

특집논문 (Special Paper)

방송공학회논문지 제26권 제3호, 2021년 5월 (JBE Vol. 26, No. 3, May 2021)

<https://doi.org/10.5909/JBE.2021.26.3.258>

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

포인트 클라우드 콘텐츠 해상도 향상을 위한 점진적 렌더링 방법

이 희 제^{a)}, 윤 준 영^{a)}, 김 종 욱^{a)}, 김 찬 희^{a)}, 박 종 일^{a)*}

A Progressive Rendering Method to Enhance the Resolution of Point Cloud Contents

Heejea Lee^{a)}, Junyoung Yun^{a)}, Jongwook Kim^{a)}, Chanhee Kim^{a)}, and Jong-Il Park^{a)*}

요 약

포인트 클라우드 콘텐츠는 3차원 포인트로 실제 객체를 나타내는 몰입형 콘텐츠이다. 포인트 클라우드 데이터를 획득하거나 포인트 클라우드 데이터를 인코딩 및 디코딩하는 과정에서 포인트 클라우드 콘텐츠의 해상도가 저하될 수 있다. 본 논문에서는 프레임 간 정합을 통해 순차적으로 포인트 클라우드 콘텐츠의 해상도를 점진적으로 향상시키는 방법을 제안한다. 포인트 클라우드 데이터를 정합하기 위해 ICP(Iterative Closest Point) 알고리즘이 일반적으로 사용된다. 기존 ICP 알고리즘은 강체를 변환할 수 있지만 포인트 클라우드 콘텐츠와 같이 로컬에서 서로 다른 방향으로 모션 벡터를 갖는 비 강체에 대해서는 변환이 불가능하다는 단점이 있다. 현재 프레임의 포인트 클라우드와 이전 프레임 사이의 포인트를 쌍을 만들고 만들어진 쌍의 움직임량을 계산하여 보상해주는 방법으로 기존 ICP 정합에서의 한계를 극복하였다. 이러한 방식으로 프레임 사이에 포인트를 정합하는 과정을 통해 기하학적 움직임이 있는 포인트 클라우드 콘텐츠의 해상도가 향상됨을 보였다.

Abstract

Point cloud content is immersive content that represents real-world objects with three-dimensional (3D) points. In the process of acquiring point cloud data or encoding and decoding point cloud data, the resolution of point cloud content could be degraded. In this paper, we propose a method of progressively enhancing the resolution of sequential point cloud contents through inter-frame registration. To register a point cloud, the iterative closest point (ICP) algorithm is commonly used. Existing ICP algorithms can transform rigid bodies, but there is a disadvantage that transformation is not possible for non-rigid bodies having motion vectors in different directions locally, such as point cloud content. We overcome the limitations of the existing ICP-based method by registering regions with motion vectors in different directions locally between the point cloud content of the current frame and the previous frame. In this manner, the resolution of the point cloud content with geometric movement is enhanced through the process of registering points between frames. We provide four different point cloud content that has been enhanced with our method in the experiment.

Keyword : point cloud, rendering, deformable registration, progressive

1. 서론

최근 4차 산업 산업의 일환으로 5G 네트워크를 통한 대용량 데이터 수집 및 전송 기술이 크게 발전하고 있다. 이러한 기술의 발달로 인해 3차원 데이터를 획득하여 실시간으로 전송하는 비디오 미디어 콘텐츠가 다양한 산업에서 사용되고 있다^[1].

3차원 몰입형 미디어 콘텐츠는 2차원 콘텐츠에 비해 사용자에게 보다 현실적인 몰입감 뿐만 아니라 인터랙티브적인 요소까지 증강시킨다. 사실적인 미디어 콘텐츠를 표현하는 대표적인 방법으로는 실제 물체를 3D 포인트로 표현하는 포인트 클라우드 콘텐츠와 폴리곤 메쉬를 표현하는 메쉬 콘텐츠가 있다. 이 두 가지 방법에는 장점과 단점이 있다. 우선, 메쉬 콘텐츠는 데이터를 압축할 경우 포인트 클라우드 콘텐츠에 비해 형상을 더 잘 보존하는 것처럼 보이지만 실제로는 텍스처의 왜곡이 심한 단점이 있다^[2]. 또한 메쉬 콘텐츠는 포인트 클라우드 콘텐츠보다 렌더링의 세밀함이 떨어진다. 이는 **bitrate**가 낮을 때 특히 두드러진다. 포인트 클라우드 콘텐츠의 경우 영상 확대, 획득, 복원 또는 움직임이 빠르면 개별 포인트로 구성된 포인트 클라우드의 구멍이 발생하는 등 콘텐츠의 품질이 저하될 수 있다. 이러한 품질 저하의 문제(특히 구멍이 생기는)에서 품질을 향상한다면 메쉬 콘텐츠보다 더 세밀하게 렌더링 할 수 있게 된다^[3].

낮은 품질의 포인트 클라우드 콘텐츠는 콘텐츠를 확대하였을 경우 각 포인트로 구성된 오브젝트의 사이사이에 포인트의 밀도가 낮아져 포인트로 채워지지 못하고 구멍이

