

Relief Texture 매핑을 이용한 가상공간 구축

이은경, 정영기

호남대학교 컴퓨터공학과

전화 : 062-940-5404 / 핸드폰 : 016-610-0643

Construction of Visual Space using Relief Texture Mapping

Eun Kyung Lee, Young Kee Jung

Dept. of Computer Engineering, Honam University

E-mail : lek780911@empal.com

Abstract

Recently, several methods have been developed for the virtual space construction. Generally, most of the methods are geometric-based rendering technic, but they are difficult to construct real-time rendering because of large data. In this paper, we present a three dimension image-based rendering method that enable a constant speed of real-time rendering regardless of object complexity in virtual space. The proposed method shows good performance for the virtual space construction with high complexity.

I. 서론

가상공간을 구축하기 위한 가상객체 표현기술은 대부분 기하학적으로 자세하게 모델링하여 장면을 구성하고 렌더링하였다. 그러나 객체를 생성하는 과정이 많은 시간과 노력을 필요로 할 뿐만 아니라 데이터에 있어서도 그 양이 방대하기 때문에 실시간 응용분야에서는 많은 어려움이 있다. 이와 같은 기존의 기하모델 기반 가상공간 구축의 한계를 극복하기 위해 최근 이미지 기반 모델링이라는 새로운 접근 방식이 등장하였다[2]. 이미지 기반의 모델링 및 렌더링은 그림이나 사진등의 이미지를 입력받아 모델링과 렌더링에 필요한 정보를 얻어서 장면을 구성함으로써 입력 받은 이미지와 같은 수준의 사실적인 장면을 연출해낼 수 있을 뿐

만 아니라 출력 장면의 생성 속도가 기하학적 복잡도에 상관없이 복잡한 모양의 객체들을 포함하고 있는 이미지를 가상공간 체험자가 바라보는 시각에 따라 실시간으로 생성, 렌더링할 수 있다.

본 논문에서는 기존의 이미지 기반 모델링 방법의 단점이라고 할 수 있는 기하정보의 결함을 극복하기 위해 모델기반 스테레오(model-based stereo) 방법을 이용한 깊이 정보 추출과 이 깊이 정보를 이용한 relief texture 매핑 방식을 적용함으로써 입체감 있는 가상공간을 설계 및 구현해 보고자 한다.

II. Relief Texture 매핑을 이용한

가상공간 설계 및 구축

제안된 가상공간 설계 및 구축은 그림 1과 같이 모델 기반 스테레오, 선 워핑, 텍스처 매핑의 3단계로 나뉘어진다.

먼저 첫 번째 단계인 모델 기반 스테레오 단계는 입력된 이미지 간의 스테레오 상관성을 통하여 이미지 객체 간의 깊이 정보를 알아낼 수 있는 이미지 기반 알고리즘이다[4]. 두 번째 단계인 선 워핑 단계는 기존의 텍스처 매핑 방법의 확장된 개념으로서, 텍스처 이미지의 컬러 값과 전 단계인 모델 기반 스테레오 방법을 통해 얻은 깊이 정보를 결합시켜 하나의 텍스처 이미지를 생성한다. 마지막 단계인 텍스처 매핑은 이렇게 얻은 새로운 텍스처 이미지를 최종적으로 현재

바라보는 위치에서의 가상공간에 적용시킨다[1].

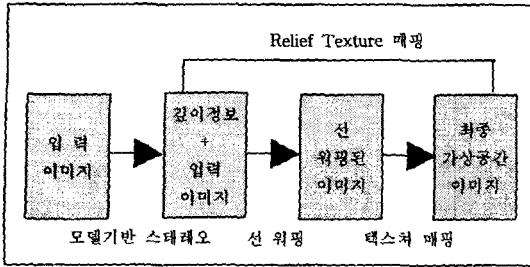


그림 1. Relief texture 매핑을 이용한 가상공간 구현의 구성도

2.1 모델 기반 스테레오

기준 이미지(key image), 오프셋 이미지(offset image), 그리고 워핑된 오프셋 이미지(warped offset image) 간의 관계를 에피폴라(epipole) 공간에 놓고, 기준 이미지와 워핑된 오프셋 이미지 간의 깊이 값이 0(zero)인 관계를 이용하여 각 이미지 안 객체들 간의 상대적인 깊이를 추출해낼 수 있다. 이는 즉, 오프셋 이미지를 기준 이미지에서 바라본 시점 파라미터로 워핑시킨 워핑된 오프셋 이미지는 이론적으로 기준 이미지와 일치해야한다. 그러나 각 객체간의 깊이 정보가 다르기 때문에 두 이미지 간에 차이가 발생하게 된다. 모델 기반 스테레오 방법은 같은 시점 파라미터에서의 두 이미지의 이러한 차이값을 이용하여 이미지 내 객체 간의 상대적 깊이 정보를 추출한다[4].

