3D 영상 기반 파노라마를 이용한 가상 환경 생성*

김 세 환°, 우 운 택
광주과학기술원 U-VR 연구실

Virtual Environment Generation using 3D Image-based Panorama

Schwan Kim°, Woontack Woo
KJIST U-VR Lab.
{skim, wwoo}@kjist.ac.kr

요 약

본 논문에서는 다수의 멀티뷰 카메라를 사용하여 영상의 3D 깊이 정보에 기반한 파노라믹 영상 생성과 이를 통한 가상 환경 구축 방법을 제안한다. 일반적으로 2D 영상에 기반한 모자이킹 방법은 제작이 용이할 뿐만 아니라 사용자 주위의 장면에 대해 확장된 넓은 시야를 제공함으로써 사용자에게 사실감을 제공한다는 등의 장점 때문에 특히 파노라마 제작에 많이 사용되고 있다. 그러나 이와 같은 특징에도 불구하고 사용자가 네비게이션 할 수 있는 가상 환경을 제공하는 데에는 한계를 지닌다. 따라서, 이와 같은 문제점을 해결하기 위해 본 논문에서는 두 대의 멀티뷰 카메라를 캘리브레이션 하고, 획득된 각 영상의 3D 깊이 정보를 이용하여 각 카메라로부터 획득된 3D point cloud 를 스티칭한다. 그리고 스티칭된 point cloud 를 텐테이블을 사용하여 확장함으로써 영상 기반 파노라마를 이용한 3D 가상 환경을 생성한다. 이와 같은 방법을 사용함으로써 사용자가 일정한 반경 내에서 가상 환경을 네비게이션하는 것이 가능할 뿐만 아니라 사용자에게 몰입감을 제공할 수 있다.

1. 서 론

파노라믹 영상 모자이크 (Panoramic Image Mosaics)는 영상 획득 장비를 이용하여 장면에 대한 일련의 영상들을 얻은 후, 이를 결합하여 사용자가 단일 방향만이 아니라 주위의 모든 방향을 바라볼 수 있도록 하기 위한 합성된 영상을 의미한다. 파노라믹 영상은 카메라와 같은 영상 획득 장비로 얻는 단일 영상과 비교해 사용자 주위의 장면에 대한 넓은 시야 (Field Of View: FOV)를 제공한다. 또한 단일 영상과 비교해 사용자에게 사실감뿐만 아니라 보다 몰입감을 제공한다는 면에서 큰 의미를 갖는다. 여러 가지 장점 때문에 파노라마 제작 프로그램이 현재 상업적으로도 많이 사용되고 있는데 Apple 사의 QuickTime VR [1]을 그 대표적인 예로 들 수 있다.

한편, 파노라믹 영상 모자이크 구성 과정은 영상 기반 가상 현실 시스템 (Image-based Virtual Reality system: IBVR)을 구축하는데 있어 가장 기본이 되는 모델링 과정 중의 하나이다 [2]. 최근, IBVR 은 컴퓨터 비전 뿐만 아니라 컴퓨터 그래픽스 영역에서도 많은 관심의 대상이 되고 있다. 이는 기존의 3D 모델에 기반한 가상 현실 시스템에 비해 IBVR 이 실사를 사용함으로써 사실감을 제공할 뿐만 아니라 렌더링 과정도 간단하다는 장점 때문이다.

그 동안 파노라믹 영상 모자이크를 생성하기 위한 여러 가지 방법들이 제안되었다. Nayar 가 제안한 Catadioptric Omnidirectional Camera [3]는 파노라믹 영상 제작을 용이하게 하지만, 영상 왜곡 요소들을 포함하고 있어 보정 작업이 요구된다. 캘리브레이션 되지 않은 카메라를 이용하여 동일한 시점에서 촬영된 영상들로부터 파노라마를 구성하는 방법 [4][5], 그리고 시점을 움직이면서 촬영된 영상을 이용하여 파노라마를 구성하려는 방법도 연구되고 있다. [6][7]. 그러나 대부분의 파노라믹 영상 제작 시스템은 2D 영상에 기반하며, 사용자에게 3D 공간이라는 느낌을 전달하기에는 한계를 지닌다.