생기는 것을 의미한다. 즉, 적절한 방법을 사용하여 포인트 밀도를 높이면 포인트 클라우드 콘텐츠의 품질이 향상된다. 이에 본 논문에서는 포인트 클라우드 콘텐츠의 해상도를 향상시키는 방법을 제안한다^{[4][5]}. 본 논문에서 제안하는 방법은 포인트 클라우드 데이터를 사용하여 비슷한 두 프레임을 합쳐 밀도를 향상시켜 해상도를 높게 만드는 것이다. 그러나 움직임이 있는 포인트 클라우드 데이터를 합칠 경우 움직임에 대해 정합이 필요하다. 그렇지 않으면 포인트 수가 증가하지만 해상도가 증가할 것이라는 예상과는 완전히 다른 렌더링이 될 것이다. 따라서 움직이는 프레임에 대한 정합이 필요하며^{[6][7]}, 정합 방법 중 전 연구^{[8][9]}에서 사용된 ICP 알고리즘이 있다. 그러나 ICP 알고리즘은 단일 물체에 대해 움직임 벡터가 다를 경우 적용하기 어려운 문제가 있다. 본 논문에서는 포인트 클라우드 이미지 데이터 프레임 t 와 $t + 1$ 의 움직임 차이 값을 각 포인트에 대한 움직임으로 보상하여 프레임 t 와 $t + 1$ 의 객체를 정합한다. 두 개의 정합된 프레임을 결합하면 포인트 수가 증가하고 밀도가 높아져 콘텐츠의 품질이 향상된다. 본 연구에서는 포인트 클라우드 콘텐츠의 품질 및 해상도를 향상시키기 위해 다중의 프레임을 정합시키는 방법으로 접근하였다. 이는 ICP 알고리즘으로는 정합하기 어려운 국소적 움직임이 다른 물체에 대해서도 각 포인트 별 움직임을 추적하여 움직임을 보상해준 후 정합하여 포인트 클라우드 콘텐츠의 품질을 효과적으로 높이는 방법을 제안한다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 ICP 알고리즘 및 포인트 클라우드 해상도를 개선하는 다양한 방법에 대해 소개한다^[10]. 3절과 4절에서는 본 논문에서 제시하는 포인트 클라우드 해상도를 향상시키는 방법에 대해 설명한다. 마지막으로 5절에서는 실험 결과와 논문의 결론에 대해 설명한다.

II. 관련 연구

1. ICP 알고리즘

3D 포인트 클라우드를 정합하기 위해 Besl이 제안한 ICP 알고리즘이^[5] 일반적으로 사용된다. ICP 알고리즘은 소스

a) 한양대학교 컴퓨터소프트웨어학과(Department of Computer Science, Hanyang University)

‡ Corresponding Author : 박종일(Jong-Il Park)

E-mail: jipark@hanyang.ac.kr

Tel: +82-2-2220-4368

ORCID: <https://orcid.org/0000-0003-1000-4067>

※ 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2020-0-00452, 적응형 뷰어 중심 포인트 클라우드AR/VR 스트리밍 플랫폼 기술 개발)

※ This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP), Korea Government (MSIT) (Development of Adaptive Viewer centric Point cloud AR/VR StreamingPlatform) under Grant 2020-0-00452.

※ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1F1A1041882).

· Manuscript received April 9, 2021; Revised May 24, 2021; Accepted May 24, 2021.

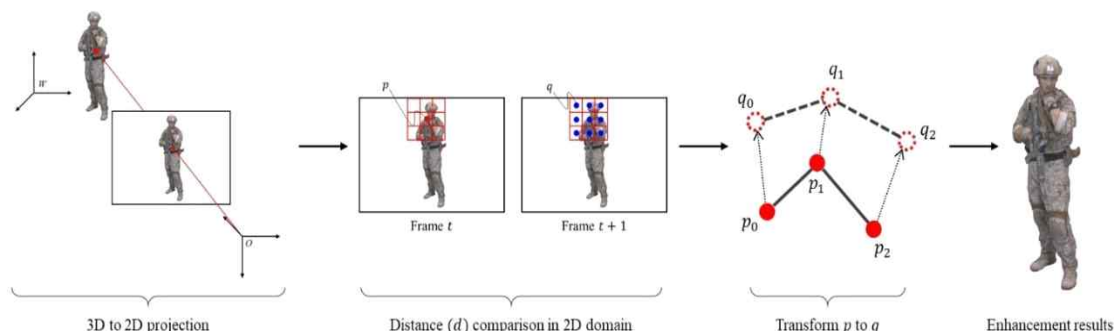


그림 1. 포인트 클라우드 해상도 향상을 위한 프로세스

Fig. 1. The process of enhancing the resolution of point cloud contents

포인트에서 인접한 타겟 포인트 쌍의 거리의 차이를 최소화하는 변환을 반복적으로 수행하여 최적의 변환 행렬을 유도하는 알고리즘이다. 특징점 쌍을 계산할 필요가 없기 때문에 변환 행렬을 쉽게 유도할 수 있다. 게다가 모양이 고정되고 변하지 않는 물체에 대한 변환인 강체 변환의 경우 강력하게 동작하는 장점이 있다^[12]. 그러나 순차적으로 변하는 포인트 클라우드 콘텐츠와 같이 국소적으로 서로 다른 방향의 모션 벡터를 갖는 비 강체의 경우 변환이 어렵다는 한계가 있다. 이러한 한계를 극복하기 위해 본 논문에서는 국소적으로 다른 방향의 모션 벡터를 갖은 부분도 정합이 가능한 방법을 제안한다.

2. 포인트 클라우드 해상도 증가와 관련된 이전 방법 및 문제점

ICP정합 알고리즘을 사용한 이전 연구^[33]에서는 ICP 알고리즘과 다중 프레임을 사용하여 소스 포인트 클라우드와 타겟 포인트 클라우드를 정합한다. 첫 번째 정합 후 소스 포인트와 타겟 포인트 사이의 거리를 비교하여 정합이 되었다고 판단된 포인트와 정합 되지 않았다고 판단된 포인트를 분리한다. 그 후 일치하지 않는 포인트만 ICP 알고리즘을 사용하여 재 정합 한다. 그러나 위의 실험에서 발생하는 두 가지 문제가 있다.

첫 번째 문제는 소스 포인트와 타겟 포인트 사이의 거리를 비교할 때 시간 복잡도가 $O(n^2)$ ^[19]라는 것이다. 데이터의 양이 많은 포인트 클라우드 데이터의 특성상 상당한 연산량을 요구한다.

