

영상 기반 가상 환경 생성을 위한 깊이 기반 정합*

김 세 환^o, 김 기 영, 우 운 택

광주과학기술원 U-VR 연구실

Depth-based Registration for Image-based Virtual Environment Generation

Schwan Kim^o, Kiyong Kim and Woontack Woo

KJIST U-VR Lab.

(skim, kkim, wwoo)@kjist.ac.kr

요 약

본 논문에서는 영상 기반 가상 환경을 생성하기 위해 다수의 멀티뷰 카메라로부터 획득된 영상의 3D 깊이 정보에 기반한 정합 방법을 제안한다. 일반적으로, 정밀한 광학 센서를 사용하는 경우에는 오차가 작은 3D 깊이 정보를 제공하기 때문에 정합과정이 비교적 용이하다. 그러나 이러한 방법은 고가의 장비를 필요로 하며 측정과 모델링에 상당한 시간을 요한다. 반면, 본 논문에서 제안한 방법은 상대적으로 3D 깊이 정보의 오차가 크지만 저가의 멀티뷰 카메라를 사용하며 제작이 비교적 용이하다. 이를 위해 두 대의 멀티뷰 카메라를 캘리브레이션하고, 획득된 각 영상의 3D point cloud에 대한 깊이 기반 정합 및 모델링 과정을 수행한다. 그리고, 가상 환경에 증강될 객체는 턴테이블을 회전하여 여러 방향에서 획득한 3D point cloud를 이용하여 정합한다. 3D 깊이 정보의 오차가 상대적으로 큰 경우에도 제안된 방법을 통해 물체의 3D point cloud가 효과적으로 정합될 수 있음을 실험 결과를 통해 알 수 있다.

서 론

실세계 환경에 대한 모델링은 로봇 네비게이션, 건축 모델링, 비행기 시뮬레이션 등의 다양한 가상 현실 응용 분야에서 중요한 역할을 한다. 환경의 사실적 모델 생성을 위해서는 광학 센서를 통해 3D 데이터를 획득하고 이에 대한 정합 과정을 통해 3D 모델을 생성한다. 최근, 이러한 실세계 공간에 대한 사실적 모델링을 이용한 영상 기반 가상 현실 시스템 (IBVR: Image-based Virtual Reality)은 컴퓨터 비전 뿐만 아니라 컴퓨터 그래픽스 영역에서도 많은 관심의 대상이 되고 있다. 이는 IBVR이 실사를 사용함으로써 사실감 제공 뿐만 아니라 렌더링 과정도 간단하다는 장점이 있기 때문이다.

그 동안 물체 모델링을 위한 다양한 정합 방법들이 제안되었다. Besl 등에 의해 제안된 ICP (Iterative Closest Point) 알고리즘은 현재까지도 널리 사용되고 있는 방법이다 [1]. Johnson 등은 색 정보를 결합한 Color ICP를 제안하고 이를 통해 실내 환경을 복원하였다 [2]. 그리고 Blais 등은 모든 뷰에서의 대응되는 부분 사이의 거리에 기반한 비용 함수를 최소화하기 위해 simulated annealing을 사용하였다 [3]. 또한 Levoy 등은 다양한 종류의 상에 대해 3D 스캐너를 통해 얻은 데이터를 복셀에 기반한

방법을 통해 정합 과정을 수행하였으며, Kanade 등은 영상에 기반한 움직이는 객체의 3D 복원을 위해 비슷한 방법을 적용하였다 [4][5]. 반면, Nishino 등은 여러 장의 레인지 영상을 위한 강건한 정합을 구현하기 위해 M-estimator를 이용한 최적화 방법을 제안하였다 [6]. 그러나 대부분의 방법은 고가의 정밀한 장비에 의존하며 3D 모델을 생성하는데 상당한 시간을 요한다는 한계를 지닌다. 또한 3D 깊이 정보가 큰 오차를 포함하는 경우 효과적인 정합이 어렵다는 단점이 있다.