한편, 파노라믹 3D 비젼 시스템은 실세계 공간에 대한 깊이 정보를 획득하고, 이를 기반으로 파노라믹 영상에 기반한 가상 환경을 구성한다. Peleg 등은 비중심 회전 카메라를 사용하여 서로 다른 양안차를 얻기 위해 기준선을 변경시키면서 가상 파노라마를 구성한다 [8]. Shum 과 Szeliski 는 여러 다른

* 본 연구는 한국전자통신연구원 (과제번호: ETRI 02-0269)의 지원에 의하여 수행됨

시점에서 깊이 정보를 획득함으로써 비슷한 시스템을 구성하였다 [9]. 반면, Benosman 과 Devars 는 두 개의 원통형 투영 영상을 생성하기 위하여 nodal point 를 포함하는 축을 중심으로 두 개의 선형 영상 센서를 회전시킨 후 깊이 맵을 생성시켰다 [10]. 그러나 3D 비전에 기반한 대부분의 시스템들은 사용자의 네비게이션이 불가능하기 때문에 몰입감을 제공하는 데에는 한계를 지닌다.

본 논문에서는 3D 깊이 정보에 기반한 스티칭 방법을 제안하고 이를 확장하여 사용자가 네비게이션할 수 있는 실사에 기반한 가상 환경을 생성한다. 우선, 두 대의 멀티뷰 카메라를 캘리브레이션하고 각 카메라로부터 획득된 영상을 가상 공간에 역투영한다. 그리고 두 카메라로부터 획득된 3D 깊이 정보를 갖는 point cloud를 스티칭하여 3D 가상 환경을 구성하고, 배경 제거 과정을 통해 추출된 정적 객체를 여러 층으로 나누어 각각 삽입한다. 마지막으로, 동적 객체를 실시간으로 분할하고 이를 가상 환경에 삽입함으로써 사용자가 가상 환경을 네비게이션할 수 있다. 또한, 스티칭된 결과를 턴테이블(turn-table)을 이용하여 확장함으로써 사용자에게 몰입감을 제공할 수 있는 가상 환경을 구축하였다.

본 논문의 구성은 다음과 같다. 2장에서는 삼차원 가상 환경 생성 방법 및 네비게이션에 대해 설명한다. 그리고 3장에서는 실험 및 분석에 대해 기술한 뒤, 4장에서 결론 및 추후 과제에 대해 언급한다.

2. 삼차원 가상 환경 생성 및 네비게이션

3D 깊이 정보를 이용한 실사 기반 가상 환경은 2D 기반 파노라마가 갖는 한계를 극복하고 사용자에게 자연스런 네비게이션을 제공함으로써 보다 큰 현실감과 몰입감을 제공한다. 이를 위해서는 3D 깊이 정보의 획득, 깊이 기반 스티칭, 그리고 파노라믹 영상 생성 방법 등이 요구된다.

2.1 화이트 밸런싱

USB 나 IEEE 1394 를 활용하는 디지털 카메라는 아날로그 카메라보다 상대적으로 가격이 저렴한 반면, CCD 센서는 빛에 민감해 색의 전이나 포화와 같은 해결해야 할 문제점이 있다. 또한 한 대 이상의 카메라를 사용하는 경우에는 각 카메라의 특성이 서로 다르기 때문에 동일한 장면에 대해서 조차도 상이한 영상이 획득될 수 있다. 이러한 문제점을 해결하기 위해 화이트 밸런싱이 요구된다.

이를 위해 우선 획득된 영상의 R, G 그리고 B 컬러 채널에 대해 각각 평균과 표준편차를 구한다. 그리고 영상의 각 화소값에 이로부터 도출된 일정한 값을 더함으로써 실시간으로 얻어지는 영상이 실제 장면과 동일한 색을 유지할 수 있도록 하였다 [11]. 획득된 영상의 빨간색 요소에 대한 평균과 표준편차가 m_{Red} 과 σ_{Red} 라면 이에 대한 보상값 α는 수식 (1)과 같다. 녹색과 파란색에 대한 보상값 β 그리고 γ의 값도 비슷한 방법으로 구할 수 있다.

$$\begin{aligned}\alpha &= 255 - m_{Red} - 2\sigma_{Red} \\ \beta &= 255 - m_{Green} - 2\sigma_{Green} \\ \gamma &= 255 - m_{Blue} - 2\sigma_{Blue}\end{aligned} \quad (1)$$

2.2 카메라 캘리브레이션 및 깊이 기반 스티칭

기존의 2D에 기반한 영상 모자이킹 방법과의 근본적인 차이는 제안된 파노라믹 영상 생성 과정이 3D에 기반한다는 것이다. 이를 위해 카메라간의 기하학적 위치 관계를 사용하여 두 영상의 point cloud 를 스티칭한다. 이를 위해서는 카메라들간의 기하학적인 정보를 활용하여 정확한 3D 정보를 얻어야 한다. 즉, 외부 카메라 모델과 내부 왜곡 모델을 결정하는 캘리브레이션 과정이 요구된다. [12]. 본 논문에서는 Zhang의 캘리브레이션 알고리즘의 수정 버전에 근거하여 수행하였다 [13].