두 번째 문제는 아래 그림 2에서 볼 수 있듯 물체의 특성이 변형 가능한 경우이다. 강체의 경우 ICP 알고리즘을 한 번만 사용하여 매칭이 가능하지만 비 강체의 경우 완벽한 매칭이 불가능하다. ICP 알고리즘을 사용하여 여러 번 정합을 시도했지만 완벽한 정합은 불가능했다. 위의 두 가지 문제를 해결하기 위해 본 논문에서 방법을 제시한다.

포인트 클라우드 데이터의 품질을 높이는 방법 중 하나는 포인트의 크기를 늘리는 것이다. 포인트의 크기를 키울 때 일반적인 방법으로 화면상에 투사되는 모든 포인트를 일정하게 키우게 되면 3D 포인트가 갖는 원근감이 줄어들게 된다. 그러나 Splatting 방식을 사용하면 깊이를 고려하여 포인트 크기가 커지므로 원근감이 보장된다. Splatting 방법은 3D 좌표에서 특정 크기의 구로 점을 만든다. 생성된 구를 화면에 투사하면 깊이에 따라 포인트 크기의 크기가 변경된다^[20].



그림 2. ICP 알고리즘을 사용한 포인트 클라우드 시퀀스 정합 결과

Fig. 2. The results of performing registration with the ICP algorithm for the point cloud data sequences

그러나 위의 방법은 단일 프레임만을 사용하여 포인트의 해상도를 향상시킨다. 포인트 클라우드 데이터가 예기치 않게 손실되면 복원이 불가능 해진다. 그러나 본 논문에서 제안한 방법을 사용하면 데이터 손실에 대해서도 강건하게 동작 한다. 또한 채워주는 지점의 색상 값은 추정하여 채워지게 된다. 그러나 본 논문에서는 포인트가 증가할 때를 추정하여 색상 값을 결정하지 않는다. 포인트가 증가하면 이전 프레임의 포인트 색상 값 정보를 사용하므로 해상도가 높아진다.

3. 처리되지 않은 포인트 클라우드의 점진적 실시간 렌더링 방법

Schuetz가 제안한 방법은 GPU 메모리가 실시간으로 보유할 수 있는 만큼의 포인트를 점진적으로 표시하는 데 사용된다. 이 방법은 보이는 점을 재 투영하고 무작위로 추가 점을 추가하여 몇 프레임 내에서 전체 결과에 균일하게 수렴하는 방식으로 수행된다. 저자는 클라우드가 로드되는 동안 LAS 또는 LAZ와 같은 일반적인 순차적 형식으로 포인트 클라우드를 직접 보는 데 사용할 수 있는 방법을 제안한다. 저자가 제안한 방법은 각 프레임을 렌더링 할 때 포인트 수를 제한하고 무작위로 섞인 배열을 데이터구조로 사용한다. 또한 공간 가속도를 미리 만들 필요가 없다. 그러나 이 방법을 사용하려면 데이터구조가 GPU 메모리에 적합해야 한다. 이 방법의 장점은 저가 GPU를 사용하는 노트북에도 적합하다는 것이다. 이 방법의 첫 번째 단계에서는 이전 프레임을 다시 투영한다. 이 방법에 따르면 이전 프레임에서 보였던 포인트의 인덱스만 인덱스 버퍼에 포함될 수 있으며 이 인덱스 버퍼로 VBO (Vertex buffer Object)가 렌더링된다. 이전에 차단된 영역이 표시되고 이전 프레임에서 인접한 지점이 현재 프레임에서 멀리 떨어져 있을 때 간격이 나타난다^[18].

위 연구에서는 최종 결과까지 일정한 진행을 위해 각 프레임에서 다른 랜덤한 포인트 셋의 변환으로 인해 생성된 간격을 채우기로 한다. 정렬된 포인트가 벤치 마크(benchmarks) 성능에서 더 빠르기 때문에 무작위 포인트가 선택되지만 결과적으로 눈에 띄고 불쾌한 깜박임이 나타난다. 다음 프레임을 준비하기 위해 컴퓨트 셰이더(Compute Shader)는 모든 보이는 지점에서 첫 번째 패스에 사용되는

동적 인덱스 버퍼를 만든다. 컴퓨트 셰이더는 포인트 인덱스 컬러 attachment의 각 픽셀을 실행하고 GPU의 버퍼에 인덱스를 직접 저장한다. 따라서 CPU와의 데이터 교환 및 동기화를 피할 수 있다.

위의 방법은 무작위로 점을 선택하여 렌더링 하므로 속도는 빠르지만 품질 향상에는 거의 영향을 미치지 않는다. 본 논문에서는 무작위 포인트가 아닌 모든 포인트를 사용하여 매칭을 시도함으로써 품질 향상 효과를 기대한다.

4. 적대적 잔차 네트워크를 사용한 포인트 클라우드 초 고해상도 복원 방법

그래프 네트워크 및 적대적 손실을 기반으로 하는 포인트 클라우드 초 고해상도를 위한 AR-GCN (Adversarial Residual Graph Convolution Network)이라는 데이터 기반 방법을 제안한다. 네트워크의 가장 주목할 만한 점은 포인트 클라우드의 로컬 유사성과 저해상도 (LR) 입력과 고해상도 (HR) 출력 간의 유사성을 사용한다는 것이다. 새로운 손실 함수로 네트워크를 훈련시키고 이 함수는 Chamfer Distance (CD)와 그래프 적대 손실을 함께 결합한다^[21]. 포인트 클라우드 초 고해상도의 성능을 개선하기 위해 PU-Net의 결합을 해결한다. GCN의 풀링 해제 계층은 x_{in} 과 x_{out} 사이의 유사성을 사용하여 둘 사이의 간차를 예측한다. 결과적으로 x_{out} 을 직접 회귀하여 발생하는 훈련 난이도의 결합을 보상한다. PU-Net의 손실 함수를 단순화하기 위해 LS-GAN을 기반으로 새로운 손실 함수, 그래프 적대 손실을 설계한다. 새로운 손실 함수는 네트워크에 의해 정의될 수 있으며 더 이상 수동으로 설계하지 않고도 데이터에서 자동으로 학습할 수 있다^{[16][17]}.

