



일반논문 (Regular Paper)

방송공학회논문지 제29권 제1호, 2024년 1월 (JBE Vol.29, No.1, January 2024)

<https://doi.org/10.5909/JBE.2024.29.1.42>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

비디오 기반 동적 메쉬 압축을 위한 정점 차수 기반 리프팅 웨이블릿 변환 방법

김민태^{a)}, 변주형^{a)}, 심동규^{a)†}

Valence based Lifting Wavelet Transform for Video-based Dynamic Mesh Compression

Mintae Kim^{a)}, Joohyung Byeon^{a)}, and Donggyu Sim^{a)†}

요약

본 논문에서는 동적 메쉬 부/복호화 시 정점 valence 기반의 리프팅 웨이블릿 변환 방법을 사용하여 동적 메쉬의 변위 벡터를 압축하는 방법을 제안한다. 제안하는 방법에서는 메쉬의 각 정점과 연결된 간선의 개수를 의미하는 valence에 기반하여 각 정점의 리프팅 업데이트 가중치를 결정해 업데이트 과정을 수행한다. 기존 고정 가중치를 사용한 업데이트 방법은 각 정점의 valence에 따라 업데이트 필터 특성이 바뀌는 반면, 제안하는 업데이트 방법은 정점의 업데이트 가중치를 valence 값으로 정규화 해줌으로써 모든 정점이 동일 특성의 업데이트 필터를 갖도록 한다. 제안하는 방법과 기존 기술인 LoD 기반 업데이트 방법을 적용하여 리프팅 변환 업데이트를 수행한 결과, MPEG V-DMC 참조 소프트웨어 TMM v4.0의 기존 압축 성능 대비 AI, LD 환경에서 별도의 부·복호화 시간 증가 없이 point cloud 기반의 BD-rate (D1 PSNR)가 각각 평균 -3.9%, -3.6% 향상되어 효과적으로 메쉬 변위 벡터의 압축이 가능함을 확인하였다.

Abstract

In this paper, we propose a vertex valence-based lifting wavelet transform method for displacement vector compression in dynamic mesh. The proposed method performs a lifting transform update process using the weight derived from vertex valence, which is the number of edges incident to that vertex. In terms of filtering, proposed method assigns an update filter with the identical characteristics to all displacement vectors by normalizing the update weight to the valence, while the update method with fixed weight assigns different update filters to displacement vectors according to the valence of that vertex. As a result of performing lifting transform update process based on proposed method and LoD scaling method which is preceding studied, the experiment result shows the improved performance with the average of -3.9% and -3.6% in terms of point cloud-based BD-rate (D1 PSNR) in AI and LD conditions compared to performance of original V-DMC reference software TMM v4.0 with no additional complexity.

Keywords : V-DMC, Mesh compression, Lifting wavelet transform, Vertex valence

Copyright © 2024 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

1. 서론

최근 3차원 데이터를 획득, 모델링, 렌더링하는 기술이 발전함에 따라, 증강현실 (AR, Augmented Reality), 가상현실 (VR, Virtual Reality), 메타버스, 자율주행 등의 다양한 분야에서 3차원 데이터에 관한 연구가 증가하고 있다. 3차원 데이터의 대표적인 예시 중 하나인 메쉬는 3차원 체적 데이터 (3D volumetric data)를 다각형들의 집합으로 표현한다. 메쉬는 3차원 공간상의 위치를 표현하는 정점 정보, 다각형을 구성하기 위한 정점 간 연결 정보, 3D 데이터 표면의 색상 정보를 2D 데이터로 저장하는 텍스처 맵, 메쉬 표면과 텍스처 맵 간의 맵핑 정보 등으로 구성된다. 메쉬의 구성요소 중 하나 이상이 시간의 흐름에 따라 변화하는 경우, 해당 메쉬를 동적 메쉬라고 정의할 수 있다^[1].

메쉬는 일반적으로 2D 데이터 대비 매우 큰 데이터양을 갖기 때문에, 여러 분야에서 메쉬 데이터의 전송 및 저장을 위해서는 효과적인 압축 기술이 요구된다. 이와 같은 요구에 따라, 메쉬 압축에 대한 표준화 과정이 활발하게 진행되어 왔다. 국제 표준화 기구 ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) JTC1 (Joint Technical Committee 1) SC29 (Sub-Committee 29) WG7 (Working Group 7) MPEG (Moving Picture Experts Group)의 3DGC (3D Graphics Coding) 서브 그룹에서는 애니메이션 모델과 같은 3차원 그래픽스 콘텐츠의 압축 및 처리를 위한 MPEG-4 파트 16 AFX (Animation Framework eXtension)의 표준화를 2004년에 완료하였다^[2]. 이후 2009년, 시간 흐름에 따라 정점 정보가 변하는 동적 메쉬를 압축하기 위한 FAMC (Frame-based Animated Mesh Compression) 표준을 발행

하였다^[3]. 하지만 FAMC는 시간 흐름에 따라 연결 정보, 텍스처 맵 정보 등이 변하는 동적 메쉬에 대해서 압축 표준을 지원하지 않았기 때문에, 이러한 동적 메쉬에 대한 압축 표준의 필요성이 요구되었다. 이에 따라 국제 표준화 기구 ISO/IEC JTC1 SC29 WG7 MPEG의 3DGH (3D Graphics and Haptics Coding) 서브 그룹에서는 동적 메쉬 압축을 위한 V-DMC (Video-based Dynamic Mesh Coding)의 표준화를 시작하였다. 2021년 10월 동적 메쉬 압축에 대한 CfP (Call for Proposals)를 시작으로^[4], 2023년 7월 WD (Working Draft) 단계에서 활발히 논의가 진행되고 있으며^[5], 2024년 10월 표준 완료를 목표로 하고 있다. V-DMC 표준화 과정은 CfP에 대한 응답으로 제출된 기술 중 세분화 방법 기반 메쉬 압축 구조^[6]를 기반으로 하여 표준화가 진행되고 있다. 해당 압축 구조는 입력 메쉬를 단순화하여 생성한 저해상도 베이스 메쉬와 베이스 메쉬를 세분화하여 생성한 메쉬의 각 정점이 원본 메쉬와 유사해지도록 보정해주는 변위 벡터를 기반으로 메쉬를 압축한다. 결과적으로, 현재 논의가 되고 있는 V-DMC 기술은 베이스 메쉬, 변위 벡터, 텍스처 맵 등을 생성 및 부·복호화하여 메쉬의 압축과 복원을 수행한다. 이때 변위 벡터는 세분화된 베이스 메쉬의 각 정점에서 원본 메쉬 표면과의 차이로 계산되며, 세분화 과정에서 생성된 LoD (Level of Details) 구조로 분류된다. 이후 변위 벡터는 LoD 단위를 기준으로 예측과 업데이트 과정을 포함하는 리프팅 웨이블릿 변환을 거쳐 부·복호화가 수행된다.

본 논문에서는 변위 벡터의 부·복호화 간 수행되는 리프팅 웨이블릿 변환 과정에서 각 정점의 valence에 기반하여 업데이트를 수행하는 방법을 제안한다. 이때 정점의 valence는 해당 정점과 연결된 간선의 개수라고 정의할 수 있다^[7]. 제안하는 방법은 리프팅 변환의 업데이트 과정에서 모든 정점에 대해 고정 가중치를 통해 업데이트를 수행하는 것이 아닌, 현재 정점의 valence 값으로 정규화한 가중치를 사용하여 업데이트를 수행한다. 제안하는 valence 기반의 가중치 할당은 각 정점의 연결 정보를 고려하여 업데이트를 수행함으로써 모든 정점에서 동일한 업데이트 필터 특성을 갖게 하며, 정점의 연결 정보에 따라 업데이트 필터 특성이 바뀌는 고정 가중치 업데이트 방법 대비 제안하는 방법을 사용함으로써 변위 벡터 압축 성능 향상이 가능함

a) 광운대학교 컴퓨터공학과(Kwangwoon University)

‡ Corresponding Author : 심동규(Donggyu Sim)

E-mail: dgsim@kw.ac.kr

Tel: +82-2-941-6470

ORCID: <https://orcid.org/0000-0003-3710-7314>

※ 본 연구는 LG전자의 실시간 XR 및 메타버스 서비스를 위한 Point Cloud Mesh 코딩 기술 개발, 2023년도 광운대학교 우수연구자 지원 사업, 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2021R1A2C2092848)의 연구결과로 수행되었음.

· Manuscript November 22, 2023 ; Revised December 13, 2023;

Accepted December 13, 2023.

