

가상 객체 합성을 위한 단일 프레임에서의 안정된 카메라 자세 추정

박종승[†] · 이범중^{**}

요 약

본 논문에서는 실시간 증강현실 시스템에서의 가상 객체 삽입을 위한 빠르고 안정된 카메라 자세 추정 방법을 제안한다. 단일 프레임에서 마커의 특징점 추출을 통해 카메라의 회전행렬과 이동벡터를 추정한다. 카메라 자세 추정을 위해 정사영 투영모델에서의 분해기법을 사용한다. 정사영 투영모델에서의 분해기법은 객체의 모든 특징점의 깊이좌표가 동일하다고 가정하기 때문에 깊이좌표의 기준이 되는 참조점의 설정과 점의 분포에 따라 카메라 자세 계산의 정확도가 달라진다. 본 논문에서는 실제 환경에서 일반적으로 잘 동작하고 융통성 있는 참조점 설정 방법과 이상점 제거 방법을 제안한다. 제안된 카메라 자세추정 방법에 기반하여 탐색된 마커 위치에 가상객체를 삽입하기 위한 비디오 증강 시스템을 구현하였다. 실 환경에서의 다양한 비디오에 대한 실험 결과, 제안된 카메라 자세 추정 기법은 기존의 자세추정 기법만큼 빠르고 기존의 방법보다 안정적이고 다양한 증강현실 시스템 응용에 적용될 수 있음을 보여주었다.

키워드 : 정사영법 카메라 위치 추정, 카메라 추적, 가상 객체 삽입

Reliable Camera Pose Estimation from a Single Frame with Applications for Virtual Object Insertion

Jongseung Park[†] · Bumjong Lee^{**}

ABSTRACT

This paper describes a fast and stable camera pose estimation method for real-time augmented reality systems. From the feature tracking results of a marker on a single frame, we estimate the camera rotation matrix and the translation vector. For the camera pose estimation, we use the shape factorization method based on the scaled orthographic projection model. In the scaled orthographic factorization method, all feature points of an object are assumed roughly at the same distance from the camera, which means the selected reference point and the object shape affect the accuracy of the estimation. This paper proposes a flexible and stable selection method for the reference point. Based on the proposed method, we implemented a video augmentation system that inserts virtual 3D objects into the input video frames. Experimental results showed that the proposed camera pose estimation method is fast and robust relative to the previous methods and it is applicable to various augmented reality applications.

Key Words : Camera Pose From Orthography, Camera Tracking, Virtual Object Insertion

1. 서 론

증강현실 응용에서 가장 중요한 요소기술은 실사 비디오에 3D 가상 그래픽 객체를 자연스럽게 삽입하는 기술이다. 가상 객체를 정확한 위치에 올바른 자세로 삽입하기 위해서는 정확하고 신뢰성 있는 카메라 자세 계산이 필요하다.

최근 카메라 자세 계산을 위한 연구는 활발히 진행되어 왔다. 가장 대표적인 방법으로 마커 기반 접근과 특징점 기

반 접근의 두 가지로 나누어 볼 수 있다. 특징점 기반의 접근에서는 특징점을 자동으로 선택하고 추적하는 방법[1]과 프레임에서 지터와 드리프트를 고려한 추적방법[2]이 있고, 선이나 원형의 특징을 이용한 추적방법[3], 배치를 사용하여 추적하는 방법[4], 특징점들을 클러스터링 하여 추적하는 방법[5]도 있다. 특징점 기반의 한 예로 정의된 평면을 추적하는 방법[6], 조명의 변화와 경계에 대해 강한 추적방법[7]이 있고, 텍스처와 모서리의 정보를 이용하여 추적하는 방법[8]도 있다. 마커 기반의 접근은 기존의 마커를 사용하거나 주어진 환경에서 확실히 측정된 물체를 사용하는 방법[9], 보이지 않는 마커도 추적할 수 있는 방법[10]이 있고, 래스터

[†] 종신회원: 인천대학교 컴퓨터공학과 조교수
^{**} 준 회원: 인천대학교 컴퓨터공학과 석사과정
논문접수: 2006년 6월 2일, 심사완료: 2006년 8월 17일

스캐ンを 사용하여 좁은 마커를 탐색하는 방법[11]과 정사각형의 마커를 사용하는 방법[12]이 있다.

