쉬어-왑 렌더링을 이용한 LDI 구현

최현상[U] 한정현
성균관 대학교 전기 전자 및 컴퓨터 공학부
컴퓨터 그래픽스 연구실
{rosebud, han}@ece.skku.ac.kr

LDI Implementation using Shear-Warp Rendering

Hyun-Sang Choi[U] JungHyun Han
School of Electrical & Computer Engineering, Sung Kyun Kwan University

요 약

영상 기반 모델링 및 렌더링을 위해 제안된 LDI(Layered Depth Images) 기법은 여러 장의 2차원 영상과 깊이 정보, 카메라 정보를 입력으로 받아 3차원 와핑을 이용해 새로운 장면을 렌더링한다. 하지만 이 기법은 홀 발생 문제 등 몇가지 결함을 가지고 있다. 본 논문은 이러한 LDI의 문제를 해결하고자, 의료 영상 가시화 분야에서 널리 사용되는 쉬어-왑 렌더링 알고리즘을 사용한 결과를 설명한다. 한편, 본 논문에서 제안된 알고리즘은 적은 데이터를 필요로 하는데, 웹 상에서 오브젝트 플레이어 플러그인으로 개발한 결과 좋은 성능을 보였다.

1. 서론

전통적인 기하기반(geometry-based) 컴퓨터 그래픽스의 대안으로 영상기반 모델링 및 렌더링 연구가 활발히 이루어지고 있다. 1998년 Shade가 제안한 LDI(Layered Depth Image)[1]는 픽셀의 3차원 정보를 이용하여 새로운 시점에서 렌더링을 수행하는 기법이다. LDI는 복수개의 2차원 영상 및 이들 픽셀 각각의 깊이 정보와 카메라 정보를 입력으로 받아 들여 3차원 와핑을 통해 하나의 LDI 카메라로 재구성한다. 복수개의 입력 영상을 이용하므로 LDI 카메라의 한 픽셀 위치에는 실제로 여러 개의 픽셀이 동시에 와핑될 수 있는데, 이런 픽셀들은 그 깊이 정보에 따라 계층적으로 저장되게 된다. 이렇게 모델링된 LDI를 렌더링하기 위해서는 McMillan의 ordering 알고리즘[2]과 스플래팅[3]을 사용한다.

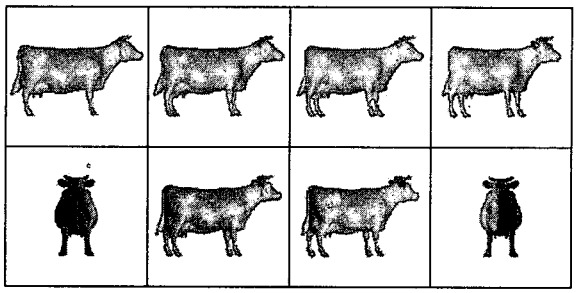

그림 1: LDI입력 영상 (150x150 해상도)

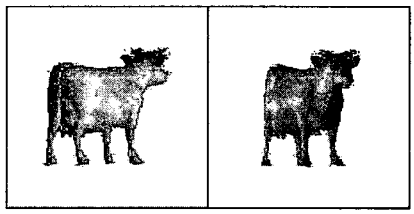

그림 2 : LDI 출력 영상

스플래팅을 사용한 LDI 렌더링의 문제점 중 하나는, 커널의 크기를 정확히 계산하는 것이 어렵다는 것이다. 스플래팅 커널의 크기를 정확히 계산해 내지 못할 경우 출력 영상에 홀 또는 blurring이 발생한다. 그림 1은 LDI를 구성하는 입력 영상을, 그림 2는 렌더링 결과를 보여준다. LDI 렌더링 구현에 있어서는 데이터의 크기를 줄이고자 일정 크기의 스플래팅 커널들만을 사용하였기 때문에 입력 영상들의 시점과 새로운 시점간의 차이가 클 경우 그림 2와 같이 홀이 발생함을 알 수 있다.