을 성능 비교를 통해 증명한다. 본 논문의 구성은 다음과 같다. 2장에서는 V-DMC 기술과 선행 연구인 LoD 기반 리프팅 변환 업데이트 방법에 대해 설명한다. 3장에서는 V-DMC 기술의 리프팅 변환 과정을 분석하고 제안하는 정점 valence 기반 리프팅 변환 업데이트 방법에 대해 설명한다. 4장에서는 제안하는 리프팅 변환 업데이트 방법의 성능을 비교하여 평가한다. 마지막으로, 5장에서 본 논문에 대한 결론을 맺는다.

II. 관련 기술

1. Video-based Dynamic Mesh Compression

V-DMC 기술의 기본 아이디어는 입력 메쉬를 단순화하여 정점의 수를 크게 줄인 베이스 메쉬와 베이스 메쉬의 세분화 및 원본 메쉬와의 피팅 과정을 통해 생성한 정점별 변위 벡터를 통해 메쉬를 압축 및 복원하는 것이다. V-DMC의 부호화 과정은 전처리 과정과 베이스 메쉬 부호화, 변위 벡터 부호화, 텍스처 맵 부호화 과정으로 구성되며, 그림1과 같은 구조를 갖는다.

전처리 과정은 메쉬 단순화, 아틀라스 매개변수화, 세분화 표면 피팅 과정으로 구성된다. 먼저 메쉬 단순화 과정에서는 입력 메쉬에 메쉬 단순화 기술을 적용하여 입력 메쉬 대비 정점의 수를 크게 줄인 베이스 메쉬를 생성한다. 아틀라스 매개변수화 과정에서는 UVAtlas^[8] 툴과 같은 메쉬 매개변수화 기술을 사용하여 메쉬 단순화 과정을 통해 생성된 베이스 메쉬의 텍스처 좌표를 재계산한다. 마지막으로 세분화 표면 피팅 과정에서는 메쉬 세분화 기술을 사용하여 베이스 메쉬를 세분화하고, 원본 메쉬의 표면과 유사해지도록 세분화된 베이스 메쉬의 각 정점을 피팅하는 과정을 수행한다.

베이스 메쉬 부호화 과정에서는 베이스 메쉬의 예측 모드에 따라 정적 메쉬 부호화 과정 또는 움직임 벡터 부호화 과정이 수행된다. 베이스 메쉬가 화면 내 예측 모드로 부호화되는 경우 정적 메쉬 부호화 과정을 수행한다. 정적 메쉬 부호화 과정에서는 전처리 과정에서 생성된 베이스 메쉬의 정점 정보, 연결성 정보 등이 EdgeBreaker^[9] 기술을 통해 부호화된다. V-DMC에서는 현재 베이스 메쉬와 참조 베이스 메쉬의 정점 수, 연결 정보 등이 모두 동일하고, 각 정점의 위치 정보만 다른 경우 두 베이스 메쉬가 일대일 대응 관계라고 정의한다. 일대일 대응 관계가 성립하는 경우 현

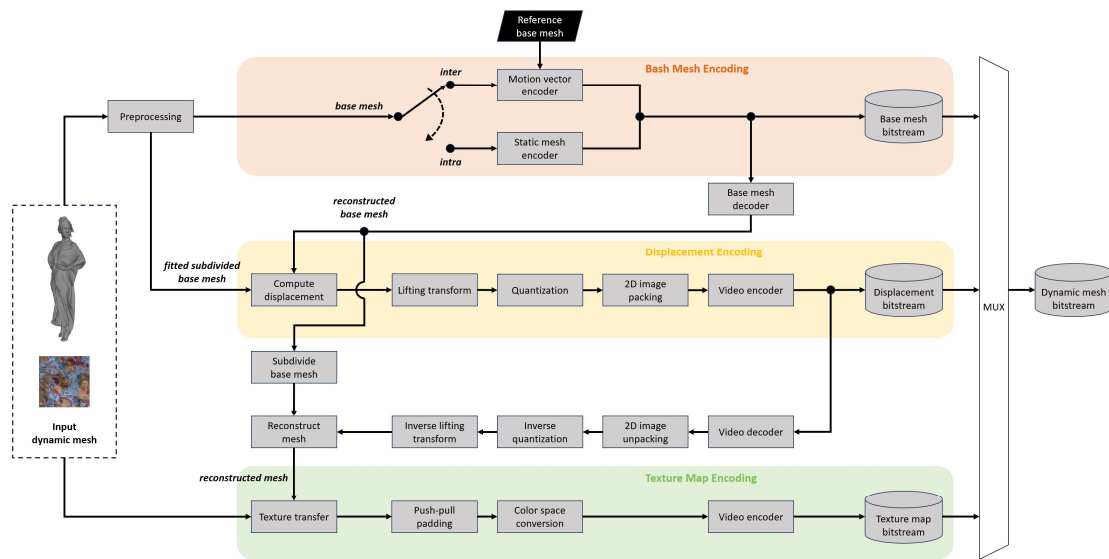
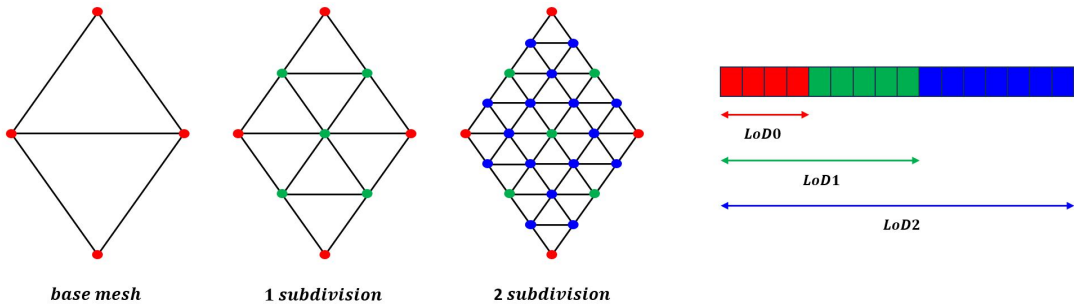


그림 1. V-DMC 부호화기 블록도
Fig. 1. Block diagram of V-DMC encoder

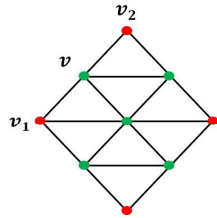
재 베이스 메쉬는 화면 간 예측 모드로 부호화되며, 움직임 벡터 부호화 과정을 수행한다. 움직임 벡터 부호화 과정에서는 현재 베이스 메쉬와 참조 베이스 메쉬의 정점별 위치를 차분하여 계산한 움직임 벡터의 부호화를 수행한다.

변위 벡터 부호화 과정에서는 먼저 부·복호화 과정을 거쳐 복원된 베이스 메쉬와 전처리 과정에서 세분화와 피팅 과정을 거쳐 생성된 베이스 메쉬의 정점별 위치를 차분하여 변위 벡터를 계산한다. 계산된 변위 벡터는 좌표계의

에너지 밀집도를 높이기 위해 (normal, tagential, bi-tangential)로 구성된 지역 좌표계로 변환된다. 변위 벡터는 세분화 과정에서 생성된 LoD (Level of Details) 구조로 분류되며, 그림2와 같이 LoD 단위로 리프팅 웨이블릿 변환^[10] 과정이 수행된다. 리프팅 웨이블릿 변환 과정은 예측 과정과 업데이트 과정으로 구성되며, 예측 과정에서는 하위 LoD의 변위 벡터를 사용해 상위 LoD의 변위 벡터를 예측하여 잔차신호를 생성하고, 업데이트 과정에서는 추후 예측 과



(a) LoD structure of V-DMC



1. Prediction process

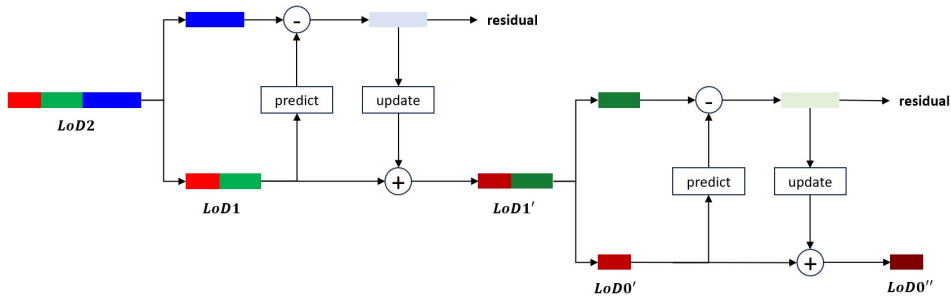
$$Signal(v) = Signal(v) - \frac{1}{2}(Signal(v_1) + Signal(v_2))$$

2. Update process

$$Signal(v_1) = Signal(v_1) + \frac{1}{8}Signal(v)$$

$$Signal(v_2) = Signal(v_2) + \frac{1}{8}Signal(v)$$

(b) Lifting wavelet transform process



(c) Displacement encoding process

그림 2. V-DMC 리프팅 웨이블릿 변환 수행 과정
 Fig. 2. V-DMC lifting wavelet transform process

정을 위해서 현재 예측 과정에서 생성된 상위 LoD의 잔차 신호로 예측에 사용된 하위 LoD의 변위 벡터를 보정해주는 동작을 수행한다. 리프팅 웨이블릿 변환 과정을 통해 생성된 변환 계수는 양자화된 후 2D 이미지로 패키징 되어 HEVC (High Efficiency Video Coding)^[11], VVC (Versatile Video Coding)^[12]와 같은 비디오 코덱을 통해 부호화된다.

