3은 임의의 삼각형 메쉬를 나타낸 것이다. 이 삼각형 메쉬를 일반화된 삼각형 줄과 일반화된 삼각형 메쉬로 표현하면 다음과 같이 된다.

<VRML form>

1,6,7; 1,2,7; 2,3,7; 3,4,7; 4,5,8; 4,8,7; 5,8,9; 5,9,11;
5,10,11; 6,13,7; 13,7,14; 7,14,15; 7,15,8; 8,15,9; 9,15,16;
9,10,16; 10,16,17; 11,10,17; 12,13,18; 13,18,19;
13,19,20; 13,14,20; 14,20,21; 14,15,21; 15,21,22;

15,22,16; 16,22,23; 16,17,23; 17,23,24; 23,24,29;
23,29,28; 23,27,28; 22,23,27; 21,22,27; 20,21,27;
20,26,27; 20,25,26; 20,25,26; 19,20,25;

<Generalized triangle strip>

R6,O1,O7,O2,O3,M4,M8,O5,O9,O10,M11,O17,M16,
M9,O15,O8,O7,M14,O13,M6,O12,M18,M19,M20,M14,
O21,O15,O22,O16,O23,O17,O25,M29,M28,M27,M22,O
21,M20,M26,O25,M19

<Generalized triangle mesh>

R6P,O1,O7P,O2,O3,M4,M8P,O5,O9P,O10,M11,O17P
,M16P,M-3,O15P,O-5,O6,M14P,O13P,M-
9,O12,M18P,M19P,M20P,M-5,O21P,O-7,O22P,O-
9,O23,O-10,O-7,M29,M28,M27,M-1,O-2,M-
3,M26,O25,M-4

첫문자인 R은 재출발(restart)을 나타내고, O는 가장 오래 것을 바꿈(replace oldest), M은 가운데 것을 바꿈(replace mi-trailing), P는 푸쉬(push), 그리고 -숫자는 팝(pop)하는 것을 나타낸다. 즉 일반화된 삼각형 줄에서는 6번 정점에서 시작하여 1번과 7번으로 첫번째 삼각형을 구성하고 가장 오래된 정점 즉, 6번 정점을 2번 정점과 바꾸어서 추가 삼각형을 구성한다. 그리고 M4를 만나면 7, 2, 3정점 중 가운데 정점 2번을 4번과 바꿔서 추가 삼각형을 구성한다. 그리고 일반화된 삼각형 메쉬는 일반화된 삼각형 줄과 비교해서 P가 붙은 정점은 스택에 넣고 -숫자는 스택에서 그 숫자에 해당하는 위치의 값을 팝한다. 그림 3에서 39개의 삼각형에 대하여 117개의 정점이 사용되나 Michael Deering의 방법을 이용하면 29개의 정점으로 표현된다.

Gabriel taubin과 Jarek rossignac은 연결성 정보의 부호화에 관해 계층분해(layered decomposition)방법을 제안하였다. 3차원물체를 분해하여 정점 신장나무와 삼각형 신장나무를 구성한뒤, 신장나무들을 부호화하여 연결성 정보를 부호화한다. 이때 3차원 물체를 분해하는 기법에 따라 부호화율이 달라진다. 부호화 효율을 높이기 위해서는 신장나무 런의 길이

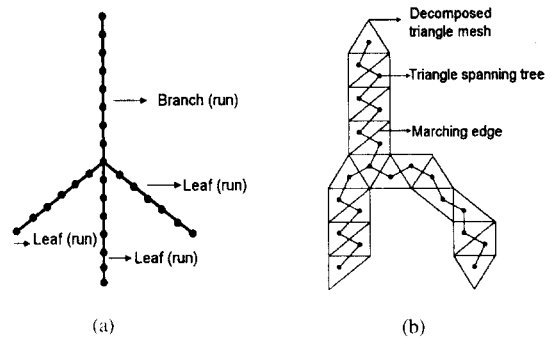


그림 4. 정점 신장나무와 삼각형 신장나무의 예
(a) 정점 신장나무 (b) 삼각형 신장나무

Fig. 4. An example of a vertex and a triangle spanning tree.
(a) vertex spanning tree (b) triangle spanning tree

가 길고 개수가 적어야 한다. 이를 위해 하나의 삼각형 열로 길게 자르는 계층분해 방법을 제안하였다. 그림 4는 기본적인 신장나무를 나타낸다. 런은 나무에서 노드간의 연결이 분기되기까지의 연결을 의미한다. (a)에는 4개의 런이 있다. 런 중에서 분기가 시작되기까지의 런을 가지라하고 연결이 끝나는 런을 잎이라한다. (b)는 삼각형 신장나무를 나타낸다.

위상학적 구조가 단순한 경우에는 절단에지가 정점 신장나무의 에지가 된다. 정점나무는 모든 정점을 포함하고 있고, 분해된 삼각형 메쉬의 쌍대그래프가 삼각형 신장나무이다. Gabriel taubin의 경우 정점 신장나무를 깊이 우선 탐색에 의해 런의 길이와 런의 상태를 나타내는 2비트로 표현하였다. 즉 정점 신장나무는 런이 앞인지를 표현하는 앞비트, 가지에 연결된 많은 런중에 깊이 우선 탐색으로 마지막으로 방문하는 앞인가를 표시하는 마지막비트, 그리고 런의 길이로 부호화하였다. 정점 신장나무의 가장 긴 줄기의 길이를 표현하는 데 필요한 비트 수를 기준으로 모든 런의 길이를 표현하는 데 일정한 비트를 할당하였다. 3차원 물체를 분해하여 생긴 삼각형 메쉬의 쌍대그래프인 삼각형 신장나무의 경우 삼각형 메쉬에서 삼각형끼리 접하는 변을 진행에지라 한다. 이러한 삼각형 메쉬의 쌍대그래프는 삼각형을 노드로 하고 진행에지를 가로질러 연결된다. 삼각형 신장나무의 경우에 삼각형이 삼면을 가지므로 이진트리 형태를 가진