2.2 Relief texture 매핑

Relief texture 매핑은 기본적으로 물체를 둘러싼 육면체를 만들고 각 여섯 면에 붙일 텍스처를 계산하여 렌더링하는 알고리즘이다. 이는 먼저 입력된 이미지를 현재 시점 파라미터에 따라 보여지는 면을 계산, 각 면의 중간 이미지를 생성하고 생성된 중간 이미지를 보여지는 각 면에 매핑시킨다[1][2].

2.2.1 선 워핑

이미지 기반 렌더링의 대표적인 방법으로 워핑(warping)을 사용한다[3]. 워핑은 기준이 되는 시점 \dot{C}_s 에서의 이미지의 각 픽셀에 대한 투영정보 P 와 깊이 값 $\delta_s(u_s, v_s)$ 를 이용하여 새로운 시점 \dot{C}_t 를 추출해 낸다.

$$\vec{x}_t = P_t^{-1} P_s \vec{x}_s + P_t^{-1} (\dot{C}_s - \dot{C}_t) \delta_s(u_s, v_s)$$

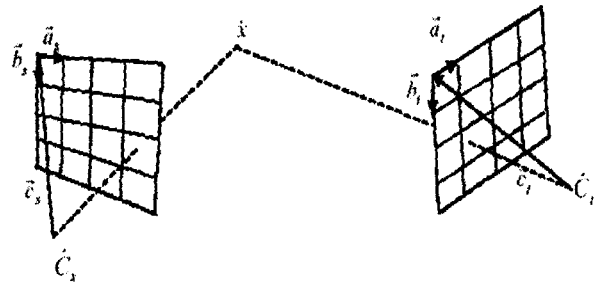


그림 2. 기준 시점에서 새로운 시점에서의 워핑

그림 2에서 보는 것처럼 기준 시점에서 새로운 시점으로의 워핑과정을 살펴보면 점 x_s 는 평행 투영된 이미지 위의 점이며, (u_s, v_s) 로 나타낸다. 이 점에 대응되는 유클리디언 좌표(Euclidean Space)상에 있는 \dot{x} 는 아래 식과 같이 계산된다. a 는 이미지 평면의 가로방향 벡터, b 는 세로방향을 나타내는 기저(Basis) 벡터이며, f 는 평면의 수직방향을 나타낸다.

$$\dot{x} = \dot{C}_s + \begin{bmatrix} a_{si} & b_{si} & c_{si} \\ a_{sj} & b_{sj} & c_{sj} \\ a_{sk} & b_{sk} & c_{sk} \end{bmatrix} \begin{bmatrix} u_s \\ v_s \\ displ(u_s, v_s) \end{bmatrix} = \dot{C}_s + P_s^{-1} \vec{x}_s'$$

\dot{x} 는 다시 원근 투영(Perspective Projection)되어 이미지 위의 한 점 x_t 로 변환된다.

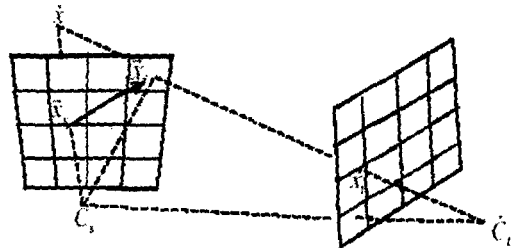


그림 3. $x_s \rightarrow x_t$ 로의 변환 과정

$$u_t = \frac{Au_s + Bv_s + D + C'displ(u_s, v_s)}{Iu_s + Jv_s + L + K'displ(u_s, v_s)}$$

$$v_t = \frac{Eu_s + Fv_s + H + G'displ(u_s, v_s)}{Iu_s + Jv_s + L + K'displ(u_s, v_s)}$$

where $A = \vec{a}_s \cdot (\vec{b}_t \times \vec{a}_t)$, $B = \vec{b}_s \cdot (\vec{b}_t \times \vec{a}_t)$, $C = \vec{f}_s \cdot (\vec{b}_t \times \vec{c}_t)$,
 $D = (\dot{C}_s - \dot{C}_t) \cdot (\vec{b}_t \times \vec{a}_t)$, $E = \vec{a}_s \cdot (\vec{c}_t \times \vec{a}_t)$, $F = \vec{b}_s \cdot (\vec{c}_t \times \vec{a}_t)$,
 $G = \vec{f}_s \cdot (\vec{c}_t \times \vec{a}_t)$, $H = (\dot{C}_s - \dot{C}_t) \cdot (\vec{c}_t \times \vec{a}_t)$, $I = \vec{a}_s \cdot (\vec{a}_t \times \vec{b}_t)$,
 $J = \vec{b}_s \cdot (\vec{a}_t \times \vec{b}_t)$, $K = \vec{f}_s \cdot (\vec{a}_t \times \vec{b}_t)$, $L = (\dot{C}_s - \dot{C}_t) \cdot (\vec{a}_t \times \vec{b}_t)$