텍스처 맵 부호화 과정에서는 원본 텍스처 맵을 복원된 메쉬에 맞춰 재생성하는 텍스처 진이 알고리즘이 수행된다. 이후 재생성된 텍스처 맵에서 텍스처가 존재하는 공간과 존재하지 않는 공간 사이의 불연속성을 줄이기 위해 반복적으로 다운 샘플링 및 업 샘플링을 수행하면서 빈공간을 채워나가는 푸쉬-풀 패딩^[13]이 수행된다. 패딩 된 텍스처 맵은

YUV 포맷으로 변환된 후 비디오 코덱을 통해 부호화된다.

V-DMC의 복호화 과정은 비트스트림으로부터 베이스 메쉬, 변위 벡터, 텍스처 맵을 복호화함으로써 수행되며, 그림3과 같은 구조를 갖는다. 베이스 메쉬가 화면 내 예측 모드로 부호화된 경우에는 정적 메쉬 복호화 과정을 통해 현재 베이스 메쉬를 복호화한다. 베이스 메쉬가 화면 간 예측 모드로 부호화된 경우에는 움직임 벡터 복호화 과정을 통해 정점별 움직임 벡터를 복호화한 뒤, 참조 베이스 메쉬에 적용하여 현재 베이스 메쉬를 복호화한다. 변위 벡터는 비디오 코덱을 통해 2D 패키징 되고 양자화된 변위 벡터를 복호화한 후, 부호화기 과정과 반대로 역패킹, 역양자화, 리프팅 웨이블릿 역변환을 수행하여 복원된다. 베이스 메쉬와 변

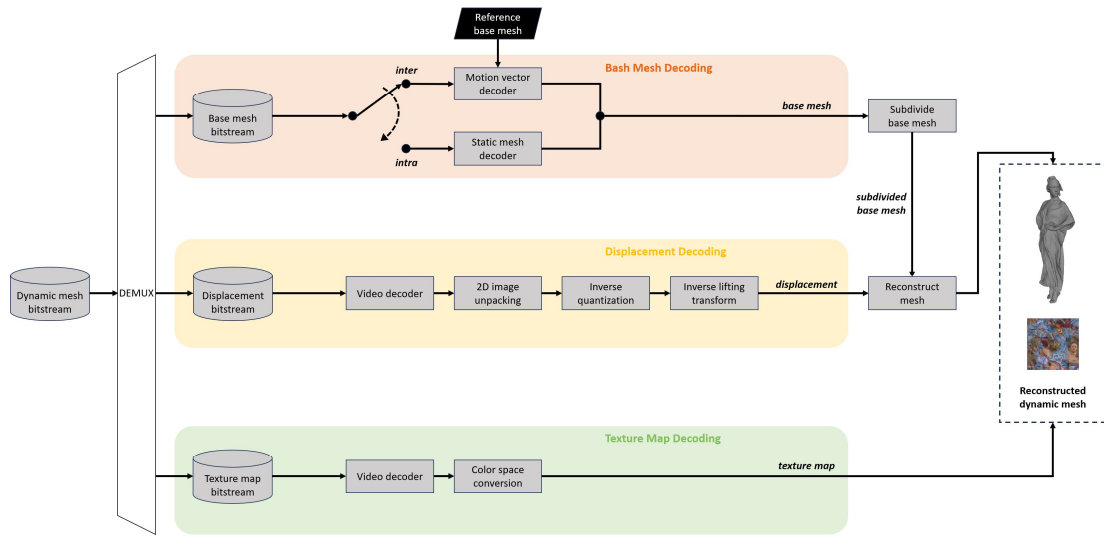


그림 3. V-DMC 복호화기 블록도
 Fig. 3. Block diagram of V-DMC decoder

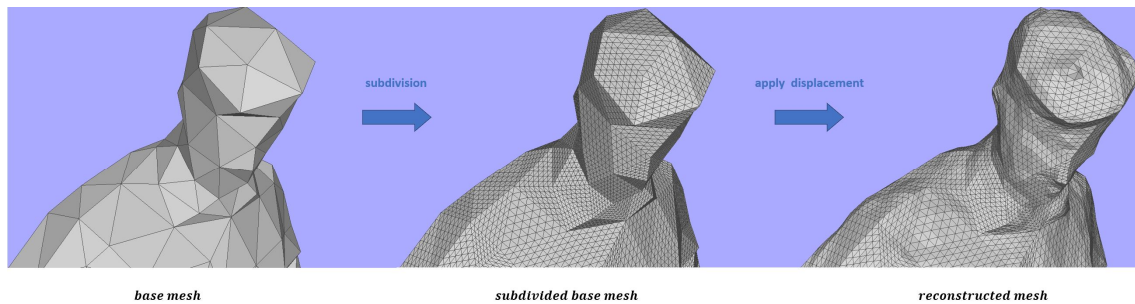


그림 4. 세분화된 베이스 메쉬에 변위 벡터가 적용되는 예시
 Fig. 4. Example of applying displacement vector to subdivision base mesh

위 벡터가 복원된 후에는 부호화기와 동일한 메쉬 세분화 기술을 적용하여 베이스 메쉬를 세분화한 뒤 정점별로 변위 벡터를 적용하여 메쉬를 복원한다^[14]. 베이스 메쉬를 세분화하고 변위 벡터가 적용되는 예시는 그림4와 같다. 텍스처 맵은 비디오 코덱을 통해 YUV 포맷의 텍스처 맵을 부호화한 후, RGB 포맷으로 변환되어 복원된다.

2. 리프팅 웨이블릿 변환 간 LoD 기반 적응적 업데이트 방법

현재 V-DMC^[5]의 리프팅 웨이블릿 변환 과정은 상위 LoD 변위 벡터를 인접한 두 하위 LoD 변위 벡터의 평균값으로 예측하여 잔차신호를 생성하는 예측 과정과 생성된 잔차신호에 고정된 가중치 0.125를 곱해 인접한 두 하위 LoD 변위 벡터를 보상해 주는 업데이트 과정으로 구성된다. 즉, 예측 과정과 업데이트 과정에서 각각 고정 가중치 0.5와 0.125를 사용한다. 예측 가중치의 경우 베이스 메쉬가 인접한 두 하위 LoD 정점의 평균 위치로 새로운 상위 LoD 정점을 생성하는 midpoint 세분화 방식으로 모든 세분화 과정을 수행하기 때문에, 상위 LoD부터 하위 LoD까지 반복되는 리프팅 변환의 모든 과정에서 고정된 가중치 0.5

를 사용하는 것이 적합할 수 있다. 하지만 업데이트 가중치의 경우 세분화 과정에서 생성된 각 LoD가 서로 다른 특성을 가질 수 있기 때문에, 리프팅 변환의 모든 과정에서 동일한 고정 가중치를 사용하는 것은 최적이지 아닐 수 있다. 이에 따라 리프팅 변환 과정이 수행되는 현재 LoD에 기반하여 적응적으로 업데이트를 수행하는 방법이 제안되었다^[15]. 해당 기술에서는 그림5와 같이 상위 LoD에서 하위 LoD로 갈수록 정점 간 거리가 멀어진다는 특성을 고려하여, LoD 단위로 리프팅 변환이 수행됨에 따라 일정 비율로 업데이트 가중치를 점차 감소시켜 사용하는 것을 제안하고 있다. 제시된 업데이트 가중치는 최초 상수 가중치 0.25부터 시작하여서 한 LoD에 대한 리프팅 변환이 수행될 때마다 50%씩 감소한 값을 사용한다. 예를 들어, 부호화기에서 세분화 횟수가 3번으로 입력되어 LoD0 ~ 3 구조를 갖는 경우, LoD3 변위 벡터에 대한 업데이트 가중치는 0.25, LoD2 변위 벡터에 대한 업데이트 가중치는 0.125, LoD1 변위 벡터에 대한 업데이트 가중치는 0.0625가 된다. LoD 기반 적응형 리프팅 변환 업데이트 방법의 성능은 V-DMC 참조 소프트웨어 TMM v4.0^[16] 대비 AI, LD 환경에서 point cloud 기반의 BD-rate (D1 PSNR)가 각각 평균 -3.2%, -3.0%의 향상된 결과를 보였다.