평면 패턴을 사용하는 Zhang의 알고리즘은 카메라 파라미터를 쉽게 얻을 수 있다는 장점이 있다. 반면, 근거리에서는 신뢰할 만한 결과를 얻지만, 카메라로부터 거리가 멀어짐에 따라 정확도가 상당히 떨어진다. 따라서 본 논문에서는 이미 세 개의 렌즈간에 캘리브레이션 되어 있는 멀티뷰 카메라 [14]를 사용하여 원거리에 대해 신뢰할 만한 결과를 얻기 위한 방법을 제안한다. 그림 1은 이를 위한 카메라 파라미터 획득 과정을 나타낸다.

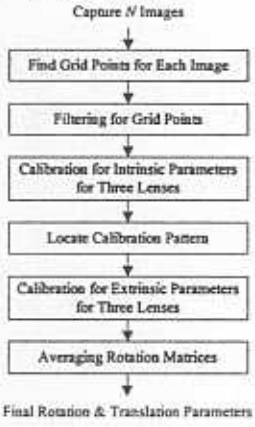


그림 1. 원거리를 위한 카메라 파라미터 획득 과정

내부 파라미터를 얻기 위해서는 패턴을 여러 위치에 두고 획득된 L장의 영상이 요구된다. 그러나 패턴이나 카메라의 정밀도 등으로 인한 영상 획득시의 오차가 발생하는데 이를 보완하기 위해 패턴을 동일한 위치에 두고 여러 장의 영상을 획득한다.

즉, 그림 2에 보인 것과 같이 획득된 각 영상을 $I_0, I_2, I_3, \ldots, I_{N-1}$ (N: 획득된 영상의 수) 이라 하고 각 영상으로부터 찾아진 격자점을 P_{ij} ($0 < i < N-1, 0 < j < K-1$, N: 획득된 영상의 수, K: 격자의 수) 라 하면 최종적인 격자점의 위치는 다음과 같다.

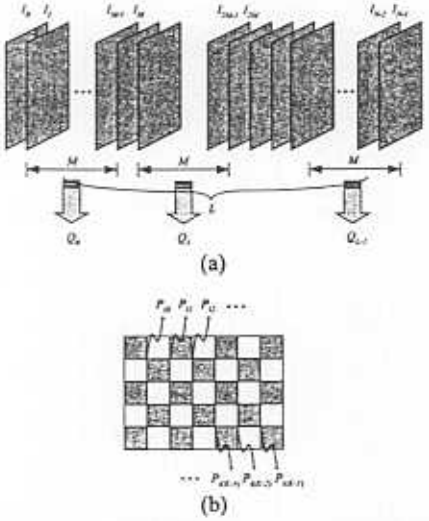
$$Q_{ij} = \frac{1}{M}\sum_{k=iM}^{(i+1)M-1} P_{kj} \tag{2}$$

위 식에서, M은 패턴을 동일한 위치에 두고 획득된 영상의 수, L은 카메라 파라미터를 얻기 위해 필요한 영상의 수로서 $N=M\times L$의 조건을 만족한다. Q_{ij}는 필터링 과정을 통해 얻은 최종적으로 캘리브레이션 과정을 수행하기 위해 필요한 격자점의 위치를 나타낸다. 비슷한 방법으로, Median 필터를 적용하여 격자점을 얻을 수도 있다.

내부 파라미터를 얻은 후, 패턴을 이미 알고 있는 위치에 두고 외부 파라미터를 얻는다. 멀티뷰 카메라의 세 렌즈는 동일한 평면상에 동일한 방향으로 위치하므로 이동 벡터는 렌즈간의 거리 만큼의 상이한 값, 회전 행렬은 정확히 동일한 값을 얻어야 하지만 오차로 인해 차이를 보인다. 따라서 각 렌즈의 카메라 파라미터를 구한 후 평균함으로써 원거리에 대해 좀더 보완된 외부 파라미터를 얻는다.

$$R_L = \frac{1}{3}\sum_{i=0}^{2} R_{Li} \qquad R_R = \frac{1}{3}\sum_{i=0}^{2} R_{Ri} \tag{3}$$

위 식에서, R_L, R_R은 각각 왼쪽과 오른쪽 카메라에 대한 회전 행렬, 인덱스는 위쪽, 왼쪽 그리고 오른쪽 렌즈를 나타낸다. 특히, 두 카메라 간의 회전 행렬은 일정한 관계를 갖기 때문에 우선 근거리에서 두 카메라 사이의 회전 행렬을 구하고 원거리에 대해서는 이동 벡터만 구함으로써 보다 정확한 파라미터를 얻을 수 있다.