그림 3에서 x_0 가 x_1 로 변화되는 과정을 보여준다. 이 때 x_0 에서 x_1 로의 변환 과정은 이미지 자체의 변환이므로 이미지 그 자체를 물체로 보아 깊이 정보가 0인 계산을 수행하는 것으로 볼 수 있다. 따라서 x_0 에서 x_1 로의 변환은 다음과 같은 식으로 나타낼 수 있다.

$$\frac{Au_i + Bv_i + D}{Iu_i + Jv_i + L} = \frac{Au_s + Bv_s + D + C'displ(u_s, v_s)}{Iu_s + Jv_s + L + K'displ(u_s, v_s)}$$

$$\frac{Eu_i + Fv_i + H}{Iu_i + Jv_i + L} = \frac{Eu_s + Fv_s + H + G'displ(u_s, v_s)}{Iu_s + Jv_s + L + K'displ(u_s, v_s)}$$

이 식을 x_i , 즉 (u_i, v_i) 에 대해 정리하면 다음과 같다.

$$u_i = \frac{u_s + k_1 displ(u_s, v_s)}{1 + k_2 displ(u_s, v_s)}$$

$$v_i = \frac{v_s + k_2 displ(u_s, v_s)}{1 + k_3 displ(u_s, v_s)}$$

2.2.2 선 워핑된 텍스처의 재건

앞 섹션에서 보여준 것처럼 기준 시점에서 중간 이미지 x_0 로의 좌표를 결정하는 방법을 보여주었다. 이렇게 결정된 좌표 위치로 실제 재건이 이루어지는 방법에 있어서 이 논문에서는 2D 변환이 아닌 가로, 세로 두 단계의 1D 변환을 통해 이를 구현한다.

2.2.2.1 두 단계의 재건

하나의 전체 이미지를 만들어가는 과정은 먼저 가로 방향을 재건하고 세로방향을 재건해 나가는 방법을 사용하였으며 중간에 생기는 빈 픽셀들은 보간(Interpolation)방법을 통해 재건되어진다.

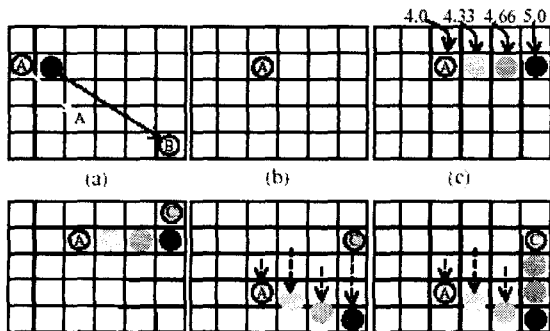


그림 4. 텍셀의 두 단계 재건 방법

그림 4에 나온 재건 과정은 살펴보면 먼저 각 픽셀 A와 B의 워핑 후의 위치가 아래 점원 안의 A와 B이다. 이것은 먼저 가로방향의 가장 마지막 픽셀위치부터 가로열의 끝까지 각 픽셀간의 값을 보간한다. 후에 같은 방법으로 세로방향으로 보간을 실행한다.

2.3 모델링

최종적으로 생성된 이미지를 가지고 하나의 객체를 생성하기 위해 각 면에 맞는 이미지를 매핑시킨다. 그리고 그 객체를 바라보는 시점 파라미터에 따라 각 이미지를 워핑시켜줌으로써 이 객체는 이미지가 아닌 하나의 기하 기반 렌더링과 같은 렌더링 결과를 생성하게 된다[1][2].

2.3.1 객체 재건

객체 렌더링의 세부 내용을 살펴보면 시점에 따라 보여지는 면을 계산하고 그 보여지는 시점에 맞게 이미지를 워핑함으로써 하나의 3차원 객체가 생성되게 된다.