본 논문에서는 LDI에서 발생하는 홀 문제점을 개선하고 LDI를 저장하는 자료구조를 개선하여 렌더링 속도를 향상시키고자 복셀을 기반으로 하는 알고리즘을 제안한다. 우리가 제안하는 알고리즘은 전처리 단계와 렌더링의 두 단계로 구성된다. 전처리 단계에선 입력으로 주어진 2차원 영상과 깊이 정보 그리고 카메라 정보를 이용해 3차원 와핑을 통해 LDI와 유사한 일종의 볼륨을 형성

한다. 이를 LDV(Layered Depth Volume)라 하자. 한편, 효율적인 자료구조와 빠른 렌더링을 위해 LDV를 x,y,z 방향으로 runlength 인코딩한다. LDV 렌더링은 뷰 행렬 분할을 통한 쉬어-왑(shear-warp) 렌더링[4]을 이용한다. 여기에는 쉬어링된 이미지에 생기는 홀을 제거해 주는 과정이 포함되며, 이렇게 생성된 중간 이미지를 와핑함으로써 최종 결과 이미지를 얻게 된다. 이렇게 함으로써 LDI에서 생기는 홀을 줄이거나 제거할 수 있었으며, 불필요한 저장 공간을 제거할 수 있었다.

우리가 제안한 방법은 적은 수의 입력 영상을 요구하며 거의 실시간으로 렌더링 될 수 있다. 따라서 우리는 이를 웹에서 작동되는 오브젝트 플레이어로도 구현을 하였다.

2. LDV 생성

입력 영상 픽셀의 깊이 정보와 카메라 정보를 이용해 픽셀의 세계좌표를 계산한 뒤, 이를 LDI 또는 LDV 카메라 뷰 볼륨으로 변환한다. LDV 카메라를 정의하는 4x4 뷰 행렬 M_{LDV}는 뷰 방향 행렬, 뷰 사상 행렬 (또는 투영 행렬) 및 뷰포트 행렬로 구성된다. M_{LDV}에 의해 생성되는 뷰 볼륨은 (x_{res}, y_{res}, z_{res})의 크기를 가지는데, 효율적인 LDV 구현을 위하여 z_{res}는 x_{res}와 y_{res} 중 작은 값을 취하도록 조정했다.

이렇게 구성된 LDV의 각 복셀들은 입력 이미지로부터 얻어온 칼라 값과 이에 해당하는 카메라의 위치 정보를 나타내는 인덱스 값을 가지게 된다. LDV의 하나의 복셀에는 여러 개의 픽셀이 놓여질 수 있다. 따라서 LDV의 한 복셀은 다음과 같이 구성된다.

$$\{(C_1,Cp_1),(C_2,Cp_2),(C_3,Cp_3),\ldots(C_m,Cp_m)\}$$

(C: 칼라 값, Cp: 입력 영상의 카메라 위치 인덱스)

이렇게 한 픽셀마다 카메라 위치 인덱스를 할당하는 이유는 Debevec[5]이 제안한 뷰-종속적(view-dependent) 텍스처 매핑을 적용하기 위해서이다. 즉, 새로운 시점이 주어지면, 이 시점과 비슷한 방향을 가진 시점들을 추출하고, 이 시점들에서 얻은 칼라 값들을 보간하여 새로운 시점에서의 복셀 칼라 값을 결정하는 것이다. (그림 3 참조.) 한편, 복셀을 구성하는 픽셀의 수를 줄이기 위해서, 비슷한 방향을 가지는 시점들의 경우, 해당 픽셀들을 보간하여 하나의 픽셀로 대치하였다.

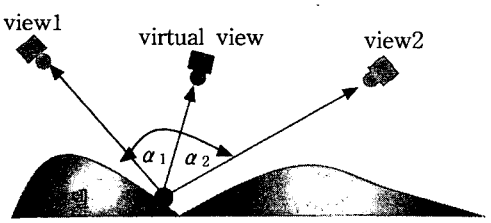

그림 3 : 인접한 두 시점에서의 칼라 값을 이용해 새로운 시점에서 칼라 값을 계산하는 방법([5]참조.)