(a)

(b)

그림 2. 카메라 캘리브레이션을 위한 격자점 위치 (a) 캘리브레이션을 위해 획득된 N 개의 영상 (b) 획득된 i 번째 영상의 패턴 격자점

이렇게 구한 외부 파라미터는 최종적으로 그림 3에 보인 것과 같이 world coordinate의 기준점으로부터 카메라까지의 회전 및 이동 행렬을 제공한다. 즉, 캘리브레이션 과정을 위한 입력으로 world coordinate에 대한 정보를 동시에 제공하면 world coordinate로부터 두 카메라까지의 회전 및 이동 행렬 $(R, T)_{right}$와 $(R, T)_{left}$를 각각 얻을 수 있다. 이렇게 얻어진 카메라의 외부 파라미터를 이용하여 실세계에 대한 3차원 point cloud를 가상 공간에 역 투영시킴으로써 실세계를 가상 환경으로 매핑 (mapping)하는 것이 가능하다. 또한, 두 대의 카메라를 사용하기 때문에 확장된 가상 환경 생성이 가능하다.

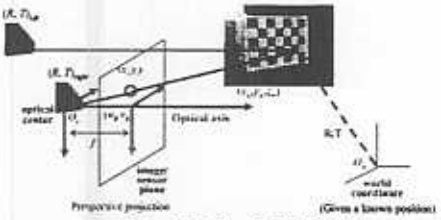


그림 3. 카메라 캘리브레이션

겹치는 부분에 대한 스티칭을 위해서는 우선 기준 카메라로부터 획득된 point cloud를 우선 역투영하고, 두 번째 카메라로부터 얻은 point cloud를 역투영 시 다음과 같은 방법을 사용한다.

그림 4와 같이 현재 역투영하려는 point cloud 점을 둘러싼 한 변이 l인 정육면체를 가정한다. 그러면 겹치는 부분에 대한 스티칭은 다음과 같다.

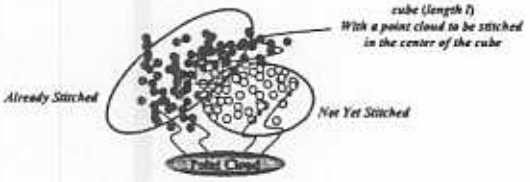


그림 4. 깊이 기반 스티칭

$$I_{LR_i} = Median_{j\in\eta_i}(I_{L_j}, I_{R_j}) \tag{4}$$

위 식에서, I_{LR_i}는 양쪽 카메라로부터 획득된 point cloud상의 한 점 i에 대해 수행한 스티칭 결과를 의미한다. 그리고 η_i는 i를 중심으로 한 정육면체 영역을 의미하므로, I_{L_j}과 I_{R_j}는 정육면체 내부에서의 왼쪽 및 오른쪽 카메라로부터 획득된 point cloud를 나타낸다. 즉, 각 카메라로부터 획득된 영상의 서로 겹치는 부분에 대해서는 각 point cloud에 대해 차례로 Median 필터를 적용하여 잡음 (noise)을 제거하는 동시에 깊이 정보의 변화를 유지하도록 하였다. 그리고 스티칭 결과로 얻은 point cloud의 점의 색깔을 i번째 점의 색깔로 사용하였다.

2.3 객체 분할 및 다중 층 생성

실시간 동적 객체를 비디오 시퀀스로부터 추출해 내는 작업은 비전 응용 분야의 기본적이면서도 중요한 문제이다. 제안된 방법은 일반적인 배경으로부터 임의의 객체를 여러 가지 특징들을 결합하여

분리하며, (1) 정적 배경 모델링 (2) 그림자 분리 (3) 움직이는 객체 분할의 세 단계로 나뉜다 [11].

그림 5는 스티칭 과정을 통해 두 카메라에서 획득된 3D 좌표를 일치시키고 배경 영상을 얻은 후, 3D 정보를 갖는 객체를 삽입하여 합성하는 과정을 나타낸다. 우선, 멀티뷰 카메라로부터 획득된 원영상과 대응되는 깊이 맵을 얻는다. 그리고 삽입된 각각의 객체에 대해 객체 분할을 거친 후, 서로 다른 층에 저장한다. 마지막으로, 이를 합성하여 최종적인 가상 환경에 대한 영상 및 깊이 맵을 생성한다.