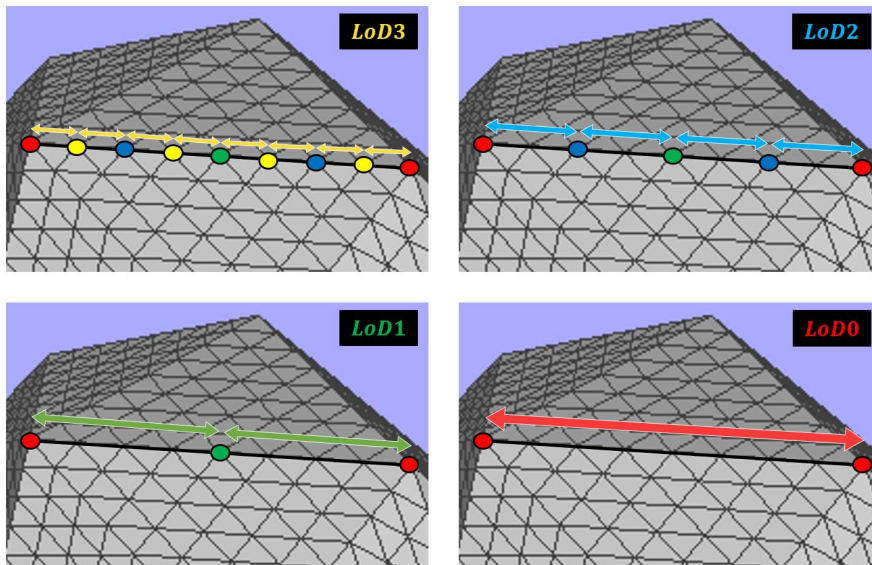
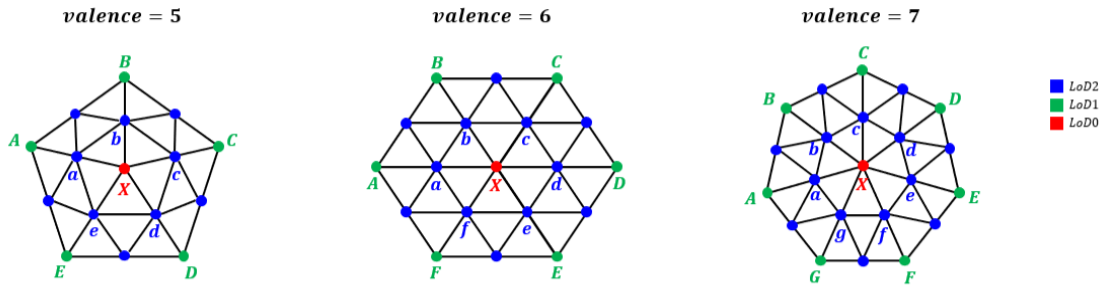


그림 5. LoD 별 정점의 거리 특성
 Fig. 5. Distance characteristic of vertex by LoD

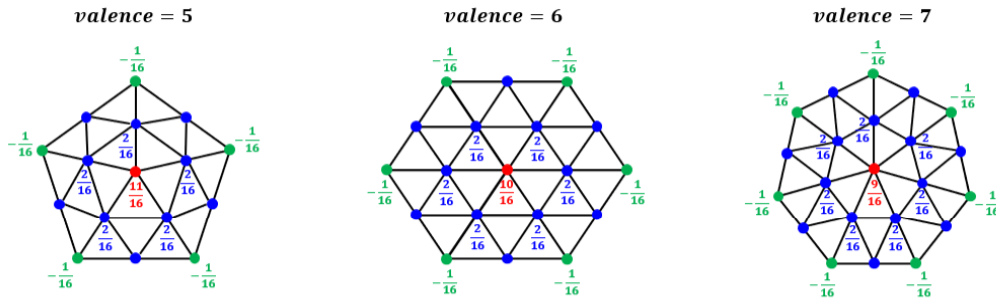
III. 제안 방법

본 절에서는 V-DMC 참조 소프트웨어 TMM v4.0^[16]을 기준으로 기존 V-DMC 기술의 리프팅 변환에 대해 분석하고, 제안하는 valence 기반 리프팅 변환 업데이트 방법을 설명한다. V-DMC 기술에서 리프팅 변환은 예측 과정과 업

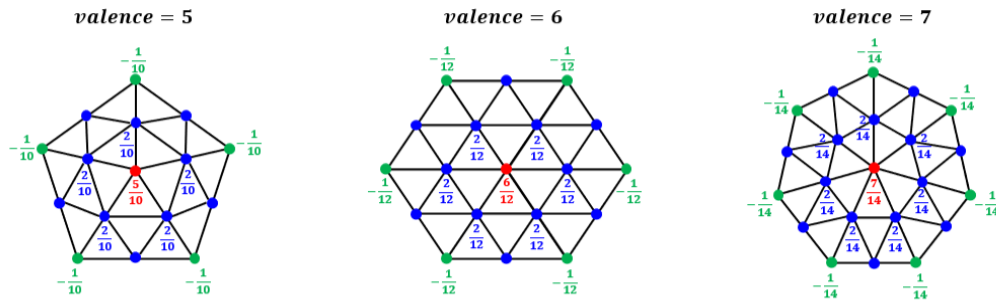
데이트 과정으로 구성된다. 예측 과정에서는 인접한 하위 LoD의 변위 벡터를 사용해 상위 LoD의 변위 벡터를 예측하여 상위 LoD 변위 벡터에 대한 잔차신호를 출력한다. 업데이트 과정에서는 이후 예측 과정을 위해 생성된 잔차신호로 예측에 사용된 하위 LoD의 변위 벡터를 보정하여 보정된 하위 LoD 변위 벡터를 출력한다. 이때, 예측 과정은



(a) Mesh vertex with different valence



(b) Filter coefficient of displacement X when updating with weight $\frac{1}{8}$



(c) Filter coefficient of displacement X when updating with weight $\frac{1}{valence}$

그림 6. 정점의 valence와 업데이트 가중치에 따른 V-DMC 리프팅 변환 업데이트 필터

Fig. 6. V-DMC lifting transform update filter based on vertex valence and update weight

예측 가중치를 통해 하위 LoD 변위 벡터로 생성되는 예측자를 결정하며, 업데이트 과정은 업데이트 가중치를 통해 하위 LoD 변위 벡터에 보정될 잔차신호의 크기를 결정한다. TMM v4.0에서는 모든 정점의 변위 벡터에 대해 고정된 예측 가중치 0.5와 고정된 업데이트 가중치 0.125를 사용하여 리프팅 변환을 수행한다. 그림6 (a)는 valence가 각각 5, 6, 7인 정점에 대한 메쉬의 일부분을 나타낸 그림이다. 해당 그림에서 각 문자는 해당 위치 정점의 변위 벡터를 의미하며, 문자 $a \sim g$ 는 LoD2의 변위 벡터를, 문자 $A \sim G$ 는 LoD1의 변위 벡터를, X 는 LoD0의 변위 벡터를 의미한다. TMM v4.0에서 고정된 예측 가중치 0.5를 사용하여 리프팅 변환 예측 과정을 수행하는 경우, LoD2 변위 벡터 $a \sim g$ 에 대한 예측 과정을 식 (1)과 같이 나타낼 수 있다.