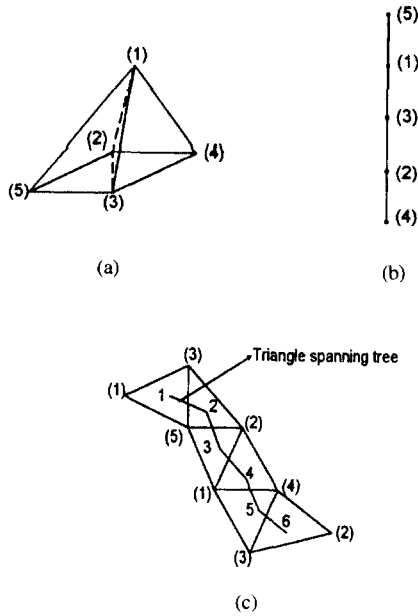


그림 5. 3차원 물체 분해를 통한 연결성정보의 부호화
 (a) 사각뿔 (b) 정점 신장나무 (c) 삼각형 신장나무
 Fig.5. Connectivity compression through 3D object decomposition.
 (a) quadrangle (b) vertex spanning tree (c) triangle spanning tree

다. 삼각형 신장나무의 부호화는 각 줄기의 길이와 각 줄기가 가지인지 앞인지를 구분하는 앞비트와 삼각형 메쉬에서 이전의 삼각형과 다음에 오는 삼각형이 오른쪽 또는 왼쪽의 면 중 어느 면과 연결되어 있는지를 표현하는 진행비트로 부호화된다. 삼각형 신장나무의 가지는 두개의 런과 연결되는 이진나무이기 때문에 정점 신장나무와 달리 마지막 비트가 필요치 않다. 줄기의 길이는 정점 신장나무의 경우와 같은 방법으로 부호화한다.

<VRML form>

연결성정보 [1, 2, 5, -1,
 2, 5, 3, -1,
 2, 4, 3, -1,
 1, 3, 5, -1,
 1, 3, 4, -1
 1, 2, 4, -1]

위의 VRML 정보는 그림 5의 사각뿔을 기술한 것이다. (a)의 사각뿔을 분해하여 정점 신장나무를 구성하면 (b)와 같이 된다. 그리고 분해된 삼각형 메쉬는 (c)에 나타나 있다. 가운데 연결된 선이 삼각형 메쉬의 쌍대그래프인 삼각형 나무이다. (b)의 정점 신장나무는 1개의 런을 가진다. 런의 길이는 4이고 앞비트는 1 가지에 연결된 마지막 런이 아니므로 마지막비트는 0이다. 그리고 삼각형 신장나무는 길이가 4이고 앞비트는 1, 진행비트는 근 삼각형을 제외하고 삼각형열의 각각의 삼각형의 진행에지의 방향이 왼쪽, 오른쪽, 왼쪽, 오른쪽이므로 1010이 된다.

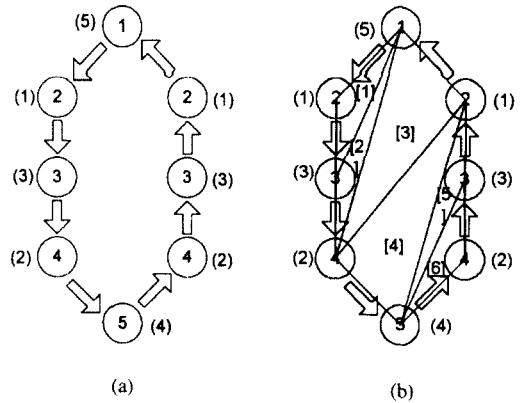


그림 6. 경계선 고리와 삼각형의 복원 (a) 경계선 고리 (b) 복원된 삼각형 메쉬
 Fig. 6. Bound loop and reconstructed triangle. (a) bound loop (b) reconstructed triangle mesh

연결성 정보를 복호화하는 방법은 다음과 같다. 먼저 부호화된 정점 신장나무로 그림 6(a)와 같이 경계선 고리(bound loop)를 구성하고 이를 삼각형 메쉬의 경계로 하여 삼각형나무의 진행비트를 이용하여 삼각형 메쉬를 복원하였다. 즉 경계선 고리를 구성한 다음 그것에서 근 정점을 선택하고 근 삼각형의 양쪽의 정점을 선택하여 삼각형을 구성하고 진행비트에 따라서 '0' 인 경우에는 경계선 고리에서 오른쪽의 정점을, '1' 인 경우에는 왼쪽의 정점을 선택하여 공통변을 가지는 새로운 삼각형을 구성한다. 그리고 계속적인 반복을 통하여 삼각형 메쉬를 복원하였다. 그

림 6에서 원안의 수는 경계선 고리에서의 인덱스이고 원호안의 수는 실제 정점의 인덱스이다. 그리고 대괄호안의 수는 복원된 삼각형의 인덱스이다.

Frank Bossen은 3차원 물체를 구성하는 삼각형에 대하여 하나의 삼각형에 3비트를 할당하였다. 계층 분해방법과 유사하게 3차원 물체를 분해하여서 삼각형 신장나무와 정점 신장나무를 구하였다. 삼각형 신장나무에서 삼각형 노드에 대하여 가지의 연결이 왼쪽인지 오른쪽인지를 나타내는 두 비트를 할당하였다. 그리고 정점 신장나무의 경우에는 정점 신장나무를 깊이 우선 탐색으로 나무의 런을 따라 방문하면서 하나의 비트를 할당하여 근 정점에서 멀어지는 것과 가까워지는 것을 표현하였다. 여기서 멀어지는 것을 '1'로 표현하고 가까워지는 것을 '0'으로 표현하였다. 표 1과 2는 삼각형 신장나무의 부호화와 정점 신장나무의 부호화에 대해 나타내고 있다. 그림 5에서의 정점 신장나무를 부호화하면 (110100)이 된다. 그리고 그림 6의 삼각형 신장나무는 (100110100100)으로 표현된다.