특징점 기반의 접근은 객체의 기하 정보를 모른다고 가정하고 카메라 자세를 계산하기 때문에 속도가 느리고 오프라인 모드에서 많이 사용한다. 대부분의 증강현실 시스템 응용들은 실시간으로 실세계에 가상 객체를 합성하기 위해 온라인으로 카메라 자세를 계산해야 한다. 마커 기반의 접근은 마커의 기하 정보를 알고 있다고 가정하고 카메라 자세를 계산하기 때문에 속도가 빠르고 온라인 모드에서 실행할 수 있다.

카메라 모델은 불필요하게 계산과정이 복잡하여 시간이 오래 걸린다. 카메라 모델은 11개의 자유도(DOF; Degree of Freedom)를 가지고 불필요하게 복잡하다. 증강현실 시스템의 응용을 위한 카메라 자세 추정은 실시간으로 처리할 수 있을 만큼 빨라야 하기 때문에 정확한 값과 큰 차이가 없는 근사값을 계산한다.

본 논문에서는 빠른 계산을 위해 마커를 사용하고 초기의 추정이 필요 없는 선형계산법을 사용하여 빠르고 강한 카메라 자세 추정 방법을 제안한다. 추정된 카메라 자세의 정확도는 마커의 기하학적 특성과 참조점의 설정에 따라서 달라진다. 마커의 특성에 민감하지 않도록 동적으로 참조점을 설정하도록 하여 카메라 자세의 정확도를 높일 수 있는 카메라 자세 추정 방법을 제안한다. 참조점과 상당히 멀리 떨어져 있는 이상점(outlier)은 카메라 자세 추정의 정확도를 떨어뜨리기 때문에 이상점의 제거 방법도 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 카메라 자세를 계산하는 방법들과 카메라 모델을 설명하고 3장에서는 본 논문에서 제안한 정사영 투영모델에서의 분해기법을 개선한 방법에 대해 설명한다. 4장에서는 본 논문에서 제안한 방법으로 실험한 결과를 보여주고 5장에서 결론을 기술한다.

2. 고속 자세 추정을 위한 단순화된 카메라 모델

공간에서 임의의 점 $x=[x \ y \ z \ 1]^T$ 는 영상의 점 $m=[u \ v \ 1]^T$ 로 투사되고 이는 투사식 $\lambda m = P x$ 으로 표현된다. λ 는 임의의 스칼라 값이고 투사 행렬 P 는 3×4 행렬이다. 원근 카메라 모델의 투사 행렬은 $P = K[Rt]$ 로 분해될 수 있다. K 는 3×3 카메라 보정(calibration) 행렬이고, 행렬 R 과 벡터 t 는 카메라에 대한 객체의 자세를 표현한다. 보정 행렬은 f_x 와 f_y 로 표현되는 영상의 두 축 방향으로의 초점과, p_x 와 p_y 로 표현되는 카메라 센서 중심 좌표의 4개의 카메라 내부 파라미터로 이루어져 있다. 행렬 $R = [r_x \ r_y \ r_z]^T$ 는 3×3 직교 행렬이고 $t = [t_x \ t_y \ t_z]^T$ 는 3×1 벡터이다. R 과 t 는 모델과 카메라 사이의 회전과 이동의 관계를 표현한다. 내부 파라미터는 미지수이지만 대부분의 경우 일정하고 부드럽게 변한다. 내부 파라미터는 미리 측정하여 구할 수 있으므로 4개의 내부 파라미터는 미리 알고 있다고 가정한다. 이제 카메라 모델의 단순화를 위해서 K 는 항등행렬로 가정한다.