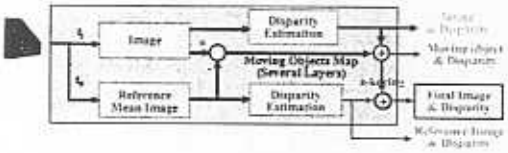


그림 5. 삼차원 정보 합성

객체를 배경과 함께 처리하지 않고 영상 분할 후 여러 층으로 나누어 각각 처리함으로써 하나의 카메라를 사용하더라도 객체들을 서로 다른 3D 공간상에 위치시킬 수 있다. 즉, 하나의 카메라에서 획득된 영상에서는 두 객체가 겹쳐 촬영되더라도 3D 가상 공간상에서는 서로 다른 층에 저장되기 때문에 서로 간섭 없이 가상 공간에 디스플레이되는 것을 확인할 수 있다. 또한 영상 분할 과정을 수행하기 때문에 배경보다 앞쪽에 위치하는 객체들 때문에 발생하는 폐색 문제를 해결하고 경계 영역에 대한 정확한 양안차 맵을 얻을 수 있다. 이와 같은 방법을 사용하여 가상 공간을 생성한 후, 사용자는 실시간으로 그 공간을 네비게이션할 수 있다 [15].

2.4 파노라믹 영상에 기반한 가상 환경 생성

두 대의 카메라를 턴테이블을 이용해 회전시킴으로써 일정한 반경 내의 모든 방향에서 바라볼 수 있는 가상 환경을 생성할 수 있다. 그림 6은 이에 대한 개념도를 나타낸다. 가상 환경을 생성하기 위해 다음과 같은 과정을 수행한다. 카메라를 가능한 근접한 거리에 위치시키고 이로부터 각 카메라의 회전 행렬을 구한다. 회전 행렬을 얻은 후에 카메라 간의 간격은 적절히 조절된다. 이때, bar 위의 카메라는 수평으로만 이동하므로 이동 벡터에만 영향을 미치며, 카메라간의 상대적인 회전 행렬은 변화하지 않는다.

양쪽의 최종적인 외부 파라미터를 $(R, T)_R$ 와 $(R, T)_L$ 이라 하면, 두 카메라의 상대적인 회전 벡터는 다음과 같이 표현할 수 있다.

$$\begin{aligned} R_{Yaw} &= R_{R,Yaw} - R_{L,Yaw} \\ R_{Tilt} &= R_{R,Tilt} - R_{L,Tilt} \\ R_{Roll} &= R_{R,Roll} - R_{L,Roll} \end{aligned} \tag{5}$$

두 카메라간의 상대적인 회전 행렬을 얻은 후, 패턴을 이미 알고 있는 곳에 위치시키고, 그 점으로부터 기준 카메라의 이동 벡터 및 회전 행렬을 구한다. 두 카메라 간의 실제 거리와 상대적인 회전 행렬을 알고 있으므로 가상 환경 상에 두 카메라를 위치시키는 것이 가능하다.

마지막으로, 그림 6에 보이는 것과 같은 턴테이블을 일정한 각도 만큼씩 회전시키고 각각의 정지된 위치에서 양쪽 카메라로부터 3D 정보를 획득한다. 3D 정보는 그림과 같이 가상 환경상에 point cloud 형태로 디스플레이되며 이를 렌더링하여 가상 공간에 역투영된 결과를 얻을 수 있다. 턴테이블을 360° 회전시켜 모든 방향에 대해 point cloud 를 얻을 수 있으며 이를 3D 깊이 기반 스티칭 과정을 통해 최종적인 3D 가상 환경을 생성할 수 있다.

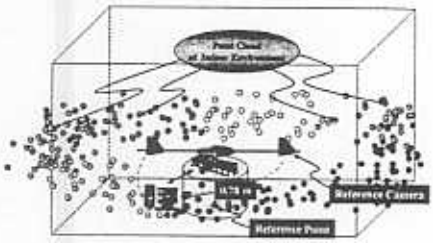


그림 6. 가상 환경 생성

그림 7은 중앙에 한 대의 카메라를 사용하는 방법과 비교해 사용자가 어느 정도의 확장된 영역을 바라볼 수 있느냐의 성능 평가를 위한 것이다. 그림 7은 가로 $2W$, 세로 $2H$ 인 일반적인 실내 환경을 나타내고, 그 내부에 멀티뷰 카메라를 장착한 턴테이블을 위치시킨 것을 위에서 바라본 것이다. 위쪽에 위치하는 원은 사람을 모델링한 것이다. 따라서, 중심에 위치한 카메라가 원을 바라보았을 때 보이는 최대 원 둘레와 제안된 방법을 통해 보이는 최대 원 둘레를 비교함으로써 제안된 방법을 사용한 사용자의 관찰 이득을 계산 할 수 있다.