본 논문에서는 영상 기반 가상 환경 생성을 위해 다수의 멀티뷰 카메라로부터 획득된 3D 깊이 정보에 기반한 정합 방법을 제안한다. 우선, 두 대의 멀티뷰 카메라를 캘리브레이션하고 각 카메라로부터 획득된 3D point cloud를 가상 환경에 역투영한다. 그리고 멀티뷰 카메라를 턴테이블을 사용해 회전시킴으로써 360° 전방향에 대해 얻어진 3D point cloud를 ICP를 이용한 정합과정을 통해 실내 환경을 모델링한다. 또한 턴테이블을 일정한 범위 내에서 회전시켜 여러 방향에서 획득된 3D point cloud를 정합 및 모델링 과정을 수행함으로써 여러 방향에서 관찰이 가능한 2.5D 객체를 생성하고 가상 환경 내에 증강한다. 이렇게 생성된 가상 환경에서 사용자는 트래커가 부착된 스테레오스코프 HMD를 통해 일정한

본 연구는 광주과학기술원 실감방송 연구센터를 통한 정보통신부 ITRC 사업과 ETRI의 지원에 의한 것임

반경 내에서 몰입감을 가지고 네비게이션 할 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 가상 환경 생성을 위한 ICP, 3 장에서는 가상 객체를 위한 깊이 기반 정합에 대해 설명한다. 그리고 4 장에서는 실험 및 분석에 대해 기술한 뒤, 5 장에서 결론 및 후속 과제에 대해 언급한다.

2. 가상 환경 생성을 위한 ICP

Tsai의 캘리브레이션 알고리즘을 사용하여 그림 1에 보인 것과 같이 실세계의 기준점으로부터 두 카메라까지의 회전 및 이동 행렬 (R_L, T_L) 과 (R_R, T_R) 을 각각 얻을 수 있다 [7]. 그림에서 각 카메라는 가로, 세로 방향으로 여러 개의 렌즈를 갖는 일반화 된 멀티뷰 카메라를 나타낸다.

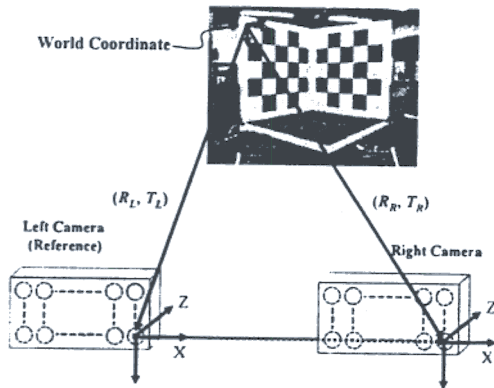


그림 1. 인터 카메라 캘리브레이션

그러나 캘리브레이션의 부정확성 때문에 각 카메라에 대해 캘리브레이션 패턴의 N_G 개 격자점의 역투영된 3D 좌표는 불일치한다. 제안된 인터 카메라 캘리브레이션은 최적화 과정을 통해 대응되는 격자점 사이의 거리를 최소화하는 (R_L, T_L) 과 (R_R, T_R) 을 구하는 과정으로, 다음과 같이 표현된다 [8][9].

$$\begin{aligned} & \text{Given two sets of corresponding points.} \\ & \text{Find } \{R_L, T_L\} \text{ \& } \{R_R, T_R\} \\ & \text{such that } \operatorname{argmin}_{\{R_L, T_L\} \text{ \& } \{R_R, T_R\}} \sum_{i=0}^{N_G-1} (G_{L,i} - G_{R,i})^2 \end{aligned} \quad (1)$$

위 식에서, G_L 와 G_R 는 왼쪽과 오른쪽 카메라로부터 획득된 캘리브레이션 패턴의 3D 격자점이 가상 환경으로 역투영된 것을 나타낸다. 이상적인 경우, G_L 와 G_R 는 정확히 일치해야 하지만 실제로는 불가능하다. 그 차이는 $\{R_L, T_L\}$ 및 $\{R_R, T_R\}$ 의 함수로 표현되며, 이 차이를 최소화하는 각 카메라의 외부 파라미터를 구함으로써 두 카메라 사이의 정확한 위치 관계를 얻을 수 있다. 본 논문에서는 최적화 방법으로 ICP 알고리즘을 이용한 방법을 제안한다 [1].

우선, 가상 공간에 역투영된 3D 격자점 G_L 와 G_R , ($i=0, 1, \dots, N_G-1$) 그리고 다음과 같은 초기값들이 주어진다.

$$G_{R,0} = G_R, \quad \bar{q}_0 = [1000000]^T, \quad k=0 \quad (2)$$

위 식에서, $G_{R,0}$ 는 오른쪽 카메라로부터 획득된 격자점의 3D point cloud 집합의 초기값, \bar{q}_0 은 쿼터니온의 초기값 그리고 k 는 인덱스를 나타낸다. ICP 알고리즘에서의 쿼터니온은 다음과 같이 회전과 이동벡터의 결합으로 구성된다.

$$\bar{q} = [\bar{q}_R | \bar{q}_T], \quad \bar{q}_R = [q_0 \ q_1 \ q_2 \ q_3]^T, \quad \bar{q}_T = [q_4 \ q_5 \ q_6]^T \quad (3)$$

첫 번째 과정으로 오른쪽 카메라의 데이터 집합 G_R 의 각 점들 $G_{R,i}$ 로부터 왼쪽 카메라의 데이터 집합 G_L 까지의 최소 거리가 되는 점 $G_{L,j}$ 를 찾는다.

$$Y_k = C(G_{R,k}, G_L) \quad (4)$$

위 식에서, C 는 최소 거리 점 연산자, Y 는 C 를 주어진 G_L 에 대해 G_R 에 적용했을 때의 최소 거리가 되는 점들의 집합을 나타낸다.