식 (1)에서 $aa \sim gg$ 는 예측 과정에서 생성되는 상위 LoD의 잔차신호이다. 기존 TMM v4.0에서 고정된 업데이트 가중치 0.125를 사용하여 리프팅 변환 업데이트 과정을 수행하는 경우, LoD0 변위 벡터 X 에 대한 업데이트 과정을 식 (2)와 같이 나타낼 수 있다.

$$\begin{aligned}
 aa &= a - \frac{1}{2}A - \frac{1}{2}X & ee &= e - \frac{1}{2}E - \frac{1}{2}X \\
 bb &= b - \frac{1}{2}B - \frac{1}{2}X & ff &= f - \frac{1}{2}F - \frac{1}{2}X \\
 cc &= c - \frac{1}{2}C - \frac{1}{2}X & gg &= g - \frac{1}{2}G - \frac{1}{2}X \\
 dd &= d - \frac{1}{2}D - \frac{1}{2}X
 \end{aligned} \tag{1}$$

식 (2)에서 X' 는 업데이트가 수행된 후 보정된 LoD0 변위 벡터 X 를 의미한다. 결과적으로 업데이트 과정은 3차원

공간상에서의 저대역 통과 필터로 해석될 수 있다. 이때 식 (2)를 통해 고정 가중치 0.125를 사용하여 업데이트를 수행하는 경우 정점의 valence에 따라 현재 변위 벡터 성분 에 대한 필터 계수가 달라지며, 정점의 valence가 커질수록 유지되는 현재 변위 벡터 성분이 감소하는 것을 확인할 수 있다. 업데이트 식 (2)를 필터링 관점에서 표현하면 그림6 (b)와 같은 필터 계수로 나타낼 수 있으며, 정점의 valence에 따라 리프팅 변환 간 변위 벡터에 적용되는 필터의 특성이 달라지는 것을 확인할 수 있다. 리프팅 변환 업데이트 과정은 현재 예측 과정에서 생성된 잔차신호로 예측에 사용된 하위 LoD의 변위 벡터를 보정해줌으로써, 이후 예측 과정 성능에 영향을 준다^[17]. 즉, 현재 업데이트 과정에서 각 변위 벡터에 적용되는 필터 특성에 따라 이후 예측 과정의 성능이 결정된다. 앞서 기술한 것과 같이 고정 가중치를 사용한 업데이트 방법은 각 정점의 valence에 따라 해당 변위 벡터에 적용되는 업데이트 필터의 특성이 달라지며, 결과적으로 이후 예측 과정은 서로 다른 필터 특성으로 보정된 변위 벡터들에 대해 수행되게 된다.

본 논문에서는 리프팅 웨이블릿 변환 간 모든 변위 벡터에 대해 동일한 특성의 업데이트 필터를 적용하여 이후 예측 과정이 동일한 특성으로 보정된 변위 벡터들에 대해 수행되도록 하기 위해, 각 정점과 연결된 간선의 개수로 정의되는 valence에 기반하여 업데이트를 수행하는 방법을 제안한다. 제안하는 방법에서는 각 정점별 valence를 계산하고, 계산된 valence 값으로 정규화시킨 가중치를 사용하여 리프팅 변환의 업데이트 과정을 수행한다. 이때 각 정점별 valence 값은 변위 벡터의 리프팅 변환 이전 부·복호화 과정 간 동일하게 수행되는 베이스 메쉬의 세분화 과정 안에

$$\begin{aligned}
 \text{if } valence = 5, X' &= X + \frac{1}{8}(aa + bb + cc + dd + ee) \\
 &= \frac{11}{16}X + \frac{2}{16}(a + b + c + d + e) - \frac{1}{16}(A + B + C + D + E) \\
 \text{if } valence = 6, X' &= X + \frac{1}{8}(aa + bb + cc + dd + ee + ff) \\
 &= \frac{10}{16}X + \frac{2}{16}(a + b + c + d + e + f) - \frac{1}{16}(A + B + C + D + E + F) \\
 \text{if } valence = 7, X' &= X + \frac{1}{8}(aa + bb + cc + dd + ee + ff + gg) \\
 &= \frac{9}{16}X + \frac{2}{16}(a + b + c + d + e + f + g) - \frac{1}{16}(A + B + C + D + E + F + G)
 \end{aligned} \tag{2}$$

서 계산된 정보를 저장함으로써, 추가적인 연산 과정 없이 정점별 **valence** 정보를 리프팅 변환 업데이트 과정에 사용할 수 있다. 제안하는 정점 **valence** 기반 리프팅 변환 업데이트는 식 (3)과 같이 수행된다.

$$\begin{aligned} signal[v1] &= signal[v1] + \left(\frac{1}{valence[v1]}\right) \times signal[v] \\ signal[v2] &= signal[v2] + \left(\frac{1}{valence[v2]}\right) \times signal[v] \end{aligned} \quad (3)$$

식 (3)에서 v 는 현재 리프팅 변환 예측 과정에서 예측 대상이 된 상위 LoD의 정점이며, $v1$ 과 $v2$ 는 예측 과정에서 예측자로 사용되어 업데이트 대상이 되는 하위 LoD의 정점들이다. 또한 $signal$ 은 해당 정점의 변위 벡터 값을 의미하며, $valence$ 는 해당 정점의 **valence** 값을 의미한다.

그림6 (a)에서 정점 **valence** 기반 가중치를 사용해서 변위 벡터 X 를 업데이트 하는 과정은 식 (4)와 같이 나타낼 수 있다.

식 (4)를 통해 정점 **valence** 기반 가중치로 업데이트를 수행하는 경우에는, 정점의 **valence**와 관계없이 현재 변위 벡터 성분과 인접 변위 벡터 성분이 항상 동일한 비율로

반영되는 것을 확인할 수 있다. 항상 기존 변위 벡터 성분의 절반이 유지되며, 절반의 인접 변위 벡터 성분이 더해진다. 업데이트 식 (4)를 필터링 관점에서 표현하면 그림6 (c)와 같은 필터 계수로 나타낼 수 있으며, 모든 변위 벡터에 대해 현재 성분의 절반을 보존하는 특성을 가진 저대역 통과 필터가 동일하게 적용되는 것을 확인할 수 있다. 따라서 현재 업데이트 과정 이후 예측 과정은 동일한 특성으로 보정된 변위 벡터들에 대해 수행되어, 예측 성능의 향상을 기대할 수 있다. 정점의 **valence**에 기반하여 결정된 가중치에 더해, 식 (6)과 같이 특정 상수값 K 를 곱하여 업데이트 과정을 수행할 수 있다.

$$\begin{aligned} signal[v1] &= signal[v1] + \left(\frac{K}{valence[v1]}\right) \times signal[v] \\ signal[v2] &= signal[v2] + \left(\frac{K}{valence[v2]}\right) \times signal[v] \end{aligned} \quad (6)$$

그림6 (a)에서 **valence**가 6인 경우, 특정 상수값 K 에 따라 변위 벡터 X 를 업데이트 하는 과정은 식 (6)과 같이 나타낼 수 있다.

식 (5)를 통해 큰 상수값이 곱해질수록 유지되는 현재

$$\begin{aligned} \text{if } valence = 5, X' &= X + \frac{1}{5}(aa+bb+cc+dd+ee) \\ &= \frac{5}{10}X + \frac{2}{10}(a+b+c+d+e) - \frac{1}{10}(A+B+C+D+E) \\ \text{if } valence = 6, X' &= X + \frac{1}{6}(aa+bb+cc+dd+ee+ff) \\ &= \frac{6}{12}X + \frac{2}{12}(a+b+c+d+e+f) - \frac{1}{12}(A+B+C+D+E+F) \\ \text{if } valence = 7, X' &= X + \frac{1}{7}(aa+bb+cc+dd+ee+ff+gg) \\ &= \frac{7}{14}X + \frac{2}{14}(a+b+c+d+e+f+g) - \frac{1}{14}(A+B+C+D+E+F+G) \end{aligned} \quad (4)$$

$$\begin{aligned} \text{if } K=0.5, X' &= X + \frac{1}{12}(aa+bb+cc+dd+ee+ff) \\ &= \frac{18}{24}X + \frac{2}{24}(a+b+c+d+e+f) - \frac{1}{24}(A+B+C+D+E+F) \\ \text{if } K=1.5, X' &= X + \frac{3}{12}(aa+bb+cc+dd+ee+ff) \\ &= \frac{6}{24}X + \frac{6}{24}(a+b+c+d+e+f) - \frac{3}{24}(A+B+C+D+E+F) \\ \text{if } K=2.0, X' &= X + \frac{4}{12}(aa+bb+cc+dd+ee+ff) \\ &= \frac{0}{24}X + \frac{8}{24}(a+b+c+d+e+f) - \frac{4}{24}(A+B+C+D+E+F) \end{aligned} \quad (5)$$

$$\begin{aligned} signal[v1] &= signal[v1] + \left(\frac{K}{valence[v1]}\right) \times (LoDBase)^{\max LoD - cur LoD} \times signal[v] \\ signal[v2] &= signal[v2] + \left(\frac{K}{valence[v2]}\right) \times (LoDBase)^{\max LoD - cur LoD} \times signal[v] \end{aligned} \quad (7)$$

변위 벡터 성분이 작아지는 것을 확인할 수 있으며, 2 이상의 상수값이 곱해지는 경우 현재 성분이 소멸하는 것을 확인할 수 있다. 결과적으로 정점 *valence* 기반 가중치를 통해 모든 변위 벡터에 대해 동일한 업데이트 필터 특성을 부여하고, 추가적인 특정 상수값을 곱해줌으로써 해당 업데이트 필터의 특성을 조절할 수 있다.