표 1. 삼각형 신장나무의 부호화
Table 1. The code of triangle spanning tree

| Triangle | Connection | Code |
|----------|---------------|------|
| 1 | right | 10 |
| 2 | right | 01 |
| 3 | left | 10 |
| 4 | left | 10 |
| 5 | right | 01 |
| 6 | no connection | 00 |

표 2. 정점 신장나무의 부호화
Table 2. The code of vertex spanning tree

| edge | Direction | 부호 |
|-------|-----------|----|
| (1-2) | away | 0 |
| (2-3) | away | 0 |
| (3-4) | away | 0 |
| (4-3) | forward | 1 |
| (4-5) | away | 0 |
| (5-4) | forward | 1 |
| (3-2) | forward | 1 |
| (2-1) | forward | 1 |

복호화는 삼각형 신장나무를 표현한 비트를 이용하여 근 삼각형에 다음 삼각형이 어디에 붙을지를 결정하여 다음 삼각형을 붙여서 삼각형 메쉬를 복원한다. 그리고 그 삼각형 메쉬의 경계부분을 운행하면서 정점 신장나무를 표현한 비트를 이용하여 '1'인 경우에는 근 정점에서 멀어지면서 인덱스를 증가시키고 '0'인 경우에는 인덱스를 감소시키면서 삼각형 메쉬 각각의 꼭지점에 인덱스를 부여하여 전체적인 삼각형 메쉬를 복원한다.

2. 기하학 정보의 부호화

Taubin은 정점 신장나무의 운행순서로 저장된 정점의 위치정보를 아래식으로 DPCM을 적용하였다.

$$P(\lambda, v_{n-1}, \dots, v_{n-k}) = \sum_{i=1}^k \lambda_i v_{n-i}, \quad (1)$$

$$\lambda = (\lambda_1, \dots, \lambda_k)$$

$$\sum_{n \geq K} \|\epsilon_n\|^2 \quad (2)$$

여기서 K 는 예측에 사용되는 이전 정점의 개수이다. 식(2)는 전체오차를 나타낸다. 예측계수를 구하기 위하여 식(2)에서의 예측값과의 오차가 최소가 되는 λ 를 구한다.

3차원 좌표의 정점을 3차원 벡터로 보고 벡터양자화를 하는 기법이 제안되었다. 계층 분해를 이용하여 분해된 3차원 물체의 정점들 즉, 정점 신장나무를 깊이 우선 탐색순서로 운행하면서 대응되는 정점을 순서대로 저장된 점들이 인접한 점들사이에 공간적인 근접성을 가지는 것을 이용하여 정점들간의 차를 구하여 상관성을 제거하여 3차원 벡터로 부호화를 하였다.

$$\hat{X}_n = P(X'_{n-1}, X'_{n-2}, X'_{n-3}, X'_{n-4}) \quad (3)$$

식(3)의 예측기에서 과거의 복원된 벡터로부터 입력의 예측벡터 X_n 을 계산한다.

식(4)에서 입력벡터 X_n 과 예측벡터 X_n 의 차를 구하고 X_n 과 X_n 의 오차를 양자화하여 인덱스를 전송한다.

$$e_{n1} = X_n - \hat{X}_n \quad (4)$$

$$e'_{n1} = Q(e_{n1}) \quad (5)$$

$$e_{n2} = e_{n1} - e'_{n1} \quad (6)$$

$$e'_{n2} = Q(e_{n2}) \quad (7)$$

$$e_{n3} = e_{n2} - e'_{n2} \quad (8)$$

$$e'_{n3} = Q(e_{n3}) \quad (9)$$

그리고 오차와 양자화된 오차의 차 즉, 잔여값을 다시 양자화하고 양자화된 값과 원래의 값의 차이를 구하여 다시 양자화하여 전송한다. 전체적으로 하나의 인덱스와 2개의 잔여벡터 인덱스를 전송한다.

$$X'_n = \hat{X}_n + e'_{n1} + e_{n2} + e'_{n3} \quad (10)$$

식(10)에서 예측벡터 X_n 과 3개의 양자화된 벡터 e'_{n1}, e_{n2}, e'_{n3} 을 더하여 X'_n 을 복원한다. 입력이 들어오면 예측기에서 예측한 입력 X_n 과 입력과의 차를 벡터 부호화기로 보내어 벡터 부호화를 한다. 그리고 양자화된 값과 부호화되기 전의 값과의 차를 제 1 잔여벡터 부호화기로 보내어 부호화를 하고 다시 양자화된 것과 양자화되기 전의 것과의 차를 제 2 잔여벡터 양자화기로 보내어 양자화한다. 각각의 양자화된 값들을 더하여 양자화된 입력 X'_n 을 복원하여 예측화기로 보내어 다음 입력을 예측하게 했다. 복호화과정은 하나의 벡터 인덱스와 2개의 잔여벡터 인덱스를 받아서 각각의 벡터 복호화기에서 복호화한다. 이들의 합을 구하여 벡터값을 복호화한다.