두 번째, 최소 자승의 관점에서 $(G_{R,i}, G_{L,j})$ 쌍들의 거리를 최소화하는 강체 변환을 구한다.

$$(\bar{q}_k, d_k) = Q(G_{R,i}, Y_k) \quad (5)$$

위 식에서, Q 는 최소 자승 쿼터니온 연산자, d_k 는 평균 자승 점 매칭 오차를 나타낸다. 이는 식 (6)과 같은 mean square objective function을 최소화하는 쿼터니온을 구하는 과정이다.

$$f(\bar{q}) = \frac{1}{N_G} \sum_{i=0}^{N_G-1} \|\bar{y}_i - R(\bar{q}_R)G_{R,i} - \bar{q}_T\|^2 \quad (6)$$

세 번째, 구해진 쿼터니온을 G_R 에 적용하여 다음과 같이 N_G 개의 격자점을 변환한다.

$$G_{R,k+1} = \bar{q}_k(G_{R,0}) \quad (7)$$

마지막으로, 최소 거리 점들간의 거리가 문턱치 (τ) 보다 작아질 때까지 위의 과정을 반복한다.

$$d_{k+1} - d_k < \tau \quad (8)$$

이와 같이 각 멀티뷰 카메라로부터 획득된 N_G 개의 격자점에 대한 역투영된 3D 좌표에 최적화 과정을 적용하면 이로부터 역으로 기준이 되는 카메라에 대한 상대적인 카메라의 최적화된 회전 및 이동 행렬을 식 (3)과 식 (9)를 통해 얻을 수 있다.

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_1^2 - q_2^2 - q_3^2 \end{bmatrix} \quad (9)$$

이를 통해 기준 카메라를 중심으로 상대 카메라의 위치를 변환한다. 기준 카메라에 대한 쿼터니온을 다시 획득하고 동일한 방법을 계속 적용하여 오차가 정의된 문턱치보다 작아질때까지 반복적으로 적용하여 최종적인 두 멀티뷰 카메라의 위치를 찾

아낸다. 이와 같은 방법을 통해 카메라 파라미터를 보정함으로써 두 카메라로부터 가상 환경에 역투영된 3D point cloud를 정합하는 것이 가능하다.

이와 같이 턴테이블 양쪽에 부착된 두 대의 멀티뷰 카메라에 인터 카메라 캘리브레이션을 수행한 후, 실내 환경을 모델링하기 위해 일정한 각도로 회전하면서 각 위치에서의 깊이 정보를 획득한다. 이를 회전각 정보만을 이용하여 3D 공간에서 디스플레이하면 방의 구조인 사각형 모양을 형성한다. 그러나 회전각 정보만을 사용했기 때문에 정확한 정합 과정이 요구된다. 본 논문에서는 이를 위해 ICP 알고리즘 및 미디언 필터를 사용한다.

그러나 멀티뷰 카메라를 사용하는 경우, 창문이나 벽, 문 등은 질감이 없이 균질한 (homogeneous) 면을 이루기 때문에 양안차 정보 획득이 불가능하거나 획득하더라도 오차가 크기 때문에 값 자체가 무의미하다. 따라서, 전처리 과정으로 유효한 양안차 정보를 기반으로 양안차가 획득되지 않은 부분을 bilinear interpolation 과정을 통해 보완하는 과정을 수행한다.

일반적으로 ICP 알고리즘을 사용하기 위해서는 정합하려는 두 3D point cloud 집합의 초기 위치가 적절해야 한다. 가상 환경 생성과정에서는 턴테이블을 일정한 각도만큼 회전해서 3D point cloud를 획득하기 때문에 정합하려는 인접 영상의 3D point cloud는 ICP 알고리즘을 적용하기에 적합한 초기값을 갖는다. 또한, ICP 알고리즘은 두 영상의 겹치는 부분에 대해서만 수행하게 되는데, 이는 턴테이블의 회전각을 고려하여 얻을 수 있다. 마지막으로, 겹치는 부분에서의 색 선택을 위해 미디언 필터를 적용하였다.