원근 모델에서 영상 좌표와 모델 좌표는 비선형 식으로 표현된다. 원근투영 모델에서 객체와 카메라의 거리가 객체의 크기에 비해 매우 크다는 제약을 돕으로써 원근투영 모델의 선형근사가 가능하다. 원근 카메라 모델의 잘 알려진 원근 근사법은 weak-perspective 모델[13]이다. Weak-perspective 모델은 객체의 크기가 객체와 카메라 사이의 거리에 비해 작을 때 원근 모델의 대안으로 사용될 수 있다. Weak-perspective 모델에서 모든 객체의 점은 객체의 참조점 \bar{x} 를 지나고 영상 평면과 평행한 평면에 있기 때문에 모든 객체의 점은 동일한 깊이 $\bar{z} = (\bar{x} - t) \cdot r_z$ 를 가진다.

객체의 점 x 와 영상의 점 m 의 투사 식은 아래와 같다.

$$u = (1/\bar{z})(r_x \cdot (x-t)), \quad v = (1/\bar{z})(r_y \cdot (x-t))$$

$$\bar{x} = 0 \text{ 이라 가정하면 } \bar{z} = -t \cdot r_z \text{ 이고 투사식을 유도하면 식 (1)과 같다.}$$

$$u = \tilde{r}_x \cdot x + \bar{x}/\bar{z}, \quad v = \tilde{r}_y \cdot x + \bar{y}/\bar{z} \tag{1}$$

여기서 $\tilde{r}_x = (1/\bar{z})r_x$, $\tilde{r}_y = (1/\bar{z})r_y$, $\bar{x} = -t \cdot r_x$, $\bar{y} = -t \cdot r_y$ 이다. 식 (1)은 각 점의 직교 투영에 대응된다. 식 (1)에서 $|\tilde{r}_x| = |\tilde{r}_y| = 1$ 과 $|\tilde{r}_x| \cdot |\tilde{r}_y| = 0$ 의 조건을 사용하여 직교 분해기법[14]에 의해 해를 구할 수 있다.

동일 평면상의 점들에 대한 카메라 자세 추정[15]도 가능하지만 매우 많은 반복횟수로 인해 시간이 오래 걸리고 정확도도 떨어지기 때문에 좋은 방법은 아니다. 본 논문은 특징점의 빠른 탐색과 추적을 위해 마커를 사용한다.

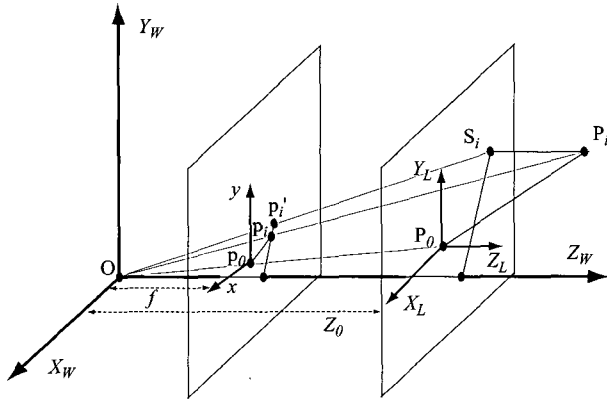
3. 제안된 카메라의 자세 계산 방법

3.1 정사영 투영모델에서의 분해기법

단일 프레임으로부터 카메라 자세를 계산하는 기법으로 POS(Pose from Orthography) 방법[18]이 있다. POS 방법은 객체의 회전행렬과 이동벡터를 구하는 알고리즘이다. 회전행렬 R 은 객체 좌표 시스템에서의 카메라 좌표 시스템의 기저(basis) 벡터 b_x, b_y, b_z 의 좌표를 행벡터로 구성하여 이루어진다. 회전행렬은 아래와 같이 쓸 수 있다.