III. 제안한 3차원 물체의 부호화방법

본 논문에서는 3차원 물체에 대한 연결성 정보와 기하학 정보의 부호화방법을 위한 새로운 알고리즘을 제안하였다. 연결성 정보 부호화에서는 3차원 물체를 2개의 신장나무로 분해하고 각각의 신장나무의 줄기에 대하여 기존의 일정한 비트를 할당하여 발생하는 과잉비트를 적응적 비트 할당을 통하여 감소시켰다. 그리고 기하학 정보의 부호화에서는 3차원 정

점을 3차원의 벡터로 보고 유한상태 동적 벡터 양자화를 적용하여 기존의 방법에 비해 비트율과 화질을 개선하였다.

1. 제안한 연결성 정보의 부호화

본 논문에서 제안한 연결성 정보의 부호화는 3차원 물체를 분해하여 정점 신장나무와 삼각형 신장나무로 표현한다. 분해하는 방법은 계층 분해방법을 사용하였다. 정점 나무와 삼각형 나무의 길이를 부호화함에 있어서 Taubin의 방법이나 Bossen방법의 과잉비트를 제거하여 압축효율을 높였다. Bossen의 방법은 정점 신장나무를 표현할 경우 근 정점에서 멀어지면 '0'을 그리고 가까와지면 '1'을 할당한다. 이 방법은 신장나무의 줄기를 따라서 깊이 우선 탐색을 이용하여 각 노드에 약 2비트를 할당하게 된다. 이것은 한 줄기를 근정점에서 멀어지면서 운행해서 다시 돌아서 근정점으로 돌아오므로 하나의 노드에 대하여 과잉정보를 가진다. Taubin의 방법은 나무를 부호화할 때, 각 줄기의 길이만을 표현하므로 반복적인 중복성은 없지만, 줄기의 길이를 표현할때 할당하는 비트가 일정하여 잉여비트를 발생시킨다. 다시말해 이진수로 최대길이의 줄기를 표현할때 요구되는 길이의 비트로 모든 나무줄기의 길이를 나타내었다. 하지만 실제 계층 분해방법으로 분해된 신장나무들은 줄기의 길이가 하나인 많은 줄기를 포함하고 있으므로, 이러한 방법은 길이가 '1'과 같은 짧은 줄기를 표현할 경우 많은 과잉비트를 발생시킨다.

본 논문에서는 짧은 줄기에 할당되는 과잉비트를 줄이기위해 Taubin의 삼각형 신장나무와 정점 신장나무의 표현에 하나의 상태비트를 추가하여 런을 길이에 따라 적응적으로 비트를 할당하여 잉여비트를

표 3. 정점 신장나무의 상태부분

Table 3. The state part of vertex spanning tree

| State part | Run bit | Leaf bit | Last bit |
|------------|---------------|----------|----------------|
| 000 | Binary number | No leaf | No last branch |
| 001 | Binary number | No leaf | Last branch |
| 010 | Binary number | Leaf | No last branch |
| 011 | Binary number | Leaf | Last branch |
| 100 | Bit stream | No leaf | No last branch |
| 101 | Bit stream | No leaf | Last branch |
| 110 | Bit stream | Leaf | No last branch |
| 111 | Bit stream | Leaf | Last branch |

줄였다. 추가된 상태비트가 '0' 인 경우에는 이어오는 일정 비트열이 이진숫자로서 런의 길이를 표현하고 상태비트가 '1' 인 경우에는 따라오는 '0' 비트열과 마지막 '1'의 비트의 전체개수가 런의 길이를 나타낸다. 표 3에서 정점 신장나무의 상태에 대하여 정의하고 있다.

표 3을 이용하여 정점 신장나무를 효율적으로 부호화 할 수 있다. 예를 들어 길이가 각 9, 400, 40, 5, 1 이고 런의 상태가 가지, 잎, 잎, 잎, 마지막 잎인 런들이 있다. 먼저 런들을 제안된 방법으로 부호화하려면 가장 긴 런을 표현하는 데 할당되는 비트를 알아야 한다. 400이면 9비트의 이진수로 표현이 가능하게 된다. 문턱값을 9로 하여 런의 길이가 9보다 크면 런비트가 '0', 아니면 '1'이 된다. 런비트가 '0'이면 9비트의 수로 길이를 표현하고 아니면 '0' 비트의 열과 마지막에는 '1'로 끝이나는 비트열로서 길이의 개수만큼의 비트로 부호화 해준다. 부호화의 결과는 다음과 같다. "000"(000001001) "010"(110010000) "010"(000101000) "110"(00001) "111"(1). 따옴표를 한부분은 상태부분이고 괄호를 한 부분은 길이부분이다.

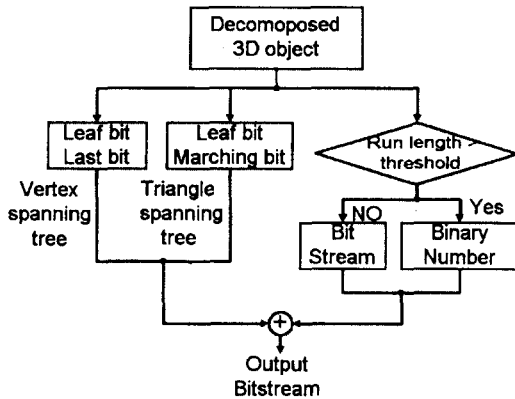


그림 7. 제안된 연결성 정보부호화기
Fig. 7. Proposed encoder for connectivity information

그림 7은 삼각형 신장나무와 정점 신장나무가 입력인 경우에 대하여 부호화부의 개략도를 나타낸다. 문턱값은 최대길이의 런을 표현하는 데 요구되어지는 이진수의 길이이다. 런의 길이가 문턱값보다 크면 길이는 이진숫자로 표현이 되어지고 작거나 같은 경

우에는 비트의 열로서 표현되어진다. 삼각형 신장나무는 런부분과 상태부분으로 구성되어 있다. 상태부분은 런비트, 잎비트 그리고 진행비트로 구성된다. 여기서 런비트와 잎비트는 정점 신장나무의 것과 동일하고 진행비트는 Taubin의 것과 동일하다. 런부분은 정점 신장나무의 것과 같다.