새로운 손실 기능은 HR 포인트 클라우드의 특성을 자동으로 캡처하고 실제 포인트 클라우드와 가짜 포인트 클라우드를 구별하는데도 사용된다. 그러나 제안된 방법의 단점은 아직 완성되지 않아 큰 구멍 등 빈 부분을 채울 수 없다는 점이다. 게다가 위의 방법은 훈련에 시간이 걸리기 때문에 실시간으로 운영 할 수 없다는 단점이 있습니다. 이러한 단점들을 극복하기 위해 본 논문의 방법을 사용하여 실시간으로 활용할 수 있으며 큰 구멍 등 품질의 저하를 극복할 수 있는 방법을 제안한다.

III. 점진적 렌더링 방법

본 논문에서 제안하는 점진적 렌더링 방법은 다음의 세 단계로 수행된다.

step 1. Pairing 3D Points

첫 번째 단계에서는 시간 복잡도를 줄이기 위해서 포인트 클라우드 데이터를 뷰어 상에 비추지는 3D 포인트에 대해서만 **projection**을 진행한다. 소스 포인트와 타겟 포인트 데이터 모두 2D **projection**을 진행한다. 그림 3에서는 3D 포인트를 2D로 **projection**을 진행한다. 또한 아래 식 (1) 과 (2)를 통해 2차원으로 **projection** 한다.

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \quad (1)$$

$$p = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \tilde{u}/\tilde{w} \\ \tilde{v}/\tilde{w} \end{pmatrix} \quad (2)$$

그 결과 src_{map} 과 dst_{map} 을 생성한다. 그리고 두 map 모두 **projection** 된 포인트들은 원본의 3D 좌표의 정보를 저장하고 있다. 3x3 window를 사용하여 src_{map} 의 인덱스와

dst_{map} 의 인덱스를 비교한다. 현재 3x3 window이므로 src_{map} 기준 dst_{map} 의 9곳의 인덱스를 확인하며 저장하고 있던 3D 좌표 상의 거리를 비교하여 거리가 가장 가까운 포인트들을 **pair**로 생각한다. 다음 작업으로는 **pair**가 된 포인트들을 RGB 색공간에서 HSV 색공간으로 변환한다^[26]. HSV 색공간으로 변환된 **pair** 점들의 **color distance**를 계산하여 일정 **threshold**를 넘어서는 점들에 대해서는 잘못 **pair** 되었다고 판단하여 **pair**를 폐기하게 된다. 아래 식 (3), (4), (5)는 RGB색공간을 HSV로 바꾼 후 **color distance**를 계산하는 식을 나타낸다.

$$(R_{src}, G_{src}, B_{src}) \rightarrow (H_{src}, S_{src}, V_{src}) \quad (3)$$

$$(R_{src}, G_{src}, B_{src}) \rightarrow (H_{src}, S_{src}, V_{src}) \quad (4)$$

$$d = \sqrt{(H_{src} - H_{dst})^2 + (S_{src} - S_{dst})^2 + (V_{src} - V_{dst})^2} \quad (5)$$

step 2. Compensation Amount of Movement

두 번째 단계에서는 첫번째 단계에서 **pair**로 만들어준 3D 포인트에 대해 이동량을 계산한 다음 이동량에 대해 보상해주게 된다. Euclidean distance를 사용하여 이동량을 구

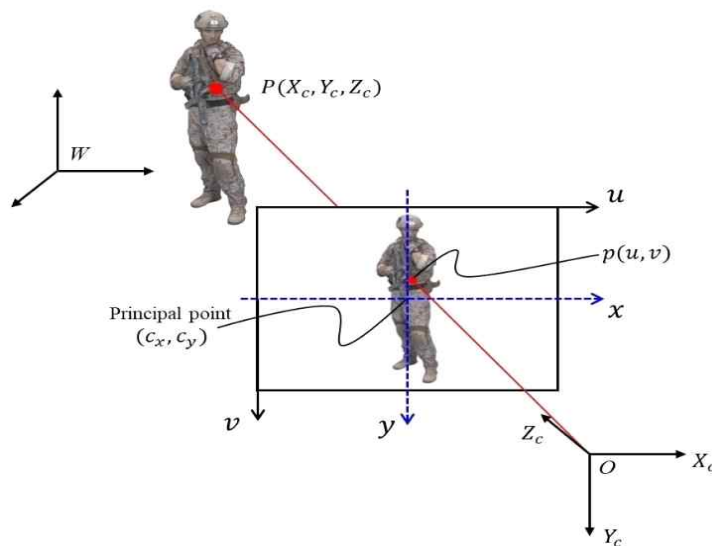


그림 3. 3D 포인트를 2D 이미지 영상 투영

Fig. 3. Projection 3D point cloud content onto a 2D image plane

한다. 소스 포인트에 이동량을 보상해주게 되면 포인트가 완전히 겹치게 된다. 그렇게 되면 전체 포인트의 개수는 증가하지만 완전히 겹침으로 인해 해상도의 변화는 없다. 그러므로 이동량을 보상해준 다음 소스 포인트 p_i 와 인접한 p_j 의 간격을 조정해주는 작업이 필요하다. 아래의 알고리즘을 통해 인접 포인트 간의 간격을 조정하여 포인트들이 완전히 겹치는 것을 방지한다.