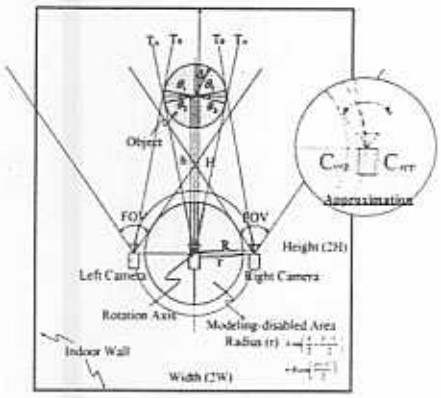


그림 7. 사용자 네비게이션 영역의 이득

우선, 중앙에 카메라 한 대를 사용하는 경우, 사용자가 회전축의 중심에서 모델링된 원통을 바라볼 때, 두 접선이 이루는 최대 각도는 $\pi - 2\theta_2$ 이다. 여기서, θ_2 는 법선과 이루는 각도를 의미한다. 반지름

이 d인 경우 원 둘레의 길이 A는 다음과 같다.

$$A = d\left(\pi - 2\sin^{-1}\left(\frac{d}{h}\right)\right) \quad (6)$$

위 식에서, d는 객체의 반지름, h는 턴테이블의 중심으로부터 객체 중심까지의 거리, r은 카메라가 관영하지 못하는 공간의 반지름으로 다음과 같다.

$$r = R\cos\left(\frac{FOV}{2}\right) \quad (7)$$

위 식에서, R은 턴테이블의 회전 반경, FOV는 멀티 뷰 카메라의 시야를 나타낸다.

반면, 제안한 방법을 사용하는 경우, 사용자가 관측 가능한 객체의 최대 각도는 $\pi + 2\theta_i$이다. 여기서, θ_i은 각각의 카메라에서 원통에 접선을 그었을 때 법선과 이루는 각도이다. 그러나 그림 7의 오른쪽에 보인 것과 같이 원 카메라의 위치 C_{org}와 반지름을 R로 하는 원과의 접선에 의한 카메라 위치 C_{app}와의 차이는 배경이 충분히 회전축의 중심으로부터 멀리 위치하므로 무시할 만하다. 따라서, 근사화하면 최대 원 둘레의 길이 B는 다음과 같다.

$$B = d\left(\pi + 2\sin^{-1}\left(\frac{R-d}{h}\right)\right) \quad (8)$$

따라서, 제안된 방법을 통해 사용자가 관측 가능한 범위에 대한 이득 C는 다음과 같으며, 턴테이블의 회전 반경 R의 함수로 표현된다.

$$C = \left\{\pi + 2\sin^{-1}\left(\frac{R-d}{h}\right)\right\} / \left\{\pi - 2\sin^{-1}\left(\frac{d}{h}\right)\right\} \quad (9)$$

따라서 주어진 h와 d에 대해 R이 증가함에 따라 이득 C는 거의 선형적으로 증가한다.

3. 실험 및 분석

본 실험은 일반적인 실내 환경의 정상적인 조명하에서 수행되었다. 영상 획득을 위해 IEEE 1394 카메라 (Digiclops)를 사용하였으며, OpenCV 라이브러리를 이용하여 카메라 캘리브레이션을 수행하였다. 본 실험은 Xeon 1.0GHz CPU 컴퓨터에서 수행되었다.

모델링 하려는 공간이 커질수록 패턴의 크기도 커지는데 본 실험에서의 실내공간 크기는 7m×5m이므로 1.75m×1.25m 크기의 패턴이 사용되었다.