3. 가상 객체를 위한 깊이 기반 정합

3.1 회전하는 카메라를 위한 인터 카메라 캘리브레이션

여러 방향에서 관찰이 가능한 2.5D 가상 객체 생성을 위해 인터 카메라 캘리브레이션을 회전하는 두 대의 멀티뷰 카메라에 적용한다. 이렇게 함으로써 각 카메라의 위치에서 카메라 파라미터를 보정하고 다수의 3D point cloud를 정합할 수 있다. 그림 2는 회전하는 두 대의 카메라에 대한 인터 카메라 캘리브레이션 방법을 보인 것이다. 기준이 되는 i 번 째 위치에서 획득한 왼쪽 및 오른쪽 카메라의 외부 파라미터를 (R_{L_i}, T_{L_i}) 와 (R_{R_i}, T_{R_i}) 라 하자. 그러면 기준 위치보다 회전각 α 이전에서의 두 카메라에 대한 파라미터는 $(R_{L(i-1)}, T_{L(i-1)})$ 와 $(R_{R(i-1)}, T_{R(i-1)})$ 이 된다. 그리고 회전각 α 이후에서의 두 카메라의 파라미터는 $(R_{L(i+1)}, T_{L(i+1)})$ 와 $(R_{R(i+1)}, T_{R(i+1)})$ 이 된다.

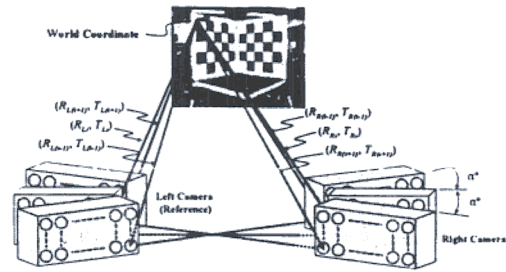


그림 2. 회전하는 다중 카메라 캘리브레이션

우선, 기준 위치에서 양쪽에 위치한 멀티뷰 카메라에 대해 Tsai 알고리즘을 적용한다. 그리고, (R_{R_i}, T_{R_i}) 와 (R_{L_i}, T_{L_i}) 를 사용하여 역투영된 N_G 개의 특징점에 대한 3D point cloud를 기반으로 인터 카메라 캘리브레이션 과정을 수행한다. 턴테이블을 일정한 각도로 회전시키면서 기준 카메라를 중심으로 인터 카메라 캘리브레이션 과정을 연속적으로 수행한다. 최종 결과로, 기준 카메라를 중심으로 보정된 N_c 개의 카메라의 위치에서, 각 멀티뷰 카메라에 대한 보정된 카메라 파라미터를 얻을 수 있다.

3.2 선 기반 정합

본 절에서는 실내 환경에 존재하는 가구들의 기본적인 구조인 선과 평면으로 표현될 수 있는 문, 벽, 캐비닛 등과 같은 물체에 대한 깊이 기반 정합을 다룬다. 보다 정확하게 설명하자면, 두 대의 멀티뷰 카메라를 회전하면서 촬영한 물체가기 때문에 단일 방향에서 획득된 2.5D 모델보다는 넓은 범위에서 획득된 물체의 3D point cloud에 대한 정합이다. 우선, 물체를 구성하는 기본 구조 중의 하나인 선의 특징을 이용한 정합 방법을 그림 3에 보였다.

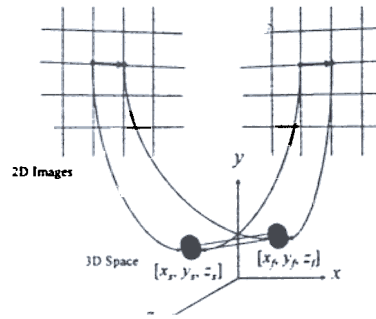


그림 3. 선 기반 정합을 위한 개념도

그림 3에서 정합하는데 사용되는 2D 영상은 정합하고자 하는 물체가 보이는 범위 내에서 턴테이블을 회전시켜 N 개 만큼 얻는다. 그리고 대응되는 양안차 맵 또한 N 개 만큼 획득된다. N 개의 양안차 맵으로부터 획득되는 3D point cloud 값들은 3D 공간 상에서 거의 비슷한 값을 갖는다. 그러나 양안차로부터 얻어진 3D point cloud의 깊이 정보는 오차를

포함하기 때문에 N 개의 영상에서의 격자점에 대한 최종적인 3D 위치 좌표는 다음과 같이 표현된다.

$$\begin{aligned} x_i' &= \text{Median}(x_i) & y_i' &= \text{Median}(y_i) & z_i' &= \text{Median}(z_i), \\ x_f' &= \text{Median}(x_f) & y_f' &= \text{Median}(y_f) & z_f' &= \text{Median}(z_f) \end{aligned} \quad (10)$$

$i = 0, 1, 2, \dots, N-1$

위 식에서, (x_i', y_i', z_i') 는 시작점, (x_f', y_f', z_f') 는 도착점, Median 은 미디언 필터링 과정을 나타낸다. 이러한 과정을 통해 여러 장의 영상으로부터 얻어지는 각 점에 대해 안정화된 3D 좌표 값을 얻을 수 있다. 이를 기반으로 선 기반 정합을 수행한다.