표 1. 포인트 간격 조절 알고리즘

Table 1. Adjust the gap of points algorithm

Algorithm 1. Adjust the gap between pair points

$p_{i,j}$: points of frame t , $q_{i,j}$: points $p_{i,j}$ transformed to frame $t+1$, σ : distance compensate-on factor, O : origin points of object coordinate

```

 $\sigma \leftarrow (\|p_i - p_j\| - \|q_i - q_j\|) * 0.5$ 
if  $\|p_i - p_j\| > \|q_i - q_j\|$ 
    if  $O - q_i \leq O - q_j$ 
         $q_i \leftarrow q_i - \sigma, \quad q_j \leftarrow q_j + \sigma$ 
    else
         $q_i \leftarrow q_i + \sigma, \quad q_j \leftarrow q_j - \sigma$ 
else if  $\|p_i - p_j\| \leq \|q_i - q_j\|$ 
    if  $O - q_i \leq O - q_j$ 
         $q_i \leftarrow q_i - \sigma, \quad q_j \leftarrow q_j + \sigma$ 
    else
         $q_i \leftarrow q_i + \sigma, \quad q_j \leftarrow q_j - \sigma$ 

```

$p_{i,j}$ 는 t 번째 프레임의 점이고 $q_{i,j}$ 는 $t+1$ 번째 프레임으로 이동한 $p_{i,j}$ 의 점이다. σ 는 $p_{i,j}$ 의 거리와 $q_{i,j}$ 거리의 차이만큼 $q_{i,j}$ 에 보상해 줄 compensation value이다. 이러한 compensation을 통해서 $q_{i,j}$ 가 $t+1$ 프레임의 점에 완전히 겹쳐지지 않도록 만들어 준다. 포인트 간의 조정이 마무리된 후 소스 포인트와 타겟 포인트를 합쳐 결과 데이터를 만들어 준다.

step 3. Increase Density

세 번째 단계에서는 추가적인 밀도 향상을 위한 작업을 진행한다. 이 단계에서는 첫 번째 단계와 같은 방법으로 진

행한다. 3D 포인트들에 대해서 2D로 projection하여 src_{map} 과 dst_{map} 을 만든다. 3x3 window를 사용하여 dst_{map} 인덱스와 비교하여 정합 되었다고 판단되는 포인트들을 분리시켜 준다. 정합 되어 분리된 포인트를 최종적으로 결과 데이터에 더해지게 되면 포인트 클라우드 데이터의 밀도가 올라가게 된다.

IV. 제안한 방법의 실험

1. 실험 환경

본 실험은 3.6GHz clock의 Intel i7-4790 CPU와 32GB의 메모리의 환경에서 수행한다. 우리가 제안하는 포인트 클라우드 콘텐츠의 해상도 증가를 위한 정합은 오직 CPU 연산만을 통해 수행한다. 또한 Hobart^[29]에서 제공하는 그림 4의 4개의 포인트 클라우드 콘텐츠를 실험에서 사용한다: long dress, soldier, loot, red and black. 포인트 클라우드 콘텐츠들은 각각 300개의 포인트 클라우드 데이터 시퀀스로 구성되어 있다. 각 포인트 클라우드 시퀀스는 약 700,000에서 1,100,000개의 X, Y, Z의 3차원 기하 정보와 이에 대응되는 8bit RGB 색상 정보가 있다. 본 논문에서 제안하는 방법이 포인트 클라우드 콘텐츠의 해상도를 얼마나 향상시키는지 정성적으로 보여주기 위해 원본의 포인트 클라우드 콘텐츠의 렌더링 결과와 비교한다.



그림 4. 실험에 사용될 4개의 포인트 클라우드 콘텐츠. soldier, long dress, red and black, loot

Fig. 4. Four point cloud contents. soldier, long dress, red and black, loot

2. 실험 결과

아래 그림 5에서는 이전 프레임과 현재 프레임 각각의 렌더링 된 모습과 두 프레임을 제안한 방법에 의해 정합한 결과 프레임이다. 본 논문에서 제안하는 방법은 콘텐츠 이미지를 확대했을 경우 생기게 되는 구멍으로 인한 품질 저하 문제를 해결하는 것이다. 제안하는 방법의 품질 향상 정도를 평가하기 위해 프레임 t , 프레임 $t+1$ 과 제안한 방법

의 결과 세 이미지의 PSNR을 수치를 비교하였다. PSNR의 값은 아래 식 6에 의해 획득되며 값이 높을수록 유사도가 증가한다. 비교 기준은 프레임 t 를 확대하기 전, 즉, 품질의 저하가 일어나기 전의 이미지를 다운 샘플링 한 후 이미지를 확대하였다.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) [dB] \quad (6)$$