표 1. 거리에 따른 외부 파라미터의 오차 범위

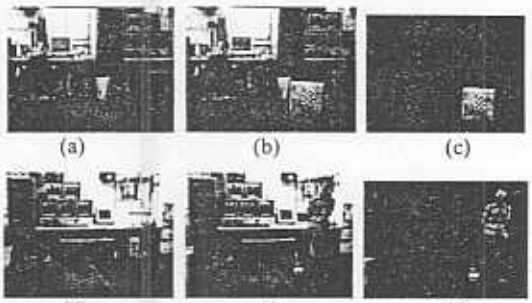
영상 크기	640×480	640×480
거리	≈1m	≈4m
패턴 크기	0.3m×0.2m	1.75m×1.25m
오차 범위	±0.01m 이내	±0.04m 이내

그림 8은 깊이 기반 스티칭 결과를 나타낸다. 그림 8(a)와 그림 8(b)는 왼쪽과 오른쪽 카메라로부터 입력된 영상의 point cloud를 캘리브레이션 과정 수행 후 가상 공간에 역투영한 결과이며, 그림 8(c)는 스티칭을 통해 결합된 영상을 나타낸다.

(a) (b) (c)

그림 8. 깊이 기반 3D 스티칭 (a) 왼쪽 영상 (b) 오른쪽 영상 (c) 결합된 영상

그림 9는 배경에 비해 앞에 위치하는 객체들로 인해 발생하는 폐색 문제를 해결하고 그 경계 영역에서의 정확한 양안차 맵을 얻기 위해 여러 층으로 나누어 가상 환경을 구성한 결과이다.

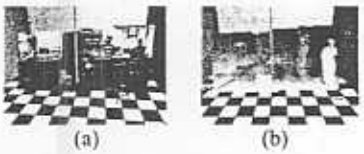


(a) (b) (c)

(d) (e) (f)

그림 9. 경지 및 동적 객체 분할 (a) 왼쪽 원영상 (b) 객체가 삽입된 왼쪽 영상 (c) 삽입된 객체가 분할된 왼쪽 영상 (d) 오른쪽 원영상 (e) 객체가 삽입된 오른쪽 영상 (f) 삽입된 객체가 분할된 오른쪽 영상

그림 10은 구성된 가상 환경 내에서 사용자가 자신의 분신은 그대로 남겨둔 채로 가상 공간을 네비게이션하는 영상의 한 장면을 보여준다. 그림 11은 파노라믹 영상을 사용하여 가상 환경 생성 후, 동일한 장면에 대해 줌 인과 줌 아웃을 실행한 결과를 보인 것이다.



(a) (b)

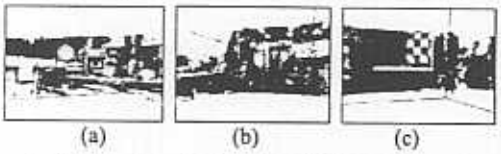
그림 10. 가상 공간 네비게이션 (a) 사용자의 가상 공간 네비게이션 (b) 대응되는 양안차 맵

(a) (b)

그림 11. 줌 인과 줌 아웃

그림 12는 가상 환경 구성 후에 사용자가 네비게이션 가능한 실내의 여러 방향에 대한 장면을 보인 것이며, 그림 13은 회전축을 중심으로 위에서 내

려다 본 실내의 가상 환경을 나타낸다. 전체적으로 사각형 모양을 이루는 것을 알 수 있으며 오른쪽 부분에 튀어나온 부분은 그림 12(b)를 나타낸다.



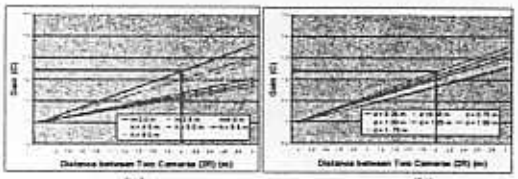
(a) (b) (c)

그림 12. 가상 환경을 여러 방향에서 본 장면

그림 13. 가상 환경을 위에서 본 장면

한 대의 PC 를 기반으로 구현된 본 시스템은 깊이 정보 계산 및 처리, 다중 층에 기반한 가상 공간 구성 등을 초당 2-3 프레임 정도로 처리한다. 즉, 최종적인 시스템에서는 오프 라인으로 가상 환경이 미리 구성되고, 단지 사용자 및 아바타만 실시간으로 움직이기 때문에 시스템의 동작 속도는 사용자의 아바타를 만드는데 소요되는 시간에 의존한다.

그림 14는 두 카메라 사이의 거리에 따른 사용자 관찰 이득 C (수식(9) 참조)을 나타낸다. 예를 들어, 반지름 0.5m 인 원통형 물체를 회전 축으로부터 3m 거리에 두고 2m 간격의 두 카메라로 촬영하는 경우 24%만큼의 이득을 얻는다. 또한 반지름 1.75m 인 물체를 4m 거리에 두고 두 카메라로 촬영하는 경우 비슷하게 24%만큼의 이득을 얻을 수 있다.