2D 영상에서의 점 사이의 거리는 객체를 바라보는 카메라의 위치 및 방향에 따라 영상마다 서로 다르다. 그러나, 선분을 이루는 두 점 사이를 균등하게 M 등분 했을 때, 그 사이에 존재하는 대응되는 3D 좌표 값은 동일한 3D 공간상에 존재한다. 그러나 실제의 경우는 약간차 추정 결과의 부정확성 때문에 3D 값들이 동일하지 않다. 따라서, 선 기반 정합에서는 N 장의 영상을 사용하는 경우 N 개의 3D 좌표 값 가운데 두 점을 잇는 선분에 가장 근접한 3D 좌표를 최종 값으로 선택한다. 그림 4는 그 예를 보인 것으로, 두 점을 잇는 직선을 사용하여 3D 좌표를 선택하는 과정을 보여준다.

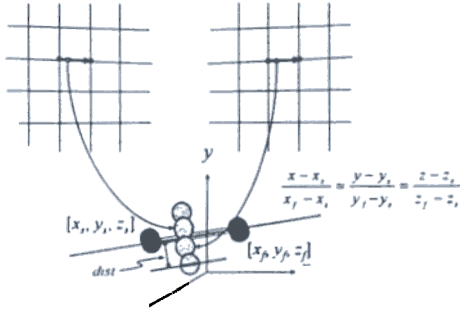


그림 4. 선 기반 정합의 예

우선, 두 점에 대한 3D 좌표를 각각 (x_1, y_1, z_1) , (x_2, y_2, z_2) 라 하면, 3D 공간 상에서 두 점을 지나는 직선 \vec{l} 은 다음과 같이 표현된다.

$$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1} = \frac{z-z_1}{z_2-z_1} \quad (11)$$

이를 t 라 설정하면 직선 위의 임의의 3D 좌표 (H)는 다음과 같이 표현할 수 있다.

$$\begin{aligned} x &= (x_2 - x_1)t + x_1, \\ y &= (y_2 - y_1)t + y_1, \\ z &= (z_2 - z_1)t + z_1, \end{aligned} \quad (12)$$

이 때, 공간상의 임의의 점 $A(x_a, y_a, z_a)$ 로부터 이 직선에 수직으로 내린 수선 \overline{AH} 는 다음과 같다.

$$\overline{AH} = ((x_f - x_i)t - (x_a - x_i), (y_f - y_i)t - (y_a - y_i), (z_f - z_i)t - (z_a - z_i)) \quad (13)$$

수선은 직선과 수직을 이루므로 다음 조건을 통해 t 값을 구한다.

$$\overline{AH} \cdot \vec{l} = 0 \quad (14)$$

이를 정리하면, t 는 다음과 같이 표현된다.

$$t = \frac{(x_a - x_i)(x_f - x_i) + (y_a - y_i)(y_f - y_i) + (z_a - z_i)(z_f - z_i)}{(x_f - x_i)^2 + (y_f - y_i)^2 + (z_f - z_i)^2} \quad (15)$$

그러면 최종적으로 직선으로부터 임의의 3D 좌표까지의 거리 ($dist$)는 식 (16)과 같이 표현된다.

$$dist = \text{sqr} \left\{ (x_f - x_i)t - (x_a - x_i) \right\}^2 + \left\{ (y_f - y_i)t - (y_a - y_i) \right\}^2 + \left\{ (z_f - z_i)t - (z_a - z_i) \right\}^2 \quad (16)$$

따라서, N 개의 3D 좌표 값 가운데 두 점을 잇는 선분에 가장 근접한 3D 좌표는 $dist$ 를 최소화하는 값이므로 이를 통해 최종적인 값을 선택할 수 있다.