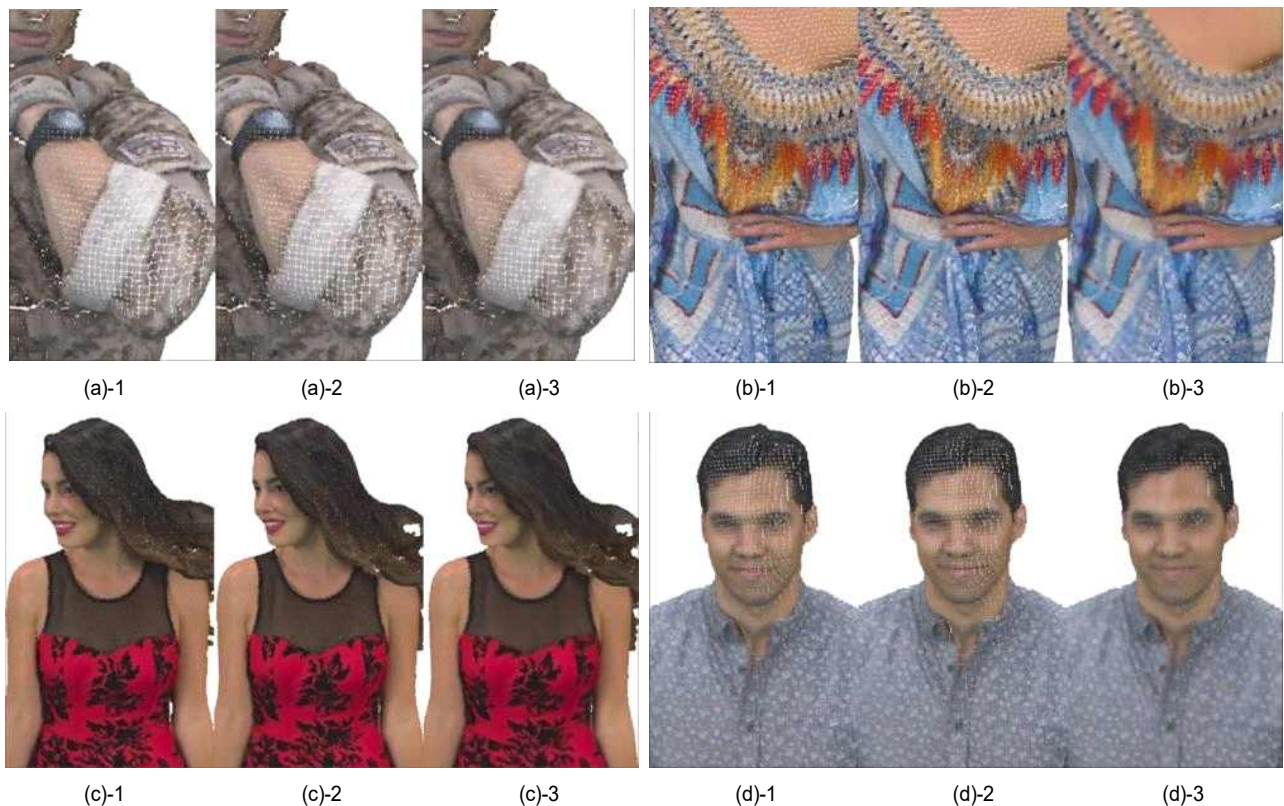


그림 5. 제시한 방법에 의한 렌더링 결과. (a,b,c,d)-1 : 프레임 t , (a,b,c,d)-2 : 프레임 $t+1$, (a,b,c,d)-3 : 결과 영상

Fig. 5. The result of rendering point cloud content with resolution enhanced by our method. (a, b, c, d)-1: frame t , (a, b, c, d)-2: frame $t+1$, (a, b, c, d)-3: enhanced contents

표 2. 콘텐츠 프레임 별 PSNR 수치 비교

Table 2. Comparison of contents frame PSNR

Contents	frame t PSNR[dB]	frame $t+1$ PSNR[dB]	our method PSNR[dB]
Soldier	34.5	31.2	38.4
Long dress	33.2	32.7	35.1
Red and black	37.1	33.2	37.5
Loot	30.3	29.6	33.2

$$MSE = \frac{1}{N^2} \sum_{x=1}^N \sum_{y=1}^N [f_{original}(x, y) - f_{in-between}(x, y)]^2 \quad (7)$$

표 2에서 볼 수 있듯이 프레임 t와 프레임 t+1에 비해서 제안한 방법의 결과 이미지가 PSNR 값이 높음을 확인할 수 있다. 제안한 방법의 결과는 프레임 t 대비 평균 2.3% PSNR 값이 높게 나왔다. 제안한 방법의 정합으로 인해 빈 영역의 부분이 채워지면서 PSNR 값이 상승하게 되었다.

시간복잡도의 문제를 해결하기 위해 보이는 부분에 대해 projection 하여 실험을 진행하였다. 표 3은 3D 포인트 개수와 projection 한 포인트 개수를 비교하고 정합 후 전체 포인트 개수와 실행시간을 나타냈다. 전체 포인트 개수에 비해 projection 하여 보이는 부분의 포인트가 훨씬 줄어들어 계산이 필요한 포인트의 개수가 현저히 감소하였다.

그림 6은 ICP알고리즘을 사용하여 정합 한 결과와 제시한 방법을 사용하여 정합 한 결과를 비교한다. 그림 (a)와 (c)는 ICP 알고리즘을 사용하여 정합 하였고 (b)와 (d)는 제안한 방법으로 정합 한 결과이다. (a)와 (c)에서는 손이나,

발 등 국소적으로 모션 벡터가 다른 영역에 대해서 정합되지 못한 모습을 볼 수 있다. 게다가 ICP알고리즘으로 정합을 할 경우 포인트의 위치 정보만을 사용하여 정합을 한다. 그 결과 (c)의 팔 부분에서 색이 맞지 않는 결과를 확인할 수 있다. 그러나 본 논문에서 제시한 방법으로 렌더링한 결과인 (b)와 (d)는 잘 정합 된 모습을 볼 수 있다.

V. 결 론

포인트 클라우드 데이터의 해상도의 향상을 위한 다양한 방법들이 있다. 그 중 본 논문에서는 다수의 포인트 클라우드 이미지를 정합 하여 밀도를 향상시키는 방법을 사용하였다. 포인트 클라우드를 정합 하는 방법 중 대표적인 방법으로 ICP 알고리즘을 사용하는 방법이 있다. 이 방법을 사용하게 되면 강체에 대해서는 강건하게 정합 할 수 있다. 그러나 비 강체 물체에 대해서는 정합이 불가능 하다. 본 논문에서 제시한 방법은 각 포인트마다의 pair를 찾고 각

표 3. 콘텐츠 별 3D 와 2D 포인트 개수 및 실행시간 비교

Table 3. Comparison of the number of 3D and 2D points of each content and the measured runtime

Contents	# of points of frame t		# of points of frame t+1		# of points after reg.	Runtime (ms)
	Original	2D projection	Original	2D projection		
Soldier	1,052,423	85,811	1,052,450	85,790	247,778	484
Long dress	775,745	84,636	765,821	85,739	237,841	391
Red and black	729,133	105,535	720,568	103,009	294,226	452
Loot	784,142	85,171	783,406	85,226	247,420	431