(a) (b)

그림 14. 카메라간의 거리에 따른 사용자의 관찰 이득 (a) d=0.5m 인 경우 h 에 따른 관찰 이득 (b) h=4m 인 경우 d 에 따른 관찰 이득

4. 결론 및 추후 과제

본 논문에서는 다수의 멀티뷰 카메라를 사용하여 영상의 3D 깊이 정보에 기반한 파노라믹 영상을 생성하고 이를 통하여 가상 환경을 구축하는 알고리즘을 제안하였다. 제안된 방법은 사용자에게 확장된 시야를 제공하여 생성된 가상 환경의 임정한 반경 내에서의 자유로운 네비게이션을 가능하게 한다. 배경보다 앞쪽에 위치하는 객체들로 인한 폐색 문제를 해결하고 그 경계 영역에 대한 정확한 양안차 맵을 구하기 위해 여러 층을 사용하여 가상 공간을 구성하였다. 이와 같은 방법을 사용함으로써 사용자는 가상 환경 내에 존재하는 여러 가상 객체들뿐만 아니라 깊이 정보에 기반하여 아바타를 통해 자연스런 네비게이션이 가능하다. 따라서, 제안된 방법은 기존의 컴퓨터 그래픽스 기반의 가상 환경이 갖는 실시간 렌더링 문제를 완화시킬 뿐만 아니라 사용자에게 보다 큰 사실감 및 몰입감을 제공할 것으로 보인다. 제안한 방법을 통해 생성된 가상 환경 내에서는 객체가 카메라로부터 가까울수록, 객체의 반경이 커질수록 턴테이블의 회전 반경에 따른 사용자의 객체에 대한 관찰 이득이 증가함을 알 수 있다. 이를 360° 전 방향으로 일반화 시키면 더 큰 이득을 보일 것을 예상할 수 있다.

추후에는 생성된 가상 환경 내에 새로운 가상 객체를 효과적으로 삽입하는 방법 그리고 이를 위한 가상 환경 내의 조명 조건의 분석에 관한 연구를 진행할 예정이다.

5. 참고문헌

[1] QuickTime VR, http://www.quicktime/qtvr

[2] H. Chen, "Building Panoramas from Photographs Taken with a Hand-held Camera," Ph.D. Dissertation, University of Hong Kong, 2002.

[3] S. K. Nayar, "Catadioptric Omnidirectional Camera," *IEEE Computer Society Conf. on CVPR'97*, pp. 482-488, 1997.

[4] S. E. Chen, "QuickTimer VR – An Image-Based Approach to Virtual Environment Navigation," *SIGGRAPH'95*, pp. 29-38, 1995.

[5] H. Y. Shum and R. Szeliski, "Construction of Panoramic Image Mosaics with Global and Local Alignment," *International Journal of Computer Vision*, vol. 36(2), pp. 101-130, 2000.

[6] B. Rousso, S. Peleg and I. Finci, "Generalized Panoramic Mosaics," *Proc. DARPA Image Understanding Workshop'97*, pp. 255-260, 1997.

[7] S. Peleg, B. Rousso, A. Rav-Acha and A. Zomet, "Mosaicing on adaptive manifolds," *IEEE Trans. on PAMI*, vol. 22(10), pp. 1144-1154, 2000.

[8] S. Peleg et. al. "Cameras for stereo panoramic imaging," *CVPR*, vol. 1, pp. 208-214, 2000.

[9] H. Y. Shum and R. Szeliski, "Stereo reconstruction from multiperspective panoramas," *ICCV*, vol. 1, pp. 14-21, 1999.

[10] R. Benosman and J. Devars, "Panoramic stereovision sensor," *Proc. 14th Intl. Conf. on Pattern Recognition*, pp. 767-769, 1998.

[11] S. Kim and W. Woo, "3D Movement Tracking with Asynchronous Multi-cameras for Interactive Systems", *Proc. of SPIE. PW-EI-VCIP'02*, vol. 4671, pp. 502-512, Jan. 2002.

[12] W. Woo, N. Kim and Y. Iwadate, "Stereo imaging using a camera with stereoscopic adapter," *Proc. of IEEE - SMC 2000*, vol.2, pp. 1512-1517, Oct. 2000.

[13] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proc. of the Seventh IEEE Int'l Conf.*, vol. 1, pp. 666-673, 1999H.

[14] Point Grey Research Inc., http://www.ptgrey.com

[15] S. Kim and W. Woo, "Image-based 3D Mosaicking using Multiple 3D Cameras," *KICS2002*, vol. 26, pp. 58-61, 2002.