제안하는 정점 *valence* 기반 업데이트 방법에 더해 [15]에서 제안한 *LoD* 기반 업데이트 방법을 같이 적용하여 리프팅 변환 업데이트를 수행할 수 있다. 제안하는 방법과 *LoD* 기반 업데이트 방법을 같이 사용하는 경우 업데이트는 식 (7)과 같이 수행된다.

식 (7)에서 *LoDBase*는 리프팅 변환이 수행됨에 따라 *LoD* 단위로 감소시킬 업데이트 비율을 의미하며, *maxLoD*와 *curLoD*는 각각 최대 *LoD*와 현재 리프팅 변환이 수행되는 *LoD*를 의미한다. 해당 업데이트 방법은 정점 *valence* 기반 가중치를 통해 모든 변위 벡터에 대해 동일한 특성의 필터를 적용하고, *LoD* 기반 가중치를 통해 *LoD* 단위로 리프팅 변환이 수행됨에 따라 멀어지는 정점 간 거리를 반영하여 업데이트를 수행한다.

IV. 실험 결과

본 논문에서 제안하는 방법의 성능을 평가하기 위하여, V-DMC 참조 소프트웨어 TMM v4.0^[16]에 제안하는 방법을 구현하여 기존 TMM v4.0 성능과의 비교를 수행하였다. 제안하는 *valence* 기반 리프팅 변환 업데이트 방법을 구현하기 위해, 기존 TMM v4.0에서 베이스 메쉬의 세분화 과정과 변위 벡터의 리프팅 변환 과정을 수정하였다. 기존 베이스 메쉬의 세분화 과정에서는 베이스 메쉬를 세분화하여 *LoD1*, *LoD2*, *LoD3*와 같은 상위 *LoD*의 정점, 간선 등을 생성하기 위해 각 정점과 연결된 간선 정보를 계산한다. 따라서 계산된 해당 정보를 정점별 *valence* 정보로 사용

하기 위해 저장하는 부분을 추가하였다. 변위 벡터의 리프팅 변환 과정에서는 기존에 사용하던 고정 가중치 0.5 대신 정점별 *valence* 정보로 정규화한 가중치를 사용하여 리프팅 변환 업데이트 과정을 수행하도록 수정하였다. 제안하는 방법의 성능을 평가하기 위한 실험은 V-DMC CTC (Common Test Condition)^[18]의 AI (All Intra)와 LD (Low Delay) 조건에서 진행하였으며, 실험 데이터셋은 V-DMC 앵커 데이터셋^[4]에 포함된 8개의 동적 메쉬 데이터를 사용하였다. V-DMC 앵커 데이터셋은 Category1-A로 분류된 *longdress*, *soldier* 동적 메쉬, Category1-B로 분류된 *basketball*, *dancer* 동적 메쉬, Category1-C로 분류된 *football*, *levi*, *mitch*, *thomas* 동적 메쉬 데이터를 포함하며, 각 동적 메쉬 데이터는 300프레임으로 구성된다. 본 논문에서는 각 동적 메쉬 데이터의 첫 32프레임을 사용했으며, 각 동적 메쉬 데이터에 대해 목표 비트율 R1 ~ R5 환경을 적용하여 실험을 진행하였다. 이때, 목표 비트율 R1은 3~5 Mbps, R2는 5~9Mbps, R3는 10~12Mbps, R4는 14~16 Mbps, R5는 21~25Mbps의 목표 비트율 환경을 의미한다. 실험은 Intel(R) i7-10세대 10700 2.9GHz 프로세서, 64G RAM, Ubuntu 20.04 운영체제 환경에서 진행하였다.

표 1은 TMM v4.0에서 기존 고정된 가중치 0.125를 통해 리프팅 변환 업데이트를 수행하여 동적 메쉬를 압축한 결과 대비 제안하는 정점 *valence* 기반 가중치를 통해 리프팅 변환 업데이트를 수행하여 동적 메쉬를 압축한 결과의 point cloud 기반 BD-Rate 성능을 나타낸다. 이때 정점별 *valence*로 정규화한 가중치만의 성능을 평가하기 위해 별도의 상수값 곱셈 없이 실험을 진행하였다. 성능 비교 평가를 위한 point cloud 기반 BD-Rate 측정법 중 D1 PSNR은 원본 메쉬의 정점과 복원된 메쉬의 정점 간 오차의 평균을 나타내는 정점-대-정점 측정법이며, D2 PSNR은 복원된 메쉬의 정점을 해당하는 원본 메쉬의 평면에 정사영하여 측정된 오류의 평균을 나타내는 정점-대-평면 측정법이다. V-DMC에서 복원된 변위 벡터는 복원된 메쉬의 정점과 원

본 메쉬의 정점 간 유사도를 결정하기 때문에, 본 논문에서는 point cloud 기반 BD-Rate 측정법 D1 PSNR과 D2 PSNR에 초점을 맞춰 성능을 측정하였다. 표 1의 성능 비교 결과를 통해 point cloud 기반 BD-rate D1 PSNR이 AI 환경에서 Category1-A는 평균 -1.4%, Category1-B는 평균 -2.1%, Category1-C는 평균 -3.4%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -2.6%의 성능이 향상됨을 확인할 수 있으며, LD 환경에서 Category1-A는 평균 -1.6%, Category1-B는 평균 -1.9%, Category1-C는 평균 -2.6%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -2.2%의 성능이 향상됨을 확인할 수 있다. 또한 D2 PSNR이 AI 환경에서 Category1-A는 평균 -1.3%, Category1-B는 평균 -2.1%, Category1-C는 평균 -3.4%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -2.6%의 성능이 향상됨을 확인할 수 있으며, LD 환경에서 Category1-A는 평균 -1.6%, Category1-B는 평균 -1.9%, Category1-C는 평균 -2.7%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -2.2%의 성능이 향상됨을 확인할 수 있다. 이는 기존 TMM v4.0에서 고정된 가중치 0.125를 사용해 서로 다른 특성의 필터로 변위 벡터를 업데이트하고 서로 다른 특성으로 보정된 변위 벡터에 대해 이후 예측 과정을 수행하는 것과 대비하여, 제안하는 valence 기반 가중치를 사용해 동일한 특성의 필터로 변위 벡터를 업데이트하고 동일한 특성으로 보정된 변위 벡터에 대해

이후 예측 과정을 수행함으로써, 변위 벡터의 리프팅 변환 간 예측 성능이 향상되어 압축 성능이 향상된 결과라고 볼 수 있다. 정점의 위치, 표면 등과 같은 메쉬 기하 정보의 압축 성능을 나타내는 D1 PSNR과 D2 PSNR이 향상됨과는 반대로, 색상 등과 같은 메쉬 속성 정보의 압축 성능을 나타내는 Chroma Cb PSNR과 Chroma Cr PSNR은 전체적으로 하락한 것을 확인할 수 있다. 메쉬의 색상 정보를 포함하는 텍스처맵은 제안하는 방법으로 복원된 메쉬의 기하 정보에 맞게 재생성된 후, 비디오 코덱을 통해 손실 압축 방식으로 압축된다. 이때 텍스처맵은 3D 메쉬의 PSNR이 아닌 2D 텍스처맵 이미지의 PSNR을 목적함수로 손실 압축되며, 특히 텍스처맵의 chroma 성분의 경우 비디오 코덱의 입력으로 들어가기 전 4:2:0 포맷으로 다운샘플링 되기 때문에, 결과적으로 일부 메쉬 데이터에서 메쉬 기하 정보의 PSNR 향상이 색상 정보의 PSNR 향상으로 이어지지 못하고, Chroma PSNR이 소량 감소하는 것으로 판단된다. 추가적으로 표 1의 성능 비교 결과를 통해 AI 환경에서 부호화 시간이 100%, 복호화 시간이 101%이고, LD 환경에서 부호화 시간이 100%, 복호화 시간이 100%로 별도의 부·복호화 시간이 증가하지 않은 것을 확인할 수 있다. 이는 기존 부·복호화 간 수행되는 베이스 메쉬의 세분화 과정에서 계산된 각 정점과 연결된 간선 정보를 저장하여 정점별 valence 정보로 사용함으로써, 별도의 valence 계산과정이 추가되지 않은 결과라고 볼 수 있다.