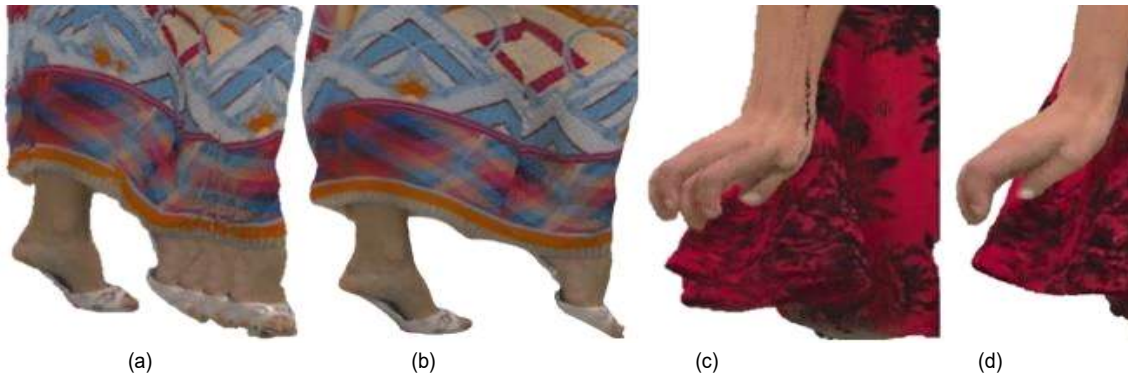


그림 6. 제안한 방법에 의한 결과와 ICP 알고리즘으로 인한 결과 차이 (a), (c) - ICP , (b),(d) - 제안한 방법

Fig. 6. The results of performing registration with our method and ICP algorithm (a), (c) - ICP , (b),(d) - Our method

pair의 이동량을 구해준다. 그 후 각 점들에 대해서 구해준 이동량을 보상해주게 된다. 이 때 완벽하게 겹치는 현상을 방지하기 위해 포인트 간 간격을 조절한다. 이러한 방법을 사용하여 정합을 하게 되면 그림 5에서 볼 수 있듯이 성공적인 정합을 하여 포인트 클라우드 해상도가 향상되게 된다. 또한 표 2에서 볼 수 있듯이 정합 한 결과의 PSNR수치가 가장 높게 나온 것을 볼 수 있다. 포인트 클라우드의 콘텐츠는 3차원의 물체를 나타내기 좋은 도구이다. 그러나 포인트 클라우드는 데이터의 크기가 매우 커 전송 시 encoding과 decoding의 작업이 필요하게 된다^[27]. decoding 과정에서 또는 전송되는 네트워크 상에서 예상치 못한 데이터의 손실이 발생할 수도 있다. 이 때 전체 데이터가 손실되지 않았다면 정합을 통한 데이터의 복구 또한 가능하다고 판단된다. 본 연구에서는 움직임이 큰 부분에 대해서 정합은 잘 되는 결과가 나오지만, 움직임이 적은 부분에 대해서는 약간의 blur가 발생하는 모습을 확인할 수 있다. 추후 연구에서는 움직임이 적은 부분에 대한 보정이 필요할 것으로 생각한다. 또한 정량적 평가를 위해 PSNR을 사용하였지만 PSNR수치는 비교하는 이미지의 유사도를 나타내는 것이 지 품질에 대한 평가 기준이 되기에는 부족하다. 렌더링 품질에 대한 정량적 평가방법에 대한 부분도 보완이 필요할 것으로 보인다.

참 고 문 헌 (References)

- [1] BHAYANI, Sam B.; ANDRIOLE, Gerald L. Three-dimensional (3D) vision: does it improve laparoscopic skills? An assessment of a 3D head-mounted visualization system. *Reviews in urology*, 7.4: 211, 2005.
- [2] D. B. Payne and J. R. Stern, "Wavelength-switched passively coupled single-mode optical network," in *Proc. IOOC-ECOC, Boston, MA, USA*, pp. 585 - 590, 1985.
- [3] MITRA, Niloy J., et al. Registration of point cloud data from a geometric optimization perspective. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, p. 22-31, 2004.
- [4] Arun, Somani; Thomas S. Huang; Steven D. Blostein (1987). "Least-square fitting of two 3-D point sets". *IEEE Pattern Analysis and Machine Intelligence*, 1987
- [5] Besl, Paul J.; N.D. McKay (1992). "A Method for Registration of 3-D Shapes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14 (2): 239 - 256. doi:10.1109/34.121791.
- [6] Chen, Yang; Gerard Medioni (1991). "Object modelling by registration of multiple range images". *Image Vision Computing*. 10(3): 145 - 155. doi:10.1016/0262-8856(92)90 066-C.
- [7] Zhang, Zhengyou (1994). "Iterative point matching for registration of free-form curves and surfaces". *International Journal of Computer Vision*. 13 (12): 119 - 152.
- [8] Rusinkiewicz, Szymon; Marc Levoy (2001). Efficient Variants of the ICP Algorithm. *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. Quebec City, Quebec, Canada. pp. 145 - 152. doi:10.1109/IM.2001.924423.
- [9] Pomerleau, François; Colas, Francis; Siegwart, Roland (2015). "A Review of Point Cloud Registration Algorithms for Mobile Robotics". *Foundations and Trends in Robotics*. 4 (1): 1 - 104. CiteSeerX 10.1.1.709.2212. doi:10.1561/23000000035.
- [10] LOW, Kok-Lim. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 2004, 4.10: 1-3.
- [11] RUSINKIEWICZ, Szymon; LEVOY, Marc. Efficient variants of the ICP algorithm. In: *Proceedings third international conference on 3-D digital imaging and modeling*. IEEE, p. 145-152, 2001.
- [12] SEGAL, Aleksandr; HAEHNEL, Dirk; THRUN, Sebastian. Generalized-icp. In: *Robotics: science and systems*. p. 435. 2009
- [13] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing ICP Variants on Real-World Data Sets. In *Autonomous Robots*, 34(3), pages 133 - 148, April 2013.
- [14] MEN, Hao; GEBRE, Biruk; POCHIRAJU, Kishore. Color point cloud registration with 4D ICP algorithm. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, p. 1511-1516, 2011.
- [15] Holz, Dirk; Ichim, Alexandru E.; Tombari, Federico; Rusu, Radu B.; Behnke, Sven (2015). "Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D". *IEEE Robotics Automation Magazine*. 22 (4): 110 - 124.
- [16] AOKI, Yasuhiro, et al. Pointnetlk: Robust & efficient point cloud registration using pointnet. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 7163-7172, 2019.
- [17] POMERLEAU, François, et al. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31.14: 1705-1711, 2012.
- [18] SCHUETZ, Markus; WIMMER, Michael. Progressive real-time rendering of unprocessed point clouds. In: *ACM SIGGRAPH 2018 Posters*, p. 1-2, 2018.
- [19] HE, Jun; YAO, Xin. Drift analysis and average time complexity of evolutionary algorithms. *Artificial intelligence*, 127.1: 57-85, 2001.
- [20] ZWICKER, Matthias, et al. EWA splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8.3: 223-238, 2002.
- [21] WU, Huikai; ZHANG, Junge; HUANG, Kaiqi. Point cloud super resolution with adversarial residual graph networks. arXiv preprint arXiv:1908.02111, 2019.
- [22] YU, Lequan, et al. Pu-net: Point cloud upsampling network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 2790-2799, 2018.
- [23] PUNET, Xavier, et al. Enhanced cell-material interactions through the