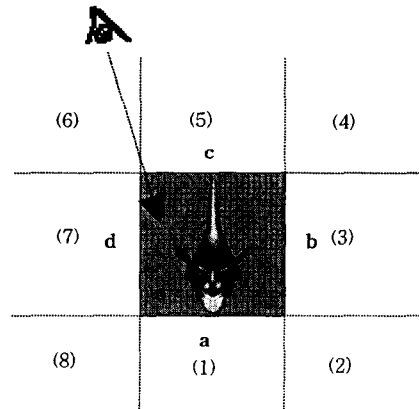


그림 5. 시점에 따른 생성할 면 선택

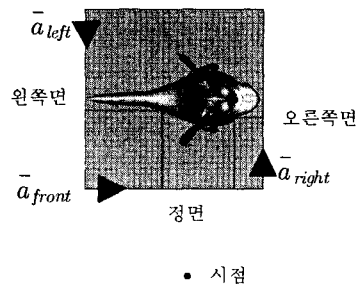


그림 6. 육면체의 각 면간의 관계

그림 6은 그러한 시점을 계산하는 방법 및 고려되어야 하는 면을 설명한 것이다. 그림에 따르면 시점이 현재 위치에 있을 경우 보여지는 면은 (5), (6), (7) 면이다. 이 세 개의 면은 하나의 시점 파라미터로 워핑되어진다. 그리고 서로 직각의 관계에 있는 6개의 면들은 상대적인 관계를 가지고 있다. 정면의 깊이 값은 오른쪽 면에 있어서는 가로 값이 되며, 왼쪽 면의 가로 값은 정면의 깊이 값이 된다.

III. 결과

다음 그림 7은 실제 기하 기반 방법과 이미지 기반 방법으로 렌더링된 결과를 보여주고 있다.



그림 7. 기하 기반 렌더링(왼), 이미지 기반 렌더링(오른)

그림 8은 두개의 면이 하나의 시점 파라미터를 향해 워핑되어져서 매핑된 결과를 보여주고 있고, 그림 9는 그 각 면에 매핑된 워핑된 텍스처를 보여주고 있다.

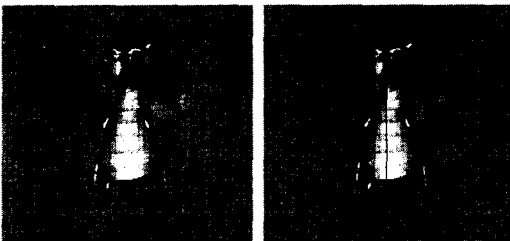


그림 8. 두 워핑된 이미지의 이미지 기반 렌더링

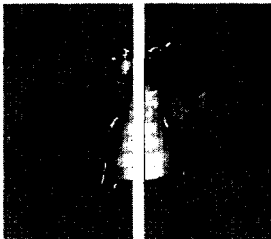


그림 9. 그림 8 객체 생성에 사용된 선 워핑된 텍스처

그림 10은 생성된 최종 3차원 객체를 각기 다른 시점에서 렌더링 시킨 결과이다. 결과에 있어 기하 기반

렌더링 방법과 비교하여 렌더링 시간과 렌더링된 객체에 있어 좋은 결과를 보임을 확인할 수 있다.

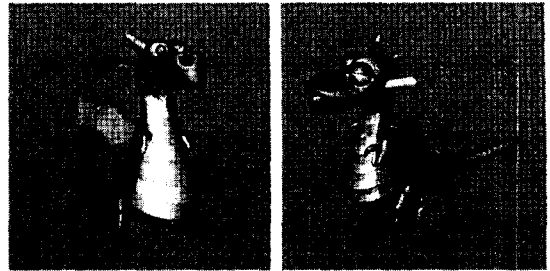


그림 10. 다른 시점에서의 이미지 기반 렌더링 결과

IV. 결론 및 향후 연구

본 논문에서는 한 시점 파라미터를 향한 육면체의 6면의 보여지는 면에 있어서는 이미지 워핑과 텍스처 매핑 작업을 통한 3차원 객체 렌더링 방법을 제안하였다. 제안된 방법은 좀 더 복잡하고 큰 규모의 가상공간 구축의 실시간 렌더링의 기본 단위로 사용될 수 있는 좋은 결과를 보였다.

향후에는 이미지 기반 렌더링의 단점인 상호작용의 부족을 극복하기 위한 3차원 기하기반(geometric-based) 데이터와의 결합 및 인터넷 Web3D 기술과의 결합인 이미지 기반 렌더링에 적합한 효율적인 VRML 노드 구성에 대한 연구도 수행할 예정이다.

V. 참고문헌

- [1] Oliveira, M. Relief Texture Mapping. Ph.D. Dissertaton. UNC Computer Science Technical Repost TR00-009. March 2000 <http://www.cs.unc.edu/~ibr/pubs/oliveira-diss/TR00-009.pdf>
- [2] Junae Kim. Intergrated Use of 2D and 3D Geometric Models for Virtual Objects. *Master's Thesis*, Dept of CSE, POSTECH, April 2002.
- [3] McMillan, L. An Image-Based Approach to Three-Dimensional Computer Graphics. Ph.D. Dissertation. UNC Computer Science Technical Report TR97-013, April 1997.
- [4] Debevec, P., Taylor, C., Malik, J. Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach. *Proc. SIGGRAPH 96* (New Orleans, LA, August 4-8, 1996, pp. 11-20.