2. 제안한 기하학 정보의 부호화 방법

3차원 물체의 기하학 정보는 3차원 좌표축에 32비트 부동소수점을 할당하여 96비트로 하나의 정점 위치를 표현된다. 하지만 이러한 정밀도는 너무 높아서 정밀도를 감소시켜도 크게 시각적으로 거슬리지 않는다. 따라서 기존의 방법에서는 양자화 기법과 DPCM, 그리고 예측 잔여 벡터 양자화 기법등이 제안되었다. 기하학 정보인 3차원 공간의 정점의 위치는 그 자체가 3차원 벡터로서 표현이 된다. 그리고 계층 분해방법으로 분해된 3차원 물체의 정점은 정점 신장나무를 깊이 우선 탐색을 이용하여 저장되어 있다. 이렇게 저장된 정점들의 인접성은 3차원공간에서의 근접성과 많은 상관성을 보여준다. 이러한 특성들은 벡터 양자화가 기하학 정보의 부호화에 적합하다는 것을 보여 준다.

본 논문에서는 벡터 양자화 기법을 이용하여 기하학 정보를 부호화하였다. 일반적인 벡터 양자화 기법은 코드북을 Linde, Buzo, Gray[8]의 근접화방법으로 구성하고 입력을 받아서 입력과 가장 가까운 코드북의 벡터값을 선택하여 그 선택된 벡터의 인덱스를 복호화부으로 전송한다. 복호화부에서 인덱스를 받아 코드북에서 그 입력에 맞는 값을 복원하는 과정으로 이루어진다.

여러 벡터 양자화 기법 중 본 논문에서는 많은 코드워드를 적은 비트로 부호화를 할수 있고 국부적인 상관성에 적합한 특성을 가진 유한상태 동적 벡터 양자화를 기하학 정보 부호화에 적용하였다. 기존의 2차원 영상 부호화에서 유한상태 동적 벡터 양자화는 영상의 국부적인 상관성을 고려하여 입력에 대해 여러개의 제한된 상태로 나누어 입력의 국부적인 특성을 예측하여 특정상태를 결정하고 상태에 따라 코드북을 구성하였다. 이때 상태예측에 따른 코드북의

구성을 동적으로 생성하였다. 일반적인 2차원 영상에서의 유한상태 동적 벡터 양자화부호기의 기본구조는 아래 그림과 같다.

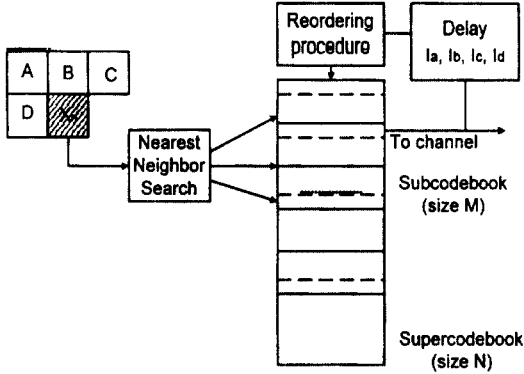


그림 8. 유한상태 동적 벡터 양자화부호화단의 기본구조
Fig. 8. Basic structure of DFSVQ encoder.

그림 8에서 주코드북은 일반적인 LBG방법으로 구성하고 부코드북은 입력 X_n 이 속하는 상태를 예측하기 위해 재배열 과정에서 상태를 예측하여 동적으로 구성한다.

$$I_n = \phi(X_n, S_n) \tag{11}$$

벡터 인덱스 I_n 은 예측된 상태 S_n 과 입력 X_n 에 의해 결정된다. 예측된 상태에 대한 부코드북 S_n 을 동적으로 생성하여 입력 X_n 과 가장 가까운 벡터값을 부코드북에서 찾는다.

벡터 인덱스 I_n 은 예측된 상태 S_n 과 입력 X_n 에 의해 결정된다. 예측된 상태에 대한 부코드북 S_n 을 동적으로 생성하여 입력 X_n 과 가장 가까운 벡터값을 부코드북에서 찾는다.

여기서 예측된 상태에 따라 부코드북 S_n 을 구성하는 것은 재배열 과정(reordering procedure)에서 상태에 예측함수(next-state-function)의해 동적으로 생성된다.

$$S_n = f(I_a, I_b, I_c, I_d) \tag{12}$$

f 는 입력 X_n 의 주변값인 이전에 양자화한 값 I_a, I_b, I_c 그리고 I_d 를 입력으로 상태를 결정한다.

$$\bar{X}_n = \psi(I_n, S_n) \tag{13}$$

부호화기에서는 부호화기의 재배열방법으로 부코드북을 동적으로 생성하고 부코드북에서 I_n 에 해당하는 벡터 X_n 을 구한다.

본 논문에서 유한상태 동적 벡터 양자화를 기반으로 한 기하학 정보의 부호화 방법이 제안되었다. 그림 9와 10은 각각 제안된 기하학 정보 부호화기와 복호화기를 보여준다.

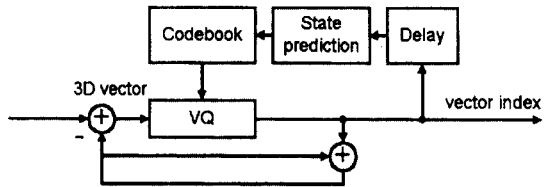


그림 9. 제안된 기하학 부호화기
Fig. 9. The proposed geometry encoder.