- biofunctionalization of polymeric surfaces with engineered peptides. *Biomacromolecules*, 14.8: 2690-2702, 2013.
- [24] ZHANG, Zhiyuan; DAI, Yuchao; SUN, Jiadai. Deep learning based point cloud registration: an overview. *Virtual Reality & Intelligent Hardware*, 2.3: 222-246, 2020.
- [25] MIELEKAMP, Pieter Maria; HOMAN, Robert Johannes Frederik. *Visualization of 3D images in combination with 2D projection images*. U.S. Patent No 7,991,105, 2011.
- [26] SARAVANAN, G.; YAMUNA, G.; NANDHINI, S. Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. In: *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, p. 0462-0466, 2016.
- [27] HOUSHIAR, Hamidreza; NÜCHTER, Andreas. 3D point cloud compression using conventional image compression for efficient data transmission. In: *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*. IEEE, p. 1-8, 2015.
- [28] SHI, Shaoshuai; WANG, Xiaogang; LI, Hongsheng. Pointcnn: 3d object proposal generation and detection from point cloud. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 770-779, 2019.
- [29] Call for Proposals for Point Cloud Compression v2, Standard ISO/IECJTC1/SC29/WG11 MPEG2017/N16763, Hobart, AU, Apr. 2017.
- [30] RHYU, Sungryeul, et al. Contextual Homogeneity-Based Patch Decomposition Method for Higher Point Cloud Compression. *IEEE Access*, 8: 207805-207812, 2020.
- [31] MIAO, Yongwei; PAJAROLA, Renato; FENG, Jieqing. Curvature-aware adaptive re-sampling for point-sampled geometry. *Computer-Aided Design*, 2009, 41.6: 395-403, 2009.
- [32] CARDANI, Darrin. *Adventures in hsv space*. Laboratorio de Robótica, Instituto Tecnológico Autónomo de México, 2001.
- [33] H.Lee, J.Yun, J.Kim, J.Park, "ICP based Point Cloud Contents Quality Improvement Method," *The Korean Institute of Broadcast and Media Engineers*, p. 200-201, 2020.

저 자 소 개



이 희 제

- 2020년 : 상명대학교 컴퓨터공학과 졸업(학사)
- 2020년 ~ 현재 : 한양대학교 컴퓨터 소프트웨어학과 석사과정
- ORCID : <https://orcid.org/0000-0003-4182-4762>
- 주관심분야 : 포인트 클라우드 렌더링, 3D 컴퓨터 비전, 증강현실



윤 준 영

- 2018년 : 한양대학교 컴퓨터공학과 졸업(학사)
- 2018년 ~ 현재 : 한양대학교 컴퓨터 소프트웨어학과 석박사통합과정
- ORCID : <https://orcid.org/0000-0002-3616-6956>
- 주관심분야 : 비주얼 트래킹, 비강체 오브젝트 트래킹, 물리 기반 가상현실 증강현실 렌더링

저 자 소 개



김 종 욱

- 2019년 : 상명대학교 컴퓨터공학과 졸업(학사)
- 2019년 ~ 2021년 : 한양대학교 컴퓨터소프트웨어학과 졸업(석사)
- ORCID : <https://orcid.org/0000-0001-8079-5859>
- 주관심분야 : 6자유도 오브젝트 포즈 추정, 컴퓨터 비전, 포인트 클라우드 렌더링



김 찬 희

- 2020년 : 서울여자대학교 디지털미디어학과 졸업(학사)
- 2020년 ~ 현재 : 한양대학교 컴퓨터소프트웨어학과 석사과정
- ORCID : <https://orcid.org/0000-0002-1621-202X>
- 주관심분야 : 포인트클라우드 렌더링, 컴퓨터 비전, 증강현실



박 종 일

- 1987년 : 서울대학교 전자공학과 졸업(학사)
- 1989년 : 서울대학교 전자공학과 졸업(석사)
- 1995년 : 서울대학교 전자공학과 졸업(공학 박사)
- 1992년 ~ 1994년 : 일본 NHK방송기술연구소 객원 연구원
- 1995년 ~ 1996년 : 한국방송개발원(현 한국콘텐츠진흥원) 선임연구원
- 1996년 ~ 1999년 : 일본 ATR지능영상통신연구소 연구원
- 1999년 ~ 현재 : 한양대학교 공과대학 교수
- ORCID : <https://orcid.org/0000-0003-1000-4067>
- 주관심분야 : 컴퓨터 비전/그래픽스, 증강현실 가상현실, HCI