표 1. TMM v4.0 대비 정점 valence 기반 리프팅 변환 업데이트 방법의 BD-Rate 성능
Table 1. BD-Rate performance of vertex valence based lifting update method compared to TMM v4.0

Condition	Sequence	Pointcloud-based BD Rate [%]				
		D1	D2	Luma	Chroma Cb	Chroma Cr
AI	Cat1-A average	-1.4%	-1.3%	-0.2%	0.1%	0.2%
	Cat1-B average	-2.1%	-2.1%	-0.3%	0.2%	0.1%
	Cat1-C average	-3.4%	-3.4%	-0.6%	0.6%	0.2%
	Overall average	-2.6%	-2.6%	-0.4%	0.4%	0.2%
	Avg. Enc Time [%]	100%				
	Avg. Dec Time [%]	101%				
LD	Cat1-A average	-1.6%	-1.6%	-0.3%	-0.5%	3.5%
	Cat1-B average	-1.9%	-1.9%	-0.2%	0.1%	0.0%
	Cat1-C average	-2.6%	-2.7%	-0.5%	0.4%	-0.1%
	Overall average	-2.2%	-2.2%	-0.4%	0.1%	0.8%
	Avg. Enc Time [%]	100%				
	Avg. Dec Time [%]	100%				

표 2. LoD 기반 리프팅 변환 업데이트 방법 대비 정점 valence 기반 방법과 LoD 기반 방법을 모두 사용한 리프팅 변환 업데이트 방법의 BD-Rate 성능

Table 2. BD-Rate performance of lifting update method with vertex valence based and LoD based weight compared to lifting update method with LoD based weight

Condition	Sequence	Pointcloud-based BD Rate [%]				
		D1	D2	Luma	Chroma Cb	Chroma Cr
AI	Cat1-A average	0.1%	0.1%	0.1%	-0.1%	-0.2%
	Cat1-B average	0.0%	-0.1%	0.1%	0.1%	-0.3%
	Cat1-C average	-1.6%	-1.7%	-0.3%	-0.2%	-0.2%
	Overall average	-0.8%	-0.8%	-0.1%	-0.1%	-0.2%
	Avg. Enc Time [%]	99%				
	Avg. Dec Time [%]	100%				
LD	Cat1-A average	0.0%	-0.1%	0.0%	0.2%	0.3%
	Cat1-B average	0.0%	-0.1%	0.1%	0.0%	-0.2%
	Cat1-C average	-1.2%	-1.3%	0.0%	0.0%	-0.2%
	Overall average	-0.6%	-0.7%	0.0%	0.1%	-0.1%
	Avg. Enc Time [%]	100%				
	Avg. Dec Time [%]	99%				

표 2는 TMM v4.0에서 LoD 기반 가중치^[15]를 사용해 리프팅 변환 업데이트를 수행하여 동적 메쉬를 압축한 결과 대비 LoD 기반 가중치와 제안하는 정점 valence 기반 가중치를 같이 사용해 리프팅 변환 업데이트를 수행하여 동적 메쉬를 압축한 결과의 point cloud 기반 BD-Rate 성능을 나타낸다. 이때 LoD 기반 가중치에서 LoD_{Base} 는 0.75로 설정하였으며, 정점별 valence로 정규화한 가중치에는 상수값 1.4를 곱하여 실험을 진행하였다. 해당 파라미터 값은 실험적으로 구한 최적의 파라미터 값으로, 제안한 방법의 최고 성능을 비교 평가하기 위해 사용하였다. 표 2의 성능 비교 결과를 통해 point cloud 기반 BD-rate D1 PSNR이 AI 환경에서 Category1-C는 평균 -1.6%로 향상되었으며 전체 평균 -0.8%의 성능이 향상됨을 확인할 수 있으며, LD 환경에서 Category1-C는 평균 -1.2%로 향상되었으며 전체 평균 -0.6%의 성능이 향상됨을 확인할 수 있다. 또한 D2 PSNR이 AI 환경에서 Category1-C는 평균 -1.7%로 향상되었으며 전체 평균 -0.8%의 성능이 향상됨을 확인할 수 있으며, LD 환경에서 Category1-C는 평균 -1.3%로 향상되었으며 전체 평균 -0.7%의 성능이 향상됨을 확인할 수 있다. 표 2의 실험은 LoD 기반 업데이트 방법에 제안하는 valence 기반 업데이트 방법이 추가로 적용되었을 때의 성능을 평가하기 위해 진행하였다. 결과적으로 LoD 기반 가중치를 통해 서로 다른 LoD 간 특성이 고려된 업데이트

필터에, valence 기반 가중치로 한 LoD 내에서 동일한 필터 특성을 적용함으로써 압축 성능이 향상됨을 확인하였다. 표 1의 실험 결과와 마찬가지로 기존 베이스 메쉬의 세분화 과정에서 계산된 각 정점과 연결된 간선 정보를 정점별 valence 정보를 사용하기 때문에, 표 2의 성능 비교 결과에서 AI 환경에서의 부호화 시간이 100%, 복호화 시간이 99%이고, LD 환경에서의 부호화 시간이 100%, 복호화 시간이 99%로 별도의 부·복호화 시간이 증가하지 않은 것을 확인할 수 있다.

표 3은 TMM v4.0에서 기존 고정된 가중치 0.125를 사용해 리프팅 변환 업데이트를 수행하여 동적 메쉬를 압축한 결과 대비 LoD 기반 가중치와 제안하는 정점 valence 기반 가중치를 같이 사용해 리프팅 변환 업데이트를 수행하여 동적 메쉬를 압축한 결과의 point cloud 기반 BD-Rate 성능을 나타낸다. 표 2의 실험과 동일하게 LoD 기반 가중치에서 LoD_{Base} 는 0.75로 설정하고, 정점별 valence로 정규화한 가중치에는 상수값 1.4를 곱하여 실험을 진행하였다. 표 3의 성능 비교 결과를 통해 point cloud 기반 BD-rate D1 PSNR이 AI 환경에서 Category1-A는 평균 -2.9%, Category1-B는 평균 -4.9%, Category1-C는 평균 -3.8%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -3.9%의 성능이 향상됨을 확인할 수 있으며, LD 환경에서 Category1-A는 평균 -3.1%, Category1-B는 평균 -4.8%,

표 3. TMM v4.0 대비 정점 valence 기반 방법과 LoD 기반 방법을 모두 사용한 리프팅 변환 업데이트 방법의 BD-Rate 성능
 Table 3. BD-Rate performance of lifting update method with vertex valence based and LoD based weight compared to TMM v4.0

Condition	Class	Sequence	Pointcloud-based BD Rate [%]				
			D1	D2	Luma	Chroma Cb	Chroma Cr
AI	Cat1-A	longdress	-2.7%	-2.5%	-0.3%	0.1%	0.0%
		soldier	-3.1%	-3.0%	-0.4%	-0.1%	0.9%
	Cat1-B	basketball	-4.7%	-4.5%	-0.4%	-0.1%	0.9%
		dancer	-5.1%	-4.9%	-1.0%	0.4%	0.9%
	Cat1-C	football	-1.6%	-2.4%	-0.4%	0.5%	0.1%
		levi	-2.5%	-2.5%	-1.1%	0.2%	0.2%
		mitch	-6.4%	-6.0%	-0.8%	0.0%	-0.4%
		thomas	-4.8%	-4.3%	-1.0%	1.8%	0.5%
	Cat1-A average		-2.9%	-2.7%	-0.3%	0.0%	0.4%
	Cat1-B average		-4.9%	-4.7%	-0.7%	0.2%	0.9%
	Cat1-C average		-3.8%	-3.8%	-0.9%	0.6%	0.1%
	Overall average		-3.9%	-3.8%	-0.7%	0.4%	0.4%
	Avg. Enc Time [%]		100%				
Avg. Dec Time [%]		99%					
LD	Cat1-A	longdress	-4.6%	-4.3%	-0.4%	0.0%	-0.1%
		soldier	-1.5%	-1.5%	-0.8%	-1.4%	6.3%
	Cat1-B	basketball	-4.7%	-4.5%	-0.4%	-0.1%	0.9%
		dancer	-5.0%	-4.7%	-0.8%	-0.2%	0.3%
	Cat1-C	football	-1.3%	-2.0%	-0.5%	0.6%	0.2%
		levi	-2.6%	-2.4%	-1.4%	-0.1%	-0.6%
		mitch	-3.9%	-3.9%	-0.2%	-0.2%	0.1%
		thomas	-4.9%	-4.7%	-0.2%	3.9%	-0.9%
	Cat1-A average		-3.1%	-2.9%	-0.6%	-0.7%	3.1%
	Cat1-B average		-4.8%	-4.6%	-0.6%	-0.1%	0.6%
	Cat1-C average		-3.2%	-3.2%	-0.6%	1.0%	-0.3%
	Overall average		-3.6%	-3.5%	-0.6%	0.3%	0.8%
	Avg. Enc Time [%]		100%				
Avg. Dec Time [%]		100%					