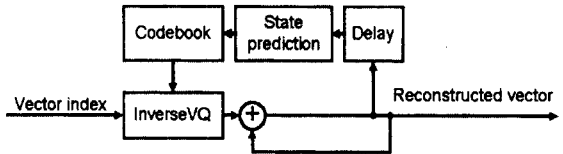


그림 10. 제안된 기하학 복호화기
Fig. 10. The proposed geometry decoder.

제안된 기하학 정보의 부호화는 다음과 같다. 정점의 좌표값은 인접한 점들 사이에 상관성을 제거하기 위해서 이전의 점과 입력 좌표값 사이의 차를 구하여 주코드북을 생성한다. 정점들간의 차를 입력으로 하였다. 입력이 들어오면 이전의 입력들로 재배열부에서 부코드북을 생성한다. 입력의 상태를 예측하여 재배열 하는 방법은 여러가지 방법이 있으나 본 논문에서는 이전의 벡터와 주코드북의 벡터간의 오차를 구

하여 오차가 적은 순서대로 벡터를 부코드북의 크기 만큼 선택하여 부코드북을 구성한다. 오차는 다음과 같이 정의 되어진다.

$$e_n = \sqrt{(X_n - X_{n-1})^2 + (Y_n - Y_{n-1})^2 + (Z_n - Z_{n-1})^2} \quad (14)$$

즉, 이전의 복원된 벡터가 현재 입력에 대한 부코드북을 구성하는 기준이 된다. 부코드북을 구성하는 과정 즉, 재배열 과정에서 입력의 상태예측이 맞지 않는 경우, 입력과 잘 맞지 않는 벡터로 부호화 될 수 있는데, 이런 경우 큰 오차를 발생시킬 수 있다. 입력의 상태예측이 맞지 않는 문제를 해결하기 위하여 본 논문에서는 오차의 크기에 따라서 주코드북과 부코드북을 섞어서 사용하여 오차를 줄였다. 부코드북에서 부호화된 벡터와 입력과의 오차와 주코드북에서 부호화된 벡터와 입력과의 오차 중 오차가 적은 벡터를 선택하여 복호화단으로 보내어준다. 그리고 코드북의 종류를 구분하기 위해서 하나의 비트를 이용하여 '0' 인 경우는 부코드북으로 부호화 되었음을 보여주고 '1' 인 경우는 주코드북으로 부호화 되었음을 나타낸다. 이와 같이 오차에 대해 적응적으로 코드북을 사용하여 전체오차를 줄였다. 복호화단에서는 입력의 플래그 비트(flag bit)로 인덱스가 주코드북의 것인지 부코드북의 것인지를 결정하여 주코드북의 것이면 주코드북에서 대응되는 벡터를 복원한다. 그러나, 부코드북의 인덱스이면 이전의 복원된 벡터를 기준으로 주코드북에서 오차가 적은 벡터들을 부코드북의 크기만큼 선택하여 부코드북을 재구성한다. 즉 부호화기의 재배열과정과 같은 과정을 거쳐서 인덱스와 대응되는 벡터를 복원한다.

본 논문에서 제안한 유한상태 동적 벡터 양자화를 기반으로 한 기하학 정보의 부호화는 많은 코드워드를 가지고 있으나 적은 비트로 코드워드를 나타내는 특성으로 인해 기존의 다른 방법에 비해 적은 오차를 보여주고 있다.

IV. 실험 및 결과

실험에서 3차원 물체는 Fandisk영상과 Triceratops

영상을 사용하였고 Fandisk영상은 삼각형 12,756개와 정점 6,574개로 구성되어져 있고, Triceratops영상은 삼각형 5,660개와 정점 2,832개로 구성되어 있다. 제안된 방법은 기존의 연결성 정보 부호화 방법에 비하여 Fandisk와 Triceratops에 대하여 적응적으로 비트를 할당하여 비트를 줄였다.



(a)



(b)

그림 11. Fandisk와 Triceratops의 원영상 (a) Fandisk (b) Triceratops

Fig. 11. Original image of Fandisk and Triceratops. (a) Fandisk (b) Triceratops

그림 11은 실험에 사용한 원영상이다. 그림 12와 그림 13은 기존의 방법과 제안한 방법의 기하학 정보 부호화 기법에 의한 Fandisk영상의 실험결과이다. 그림 12의 (a)와 (b)는 다양한 비트율에 대하여 DPCM으로 부호화한 결과이다. 3차원 좌표축에 대해 일정하게 양자화를 하므로 블록이 생김을 볼 수 있다. (c)와 (d)는 예측 잔여 벡터 양자화 방법으로 각각 다른 비트율로 부호화하여 복원한 결과이다. 그리고 (e)-(f)는 제안된 방법으로 부호화하여 복원한 영상이다. 평면이 고르지 못한 정도가 상대적으로 적다. 여기서 벡터 양자화의 코드북은 실험영상인 Fandisk영상을 가지고 생성하였다. 국부적인 상관성과 다양한 부코

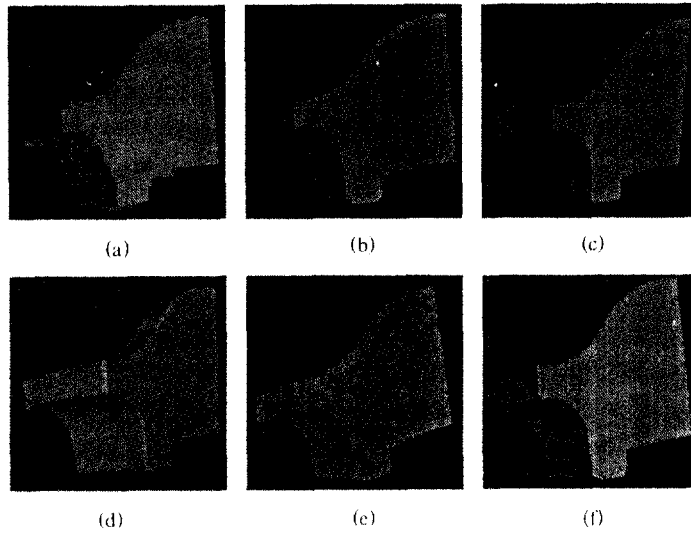


그림 12. Fandisk에 대한 복호화된 영상 (a) 9 bpv DPCM로 부호화 (b) 12 bpv DPCM로 부호화 (c) 9 bpv 예측잔여 벡터 양자화로 부호화 (d) 11 bpv 예측잔여 벡터 양자화로 부호화 (e) 8.50 bpv 유한상태 동적 벡터 양자화로 부호화 (f) 10.40 bpv 유한상태 동적 벡터 양자화로 부호화 (bpv : bit per vertex)