3.3 평면 기반 정합

평면 기반 정합을 위한 개념도를 그림 5에 보였 다. 평면 기반 정합은 주어진 네 개의 격자점에 대한 3D 좌표가 주어지고, 그 사이에 존재하는 점들이 평면 위에 존재한다는 조건 하에 수행된다. 선 기반 정합에서는 2D 영상에서의 두 점 사이를 M 등분 했을 때, 그 사이에 존재하는 대응되는 3D 좌표 값들은 동일한 3D 공간상에 존재한다. 평면에서는 이를 확장하여 2D 영상에서 네 개의 격자점을 둘러싸는 바운딩 박스를 균등하게 나누면 동일한 개념이 적용된다. 그러나 선의 경우와는 달리 평면에서는 x 축과 y 축에 대해 각각 M 등분 및 N 등분 해야 한다. 평면 기반 정합에서도 약간차 추정을 통해 획득된 3D 좌표 값이 오차를 포함하고 있기 때문에 N 개의 영상을 사용하는 경우 N 개의 3D 좌표 값 가운데 네 개의 격자점으로 구성되는 평면에 가장 근접한 3D 좌표를 최종 값으로 선택한다.

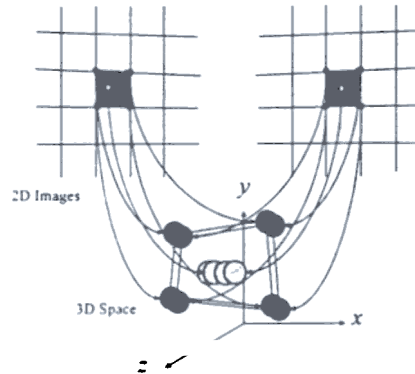


그림 5. 평면 기반 정합을 위한 개념도

그러나, 평면 기반 정합에서의 문제점은 네 개의 3D 점을 사용하여 형성되는 평면이 완전한 평면을 만들지 못한다는 것이다. 즉, 네 개의 3D 좌표가 오

차를 포함하기 때문에 동일한 평면상에 위치하지 않는다는 문제점이 존재한다. 따라서, 네 개의 3D 점을 통해 형성되는 평면으로부터의 거리를 사용하는 방법과 비슷하게 네 개의 점으로부터의 거리를 최소화하는 3D 좌표를 최종 값으로 선택한다.

네 개의 격자점에 대한 3D 좌표를 각각 (x_i, y_i, z_i) ($i=0, 1, 2, 3$)라 하면, 임의의 점 $A(x_a, y_a, z_a)$ 로부터 각 점까지의 거리의 합 $dist$ 는 다음과 같이 표현된다.

$$dist = \sum_{i=0}^3 \text{sqr}((x_i - x_a)^2 + (y_i - y_a)^2 + (z_i - z_a)^2) \quad (17)$$

따라서, N 개의 영상에서 얻어지는 N 개의 화소에 대한 대한 3D 좌표 가운데 위의 식을 최소화하는 3D 좌표를 최종 값으로 선택한다. 이와 같은 선과 평면을 이용한 정합 방법을 사용함으로써 깊이 정보가 정확하게 획득되지 않는 부분이나 잡음에 의해 양안차 추정의 결과가 정확하지 않는 경우 등에 있어 제안된 방법이 유용하게 사용될 수 있다.

정합된 임의의 3D point cloud를 모델링하기 위해 카메라와 객체와의 기하학적 관계 및 카메라에 투영된 객체의 영상과 3D point cloud와의 관계를 사용한다 [10]. 즉, 정합을 통해 최종적으로 얻어진 3D point cloud를 카메라의 외부 파라미터를 이용하여 영상 획득에 사용된 두 대의 카메라의 영상 평면으로 투영한다. 이때, 서로 보이지 않는 부분을 모델링에 반영하기 위해 정합된 3D point cloud를 오른쪽 카메라와 왼쪽 카메라로 투영될 두 부분으로 나누고 이들을 오른쪽과 왼쪽 카메라 각각에 투영함으로써 두 개의 영상 평면상의 점들을 얻어낸다. 그리고 bilinear interpolation을 통해 보간하여 영상 평면의 격자점에 대응하는 3D 좌표계의 점을 구한다. 영상 평면상의 연결 정보를 이용하여 양쪽에 대해 각각 메쉬 모델을 만들고 메쉬 연결 과정을 통해 나뉘어진 부분을 연결함으로써 3D 공간에서의 메쉬 모델을 생성한다. 이렇게 생성된 가상 환경을 사용자가 스테레오스코픽 HMD를 착용하고 네비게이션할 수 있도록 하였다. 즉, HMD 상에 부착된 트랙커에 의해 장면이 실시간으로 변화되어 사용자가 몰입감을 효과를 느끼며 네비게이션이 가능하다.