Category1-C는 평균 -3.2%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -3.6%의 성능이 향상됨을 확인할 수 있다. 또한 D2 PSNR이 AI 환경에서 Category1-A는 평균 -2.7%, Category1-B는 평균 -4.7%, Category1-C는 평균 -3.8%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -3.8%의 성능이 향상됨을 확인할 수 있으며, LD 환경에서 Category1-A는 평균 -2.9%, Category1-B는 평균 -4.6%, Category1-C는 평균 -3.2%로 모든 동적 메쉬 데이터셋에 대해서 향상되어 전체 평균 -3.5%의 성능이 향상됨을 확인할 수 있다. 이는 LoD 기반 가중치로 서로 다른 LoD 간의 거리 특성을 업데이트 필터에 반영하고, 정점별 valence 기반 가중치로 한 LoD 내에서 동일한 필터 특성을 갖게 해줌으로써, 기존 고정 가중치만을 사용하여 업데이트 필터를 적용하였을 때 대비 변위 벡터의 리

프팅 변환 압축 성능이 향상된 결과라고 볼 수 있다. 추가적으로, LoD 기반 가중치는 LoD 당 한 번에 실수 연산으로 계산되며, valence 기반 가중치는 기존 베이스 메쉬의 세분화 과정에서 사용되던 정보로 계산되기 때문에 별도의 부·복호화 시간이 증가하지 않고 압축 성능이 향상된 것을 확인할 수 있다.

V. 결론

본 논문에서는 MPEG에서 표준화가 진행 중인 동적 메쉬 압축 표준 V-DMC 기술에서 각 정점의 valence에 기반하여 변위 벡터의 리프팅 웨이블릿 변환 업데이트 과정을 수행하는 방법을 제안하였다. 제안하는 방법에서는 메쉬의

각 정점에서 valence를 계산하고, 계산된 valence 값으로 정규화한 가중치를 업데이트 가중치로 사용함으로써, 각 정점의 변위 벡터에 대해서 동일한 특성을 가진 필터로 리프팅 변환 업데이트를 수행한다. 제안하는 방법은 별도의 부·복호화 시간 증가 없이 point cloud 기반의 BD-rate (D1 PSNR)가 기존 TMM v4.0 대비 AI, LD 환경에서 각각 평균 -2.6%, -2.2% 향상된 성능을 보였으며, 이를 통해 메쉬 변위 벡터의 압축 효율이 향상됨을 확인하였다. 또한 제안하는 방법과 기존 선행 연구에서 제안된 LoD 기반 업데이트 방법^[15]를 같이 고려하여 리프팅 웨이블릿 변환 업데이트를 수행한 결과, point cloud 기반의 BD-rate (D1 PSNR)가 기존 TMM v4.0 대비 AI, LD 환경에서 각각 평균 -3.9%, -3.6% 향상되는 것을 확인하였다.

참 고 문 헌 (References)

- [1] Y. Choi, J. Jeong, S. Lee, and E. Ryu, "Overview of the Video-based Dynamic Mesh Coding (V-DMC) Standard Work," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2022, p. 578-581. doi: <https://doi.org/10.1109/ICTC55196.2022.9952734>
- [2] Information technology – Coding of audio-visual objects – Part 16: Animation Framework eXtension (AFX), ISO/IEC 14496-16, 2004. <https://www.iso.org/standard/38569.html>
- [3] Information technology – Coding of audio-visual objects – Part 16: Animation Framework eXtension (AFX) – Amendment 2: Frame-based Animated Mesh Compression (FAMC), ISO/IEC 14496-16:2006/Amd 2, 2009. <https://www.iso.org/standard/50471.html>
- [4] CfP for Dynamic Mesh Coding, ISO/IEC JTC1/SC29/WG7, N145, Oct. 2021.
- [5] WD 4.0 of V-DMC, ISO/IEC JTC1/SC29/WG7, N00680, Geneva, Jul. 2023.
- [6] K. Mammou, J. Kim, A. Tourapis, D. Podborski, K. Kolarov, [V-CG] Apple's Dynamic Mesh Coding CfP Response, ISO/IEC JTC1/SC29/WG7, m59281, Apr. 2022.
- [7] Alliez, P, "Recent advances in compression of 3D meshes," 2005 13th European Signal Processing Conference, IEEE. 2005, pp.1-4. doi: https://doi.org/10.1007/3-540-26808-1_1
- [8] UVAtlas, <https://github.com/microsoft/UVAtlas> (accessed Oct, 02, 2023) <https://github.com/microsoft/UVAtlas>
- [9] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes," IEEE Trans. Vis. Comput. Graphics, vol.5, no.1, pp.47-61, 1999. doi: <https://doi.org/10.1109/2945.764870>
- [10] W. Sweldens, "The lifting scheme: A construction of second generation wavelets." SIAM journal on mathematical analysis, vol.29, no.2 pp511-546, 1998. doi: <https://doi.org/10.1137/S0036141095289051>
- [11] G. Sullivan, J. Ohm, W. Han, T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Transactions on circuits and systems for video technology, Vol.22, No.12, pp.1649-1668, December 2012. doi: <https://doi.org/10.1109/TCSVT.2012.2221191>
- [12] M. Lee, H. Song, J. Park, B. Jeon, J. Kang, J. Kim, Y. Lee, J. Kang, D. Sim, "Overview of Versatile Video Coding (H. 266/VVC) and Its Coding Performance Analysis," IEIE Transactions on Smart Processing & Computing, Vol. 12, No.2, pp.122-154, April 2023. doi: <https://doi.org/10.5573/IEIESPC.2023.12.2.122>
- [13] M. Kraus, "The pull-push algorithm revisited," Proceedings GRAPP 2: 3, 2009. doi: <https://doi.org/10.5220/0001772601790184>
- [14] J. Lee, "Introduction of Video-based Dynamic Mesh Coding Standardization Trend," Journal of Broadcast Engineering Vol.28, No.3, pp.23-32, July 2023.
- [15] Chao Cao, [V-DMC][NEW] LOD-based adaptive update weight for Forward Linear Lifting Wavelet Transform, ISO/IEC JTC1/SC29/WG7, m64223, Jul, 2023.
- [16] V-DMC TMM4.0, ISO/IEC JTC1/SC29/WG7, N00595, Apr, 2023.
- [17] Szczesna, A, "Designing lifting scheme for second generation wavelet-based multiresolution processing of irregular surface meshes," Proceedings of Computer Graphics and Visualization, 103-111, 2008.
- [18] Common Test Condition for V3C and V-DMC, ISO/IEC JTC1/SC29/WG7, N00555, Mar, 2023.

저 자 소 개



김민태

- 2023년 2월 : 광운대학교 컴퓨터공학과 학사
- 2023년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 석사과정
- ORCID : <https://orcid.org/0009-0007-9100-8567>
- 주관심분야 : 3D데이터압축, 영상압축, 컴퓨터비전



변주형

- 2019년 2월 : 광운대학교 컴퓨터공학과 학사
- 2021년 2월 : 광운대학교 컴퓨터공학과 석사
- 2021년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 박사과정
- ORCID : <https://orcid.org/0000-0002-6165-9189>
- 주관심분야 : 3D데이터압축, 영상압축, 컴퓨터비전



심동규

- 1993년 2월 : 서강대학교 전자공학과 공학사
- 1995년 2월 : 서강대학교 전자공학과 공학석사
- 1999년 2월 : 서강대학교 전자공학과 공학박사
- 1999년 3월 ~ 2000년 8월 : 현대전자 선임연구원
- 2000년 9월 ~ 2002년 3월 : 바로비전 선임연구원
- 2002년 4월 ~ 2005년 2월 : University of Washington Senior research engineer
- 2005년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 교수
- ORCID : <https://orcid.org/0000-0002-2794-9932>
- 주관심분야 : 영상신호처리, 영상압축, 컴퓨터비전