이렇게 만들어진 LDV에서 실제 복셀이 존재하는 부분은 전체 볼륨에 비해 매우 적게 된다. 따라서 불필요하게 기억장치를 많이 소모하고, 렌더링 시에도 데이터 값이 존재하지 않는 곳까지 처리하게 되어 수행 속도를 저하시킨다. 본 논문에선 이런 문제점을 제거 하고자 쉬어-왑에서와 같이 LDV를 x; y, z축에 대해서 각각 runlength 인코딩 하였다. 이는 전처리 단계의 일부로, 주어진 입력 영상에 대해서 한번만 수행하게 된다.

3. 쉬어-왑을 이용한 렌더링

LDV 공간상에 존재하는 복셀을 쉬어-왑 렌더링을 사용해 렌더링 하기 위해선 뷰 행렬에 대한 재조정이 필요하다. 따라서 본 절에서는 뷰 행렬을 재조정하는 과정과 이를 분할하는 과정에 대해서 설명하고, 마지막으로 출력 영상에 발생하는 홀을 제거하는 방법에 대해서 설명하겠다.

새로운 시점에서 뷰 행렬 M_{view}가 계산되면 M_{LDV}를 이용하여 다음과 같이 쓸 수 있다. 여기에서 $[x_o,y_o,z_o,w_o]^T$는 복셀의 좌표이고, $[x_i,y_i,z_i,w_i]^T$는 출력 영상 픽셀의 좌표이다.

$$\begin{aligned}[x_i,y_i,z_i,w_i]^T &= M_{view}\cdot[x_o,y_o,z_o,w_o]^T \quad \text{--------} \quad (1)\\ &= M_{view}\cdot M^{-1}{}_{LDV}\cdot M_{LDV}\cdot[x_o,y_o,z_o,w_o]^T\\ &= \overline{M}_{view}\cdot M_{LDV}\cdot[x_o,y_o,z_o,w_o]^T\end{aligned}$$

이렇게 재조정된 뷰 행렬을 $\overline{M}_{view}$라 하면 이는 다음과 같이 분할 할 수 있다.

$$\overline{M}_{view} = \overline{M}_{warp}\cdot\overline{M}_{shear}\cdot\overline{P} \quad \text{---------} \quad (2)$$

수식 (2)에서 얻은 쉬어링 행렬 $\overline{M}_{shear}$를 이용하여 복셀 스캔 라인과 이미지의 스캔 라인을 따라 가면서 중간 이미지를 생성해 간다. 이렇게 생성된 중간 이미지는 홀을 가질 수 있다. 이런 홀을 제거하는 과정은 다음과 같다. 전통적인 쉬어-왑 렌더링과는 다르게 중간 이미지를 생성하면서 중간 이미지에 투영된 픽셀의 깊이 정보를 z-buffer에 저장한다. 이렇게 저장된 각 픽셀의 깊이 정보를 이용하여 렌더링된 중간 이미지에서 홀을 검출할 수 있는데, 홀을 검출하기 위해서는 일정 크기의 윈도우를 설정하여 현재 픽셀과 주변 픽셀들 사이의 깊이값 차이의 합이 일정한 기준치(threshold)를 초과하면 홀로 간주하고 이 홀은 칼라 값을 저장하고 있는 중간 이미지의 프레임 버퍼의 칼라값을 가우시안 필터를 사용하여 채우므로써 제거한다. 이렇게 홀이 제거된 중간 이미지를 $\overline{M}_{warp}$를 이용해서 bilinear 필터를 사용한 backward projection으로 와핑함으로써 최종 결과 이미지를 얻는다.

그림 4는 홀 제거 과정을 생략하고 얻은 최종 출력 영상을 보여준다. 이 영상들은 LDI에서 사용한 그림 1의 입력 영상을 그대로 이용하였고, 그림 2의 시점과 동일한 시점에서 렌더링된 것이다. LDI에 비해 다소 많은 수의 홀을 가지지만, 각각의 홀의 크기는 LDI의 것보다 작음을 알 수 있다. 그림 4와 같은 영상에서 홀 제거를 마친 결과가 그림 5이며, 여기서 우리는 기존의 LDI 방법

보다 더 나은 결과 이미지를 얻을 수 있었다.