Fig. 12. Decoded results for Fandisk image. (a) 9 bpv by DPCM (b) 12 bpv by DPCM (c) 9 bpv by PRVQ (d) 11 bpv by PRVQ (e) 8.50 bpv by DFSVQ (f) 10.40 bpv by DFSVQ(a)(b)(c)(d)(e)(f)

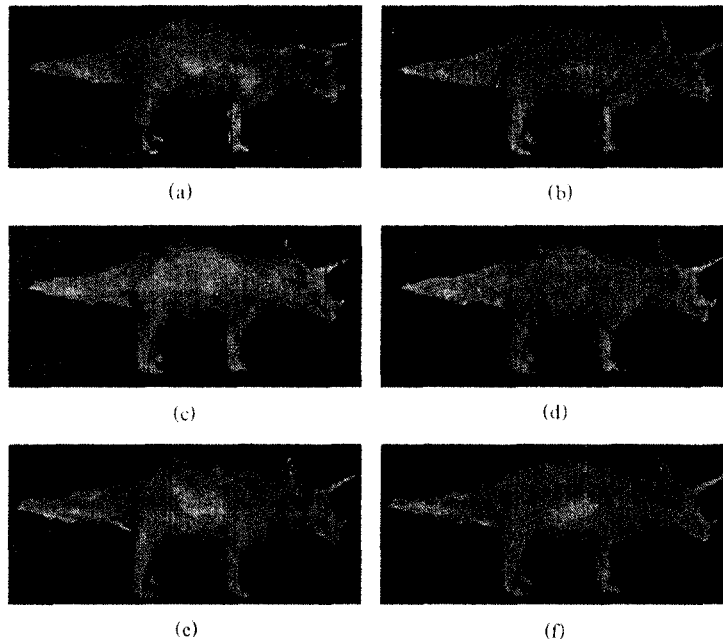


그림 13. 복호화된 Triceratops영상 (a) 9 bpv DPCM로 부호화 (b) 12 bpv DPCM로 부호화 (c) 9 bpv 예측잔여 벡터 양자화로 부호화 (d) 11 bpv 예측잔여 벡터 양자화로 부호화 (e) 8.38 bpv 유한상태 동적 벡터 양자화로 부호화 (f) 10.38 bpv 유한상태 동적 벡터 양자화로 부호화

Fig. 13. Decoded results for Triceratops image. (a) 9 bpv by DPCM (b) 12 bpv by DPCM (c) 9 bpv by PRVQ (d) 11 bpv by PRVQ (e) 8.38 bpv by DFSVQ (f) 10.38 bpv by DFSVQ

드북을 이용한 제안된 방법에 의한 실험결과가 기존의 방법보다 비트율이 작고 시각적으로 개선된 결과를 보여준다. 그림 13은 Triceratops에 대해 그림 12와 같이 실험한 결과이다. 여기서는 코드북을 Triceratops로 만든 코드북을 쓰지 않고 Fandisk의 코드북을 사용하여 자신의 코드북이 아닌 경우에 나타나 는 실험 결과를 비교하여 보았다. (a)와 (b)에서는 블록으로 화면이 깨어지는 것을 볼수 있다. 곡선은 블록으로 깨어지지 않지만 세부적인 부분에서는 블록으로 깨어진다. (c)와 (d)는 예측 잔여 벡터 양자화로 부호화한 영상이다. (e)와 (f)는 곡률적인 부분에서 DPCM보다 화질이 떨어지나 세부적인 부분에서는 블록으로 깨어지는 오차는 없고 예측 잔여 벡터 양자

화보다 오차가 적다. 그리고 전체 오차에서 가장 향상된 결과를 보여준다. 전체적으로 자신의 영상으로 구성되지 않은 코드북을 사용하여도 전체적인 오차가 제안한 방법이 가장 적을 것을 알수있다.

표 4는 연결성 정보의 부호화 결과를 보여주고 있다. 전체적인 비트량에 있어서 제안한 적응적 비트 할당 방법이 일정비트를 할당한 Taubin방법과 Bossen의 부호화방법에 비해 많게는 2배 정도의 비트율의 향상을 보여준다. 이는 계층적 분해에 의해 발생된 길이가 일정하지 않는 런의 부호화에 적응적 비트 할당이 적합함을 보여준다. 그리고 표 5와 표 6은 기하학 정보의 부호화의 결과이다. 다양한 방법에 대하여 다양한 비트율에 대해 제안한 방법의 비트율이 낮음에도 불구하고 전체적인 오차는 DPCM과 예측 잔여 벡터 양자화 방법에 비하여 적음을 보여준다. 이는 유한상태 동적 벡터 양자화 기법의 국부적 상관성에 따른 적절한 코드북의 생성과 많은 코드워드를 가지고 낮은 비트율로 부호화하는 특성이 기하학 정보의 부호화에 적합함을 보여준다.