4. 실험 및 분석

본 실험은 일반적인 실내 환경의 정상적인 조명하에서 수행되었다. 영상 획득을 위해 멀티뷰 카메라인 Digiclops를 사용하였으며, Xeon 2.8 GHz CPU 컴퓨터에서 수행되었다 [11]. 본 실험에서 사용된 캘리브레이션 패턴은 체스 보드 모양을 띄며, 두 개의 직각 평면 상에 32 개의 격자들을 갖는다. 그림 6은 실험을 위한 시스템 셋업을 나타낸다. 그림 6(a)는 시스템 구성, 그림 6(b)는 직각 패턴으로, 체스 보드 상의 격자간의 간격은 25 cm이다.

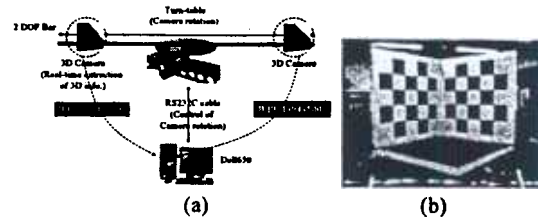


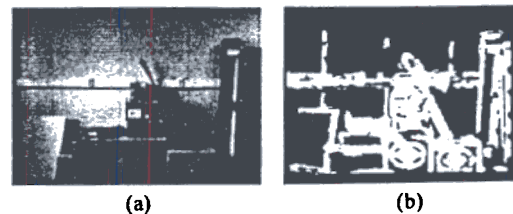
그림 6. 실험 셋업 (a) 시스템 구성 (b) 직각 패턴

그림 7은 선 및 평면 기반 정합의 예를 직각 캘리브레이션 패턴에 적용한 결과를 보인 것이다. 그림 7(a)와 그림 7(d)는 왼쪽과 오른쪽 카메라에서 얻은 두 장의 영상에 대해 선 및 평면 기반 정합을 적용한 결과를 정면과 위에서 본 그림 (b)와 (e)이다. 그림 7(b)와 그림 7(c), 그림 7(e)와 그림 7(f)는 각각 턴테이블을 한 번, 두 번 회전시켜 얻은 네 장, 여섯 장에 대해 정합 과정을 수행한 결과를 나타낸다. 그림에 보이는 것과 같이 영상의 수가 늘어남에 따라 정합 결과가 보다 개선됨을 알 수 있다. 그러나, 각 영상으로부터 얻은 3D 깊이 정보가 오차를 포함하고 있기 때문에 많은 영상을 사용하더라도 일정한 값 이하로는 오차가 감소하지 않는 것 또한 예측할 수 있다.



그림 7. 선 및 평면 기반 정합 (a)(d) 두 장면을 이용한 정합 결과를 정면과 위에서 본 그림, (b)(e) 네 장면을 이용한 정합 결과를 정면과 위에서 본 그림 (c)(f) 여섯 장면을 이용한 정합 결과를 정면과 위에서 본 그림

그림 8은 유효한 깊이 정보를 이용해서 양안차 정보가 없는 부분을 채운 결과를 보인 것이다. 그림 8(a) 및 그림 8(b)는 원영상과 대응되는 양안차 맵을 보인 것이다. 그리고, 그림 8(c)는 유효한 깊이 정보. 그림 8(d)는 이를 이용하여 보완된 깊이 정보를 보인 것이다. 이러한 과정을 통해 양안차를 획득하기 어려운 부분에 대해서도 3D point cloud를 획득하는 것이 가능하다.



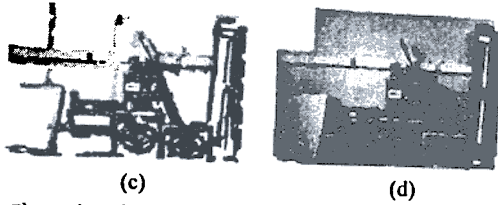


그림 8. 유효한 깊이 정보를 이용한 보완 (a) 원영상 (b) 양안차 맵 (c) 유효한 깊이 정보 (d) 보완된 깊이 정보

그림 9는 모델링된 실내 환경을 여러 방향에서 바라본 장면들을 보인 것이다. 그림 9(a)는 실내 환경에 대한 조감도를 보인 것이다. 여기서, 원으로 표시된 부분이 그림 9(b), 그림 9(c) 그리고 그림 9(d)를 각각 중심 위치에서 바라본 것이다.