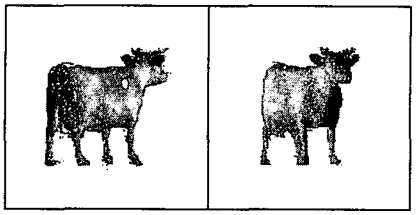

그림 4 : 홀 제거 과정을 생략한 출력 영상

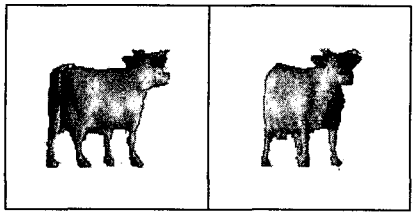

그림 5 : 홀을 제거한 출력 영상.

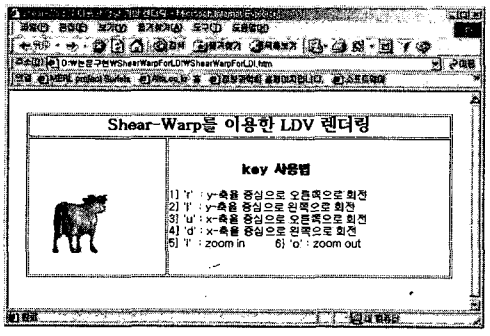

그림 6 : ActiveX로 제작된 플러그인

4. 플러그인 구현

본 논문에서 제안된 알고리즘은 웹 상에서 3차원 물체를 디스플레이 하기 위해서 Visual C++ 6.0 환경에서 ActiveX로 제작되었다. (그림 6 참조.) 우리가 구현한 결과는 [그림 1]에서처럼 물체를 중심으로 8장의 2차원 영상과 이에 대응하는 깊이 정보 그리고 카메라 정보를 입력으로 받아 [그림 6]에서 보는 것과 같이 자연스럽게 물체를 확대 혹은 축소 해 볼 수 있으며 또한 물체 주위를 둘러 볼 수 있다.

5. 결론 및 향후 과제

본 논문에서는 LDI의 렌더링 방법을 개선하여 기존의 LDI의 홀 발생 문제와 LDI 자료 구조에서 불필요한 메모리 낭비를 개선할 수 있었다. 또한 그 응용으로 웹에서 동작하는 오브젝트 플레이어를 개발하였다. 이 논문에서 제안한 알고리즘은 적은 수의 입력 데이터를 이용하여 자연스럽게 물체를 회전 및 확대, 축소해 볼 수 있다. 하지만 우리가 개발한 오브젝트 플레이어는 각 픽셀의 정확한 깊이 정보를 필요로하므로 입력으로 합성 장면만을 사용하고 있다. 컴퓨터 비젼 기술을 이용해 실제 사진으로부터 깊이 정보와 카메라 정보를 추출할 수 있다면 실제 사진을 입력으로 받아들이는 오브젝트 플레이어로도 확장 가능하며, 이를 위해 실제 사진으로부터 깊이 정보를 추출하는 연구가 현재 진행중이다. 또한 본 논문에서 제안된 알고리즘은 물체와 배경의 경계면에서 다소 부자연스럽게 처리되는 단점이 있다. 좀 더 사실적인 오브젝트 플레이어의 기능을 제공하기 위해서 이부분에 대한 처리도 연구중이다.

6. 참고 문헌

1.Jonathan Shade, Steven Gortler, Li-wei He, Richard Szeliski,"Layered Depth Images",SIGGRAPH98,1998.
2.Leonard McMillan, "Computing Visibility Without Depth", Technical Report 95-047, University of North Caronlina, 1995.
3.Lee Westover, "Footprint Evaluation for Volume Rendering", In Forest Baskett, editor, Computer Graphics(SIGGRAPH'90 proceedings) volume 24, pages 367-376, August 1990.
4.Philippe Lacroute, Marc Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation", Proc. of SIGGRAPH'94,pp 451-458,1994.
5.Paul E.Debevec,Camillo J. Taylor, Jitendra Malik, "Modeling and Rendering Architecture from Photographs: A hybrid geometry-and image-based approach", SIGGRAPH96, 1996.