표 4. 연결성 정보의 부호화 결과

Table 4. Compression results for connectivity information

| 3D model | Method | Triangle tree (bit) | Vertex tree (bit) | Total bit |
|-------------|----------|---------------------|-------------------|-----------|
| Fandisk | EPFL | 25892 | 12948 | 38840 |
| | IBM | 18041 | 845 | 18888 |
| | Proposed | 14941 | 459 | 15400 |
| Triceratops | EPFL | 11320 | 5882 | 18982 |
| | IBM | 8500 | 1210 | 9710 |
| | Proposed | 7592 | 978 | 8588 |

표 5. Fandisk에 대한 기존의 방법과 제안한 방법의 결과 비교

Table 5. Comparison of simulation results between the conventional methods and the proposed method for Fandisk

| Method | Bpv | Distortion |
|--------|-------|------------|
| DPCM | 9.00 | 421.40 |
| | 12.00 | 255.64 |
| PRVQ | 9.00 | 43.54 |
| | 11.00 | 22.47 |
| DFSVQ | 8.50 | 36.81 |
| | 10.40 | 16.85 |

표 6. Triceratops에 대한 기존의 방법과 제안한 방법의 결과 비교

Table 6. Comparison of simulation results between the conventional methods and the proposed method for Triceratops

| Method | Bpv | Distortion |
|--------|-------|------------|
| DPCM | 9.00 | 905.13 |
| | 12.00 | 520.34 |
| PRVQ | 9.00 | 657.92 |
| | 11.00 | 476.85 |
| DFSVQ | 8.38 | 488.17 |
| | 10.38 | 388.92 |

V. 결 론

본 논문에서는 3차원 합성영상 부호화기법에 대한 새로운 알고리즘을 제안하였다. 연결성 정보의 부호화에 있어서 런의 길이에 따라 최대길이의 런을 표현하기 위해 할당된 비트를 문턱값으로 하여 런의 길이가 문턱값보다 길면 문턱값 만큼의 비트를 할당하여 이진숫자로 길이를 표현하고 작으면 길이만큼의 비트를 할당하여서 비트열로서 그 런의 길이를 표현하여 적응적으로 과잉비트를 줄여 기존의 방법보다 많은 전송 정보량의 감소를 나타낸다.

기하학 정보의 부호화에서는 정점 신장나무의 깊이 우선 탐색의 순서대로 저장된 정점의 국부적인 상관성을 이용하여 유한상태 동적 벡터 양자화를 적용하여 부호화를 하였다. 3차원 정점정보는 정점 자체가 3차원 벡터이므로 벡터 양자화의 개념에 적합하다. 이러한 유한상태 동적 벡터 양자화는 기존의 방법들 보다 비트율과 오차에 있어서 많이 개선이 되었으며 또한 개선된 시각적 화질을 보여준다.

VI. 참고 문헌

1. Michael Deering, "Geometric Compression," Computer Graphics (Proc. SIGGRAPH), pp.13-20, Aug. 1995.
2. Gabriel Taubin and Jarek Rossignac, "Geometric compression through topological surgery," Technical Report RC20340, IBM Research Division Jan. 1996.
3. Frank Bossen, "Efficient representation of triangulations," Contribution Document number MPEG97/M1784, Seville Meeting of ISO/IEC JTC1/SC29/WG11 Feb. 1997.
4. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," Computer Graphics (Proc. SIGGRAPH), pp.19-26, Aug. 1993.
5. Jin-soo Choi, Myoung-Ho Lee and Chietek Ahn "Geometry compression using predictive vector quantization," Contribution Document number MPEG97/M-1744, Seville Meeting of ISO/IEC JTC1/SC29/WG11 Feb. 1997.
6. W. T. Chen, R. T. Chang, and J. S. Wang, "Image Sequence Coding Using Adaptive Finite-State Vector Quantization," IEEE Trans on circuits and systems for video technology, vol. 2. no. 1, Mar. 1992.
7. N. M. Nasrabadi, S. A. Rizvi, Next-State Functions for Finite-State Vector Quantization, IEEE Trans on image processing vol. 4. no. 12. Dec. 1995.
8. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans on Commun, vol. COM-28, no. 1, pp.84-95, Jan. 1980.
9. M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution Analysis of Arbitrary Meshes," Computer Graphics (Proc. SIGGRAPH), pp.173-182, Aug. 1995.
10. Jin-Woo Park, Kun-Woen Song, Jae-Yeal Nam, and Yeong-Ho Ha, "Connectivity compression for 3D object based on adaptive representation of spanning tree's run," Contribution Document number MPEG97/M2332, Stockholm Meeting of ISO/IEC JTC1/SC29/WG11 July 1997.
11. Jin-Woo Park, Kun-Woen Song, Jae-Yeal Nam, and Yeong-Ho Ha, "Connectivity and geometry compression of 3D object based on adaptive bits allocation and DFSVQ," Picture Coding Symposium, pp.479-483, Sep. 1997.



박진우 (Jin Woo Park) 정회원
 1974년 1월 23일생
 1996년 2월: 경북대학교 공과대학
 전자공학과 졸업(공학사)
 1998년 2월: 경북대학교 대학원
 전자공학과 졸업(공학석사)

1998년 3월~현재: 경북대학교 대학원 전자공학과
 박사과정
 <연구분야> 영상부호화, 신호처리, 컴퓨터 그래픽스등

송근원 (Kun Woen Song) 정회원
 1968년 7월 1일생
 한국통신학회 제 22권 4호 참조

박영식 (Young Sik Park) 정회원
 1969년 4월 18일생
 한국통신학회 제 22권 4호 참조
 현재: 삼성전자주식회사 연구원

남재열 (Jae Yeal Nam) 정회원
 한국통신학회 제 22권 4호 참조

하영호 (Yeong Ho Ha) 정회원
 한국통신학회 제 22권 4호 참조
 현재: 경북대학교 전자전기공학부 교수