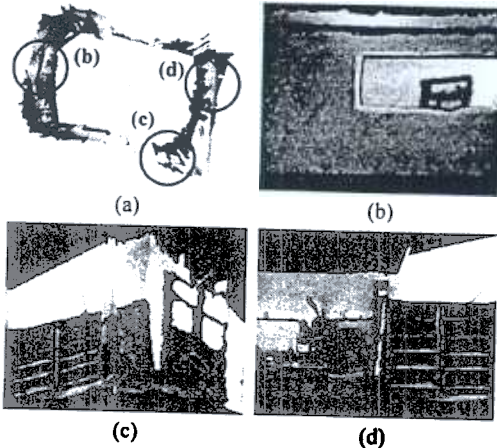


그림 9. 모델링된 실내 환경 (a) 조감도 (b) 장면 1 (c) 장면 2 (d) 장면 3

그림 10은 가상 환경에 증강된 모델링된 패턴을 보인 것으로, 그림 10(a) 및 그림 10(b)는 사용자가 가상 환경의 중심 위치에서 고개를 좌측과 우측으로 움직이는 경우의 장면을 보인 것이다. 그림에서 알 수 있는 것과 같이 사용자의 시선이 움직임에 따라 가상 환경 상에 존재하는 객체가 3D 공간상에 모델링되어 존재하는 것임을 알 수 있을 뿐만 아니라 HMD의 양안에 서로 다른 장면을 디스플레이함으로써 사용자에게 3D 느낌을 전달하는 것도 가능하게 된다.

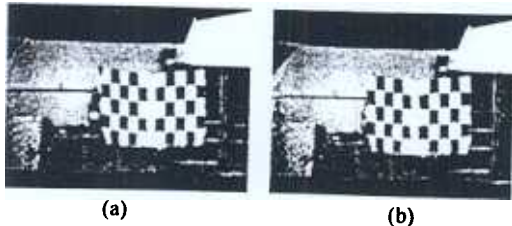


그림 10. 패턴의 증강 (a) 왼쪽 방향에서 보는 경우 (b) 오른쪽 방향에서 보는 경우

5. 결론 및 추후과제

본 논문에서는 영상 기반 가상 환경 생성을 위해 다수의 멀티뷰 카메라를 사용하여 3D 깊이 정보에 기반한 정합 방법을 제안하였다. 제안한 방법을 사용함으로써 깊이 정보가 정확하게 획득되지 않는 부분이나 잡음에 의해 양안차 추정 결과의 정확도가 낮은 경우 등에 있어 효과적으로 사용될 수 있다. 또한 이렇게 생성된 가상 환경을 사용자가 트래커가 부착된 스테레오스코픽 HMD를 착용하고 몰입감을 느끼며 네비게이션 할 수 있도록 하였다. 제안된 방법은 기존의 컴퓨터 그래픽스 기반의 가상 환경이 갖는 실시간 렌더링 문제를 완화시킬 뿐만 아니라 사용자에게 보다 큰 사실감 및 몰입감을 제공할 것으로 보인다. 추후에는 생성된 가상 환경 내에 새로운 가상 객체를 효과적으로 삽입하는 방법 그리고 이를 위한 가상 환경 내의 조명 조건의 분석에 관한 연구를 진행할 예정이다.

감사의 글

생성된 3D point cloud의 모델링 및 HMD 상의 재현을 도와준 이원우와 Muhammad Usman Sargaana에게 감사드립니다.

6. 참고문헌

- [1] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.
- [2] A. Johnson and S. Kang, "Registration and Integration of Textured 3-D Data," Tech. report CRL96/4, Cambridge Research Lab, 1996.
- [3] G. Blais and M.D. Levine, "Registering multiview range data to create 3-D computer objects," *IEEE Trans. PAMI*, vol. 17, no. 8, pp. 820-824, 1995.
- [4] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo project: 3D scanning of large statues. *SIGGRAPH'00*, pp. 131-144, July 2000.
- [5] T. Kanade and P. Rander, "Virtualized Reality: Constructing Virtual Worlds from Real Scenes," *IEEE MultiMedia*, vol. 4, no. 1, pp. 34-47, Jan. - Mar. 1997.
- [6] K. Nishino and K. Ikeuchi, "Robust Simultaneous Registration of Multiple Range Images Comprising A Large Number of Points," *ACCV2002*, 2002.
- [7] 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.
- [8] S. Kim, E. Chang, C. Ahn and W. Woo, "Image-based Panoramic 3D Virtual Environment using Rotating Two Multi-view Cameras," *IEEE Proc. ICIP2003*, vol. 1, pp. 917-920, 2003.
- [9] S. Kim and W. Woo, "Virtual Environment Generation using 3D Image-based Panorama," *15th Workshop on IPIU03*, pp. 111-116, Jan. 8-10, 2003.
- [10] W. Lee and W. Woo, "Mesh Modeling for Image-based Virtual Environment Generation," *16th Workshop on IPIU04*, Jan. 9-10, 2004. (accepted)
- [11] Point Grey Research Inc., <http://www.ptgrey.com>, 2002.