$$R = [b_x^T \ b_y^T \ b_z^T]^T$$

회전행렬을 계산하기 위해서 객체 좌표 시스템에서의 벡터 b_x 와 b_y 를 계산하고 벡터 b_z 는 b_x 와 b_y 의 외적으로 구한다. 이동벡터는 투사의 중심점 O 와 참조점 $P_0 = (X_0, Y_0, Z_0)$ 사이의 벡터 $-V_0$ 이다. 초점거리를 f 라 하고 투사의 중심점 O 와 참조점의 영상 좌표 $p_0 = (x_0, y_0)$ 사이의 벡터를 $-v_0$ 라 하면 이동벡터는 $t = (Z_0/f)(-v_0)$ 와 같이 나타낼 수 있으므로 객체의 이동을 계산하기 위해서는 단지 Z_0 만을 계산하면 된다. 카메라 자세를 계산하기 위해서는 b_x , b_y 와 Z_0 만을



(그림 1) 정사영 투영모델의 기하도

계산하면 된다. (그림 1)에서 정사영 투영 모델의 기하구조를 보여주고 있다.

SOP(Scaled Orthographic Projection)[18]에서는 객체의 참조점 P_0 의 깊이좌표 Z_0 가 객체의 다른 모든 점들의 깊이 좌표와 같다는 가정이 필요하다. SOP의 스케일 요소 s 는 f/Z_0 이다. SOP와 원근 투영에서 참조점 P_0 의 영상 좌표는 $p_0=(x_0, y_0)$ 와 같다. SOP의 영상 좌표 $p'_i=(x'_i, y'_i)$ 는 식 (2)와 같다.

$$\begin{aligned} x'_i &= fX_0/Z_0 + f(X_i - X_0)/Z_0 = x_0 + s(X_i - X_0), \\ y'_i &= fY_0/Z_0 + f(Y_i - Y_0)/Z_0 = y_0 + s(Y_i - Y_0) \end{aligned} \quad (2)$$

객체 좌표 시스템에서 참조점과 객체의 점 P_i 에 대한 벡터 V_i 의 좌표와 두 영상 점 $p_0=(x_0, y_0)$ 와 점 $p_i=(x_i, y_i)$ 의 위치와의 관계식은 식 (3)과 같다.

$$V_i \cdot b_x = x_i(1 + \epsilon_i) - x_0, \quad V_i \cdot b_y = y_i(1 + \epsilon_i) - y_0 \quad (3)$$

원근 투영 영상 좌표와 SOP 영상 좌표간의 차이 즉, SOP 영상 좌표의 오차를 표현하기 위한 ϵ_i 는 다음과 같은 식으로 정의된다.

$$\epsilon_i = \frac{1}{Z_0} V_i \cdot b_z$$

식 (3)에서의 $x_i(1 + \epsilon_i)$ 와 $y_i(1 + \epsilon_i)$ 는 P_i 의 스케일된 직교 투영점의 좌표 $p'_i=(x'_i, y'_i)$ 이다. 벡터 V_i 는 벡터 S_i 와 $V_i - S_i$ 의 합이다. 벡터 S_i 와 영상 벡터 v'_i 은 Z_0/f 의 비율로 비례한다. 이를 식으로 나타내면 아래와 같다.

$$V_i = \frac{Z_0}{f} v'_i + (V_i - S_i)$$

위의 식의 양변에 카메라 좌표 시스템의 단위벡터 b_x 를 내적하면, $(V_i - S_i) \cdot b_x$ 는 x-축과 수직이므로 0이 되고

$v'_i \cdot b_x$ 는 벡터 v'_i 의 x-좌표인 $x'_i - x_0$ 가 된다. 이를 정리하면 식 (4)를 얻을 수 있다.

$$V_i \cdot b_x = x'_i - x_0, \quad V_i \cdot b_y = y'_i - y_0 \quad (4)$$

식 (4)를 식 (3)과 비교하면 p'_i 의 좌표는 $x'_i = x_i(1 + \epsilon_i)$ 와 $y'_i = y_i(1 + \epsilon_i)$ 로 나타낼 수 있다. 또한 $b'_x = (f/Z_0)b_x$ 와 $b'_y = (f/Z_0)b_y$ 임을 이용하여 식 (3)을 다시 정리하면 식 (5)가 된다.

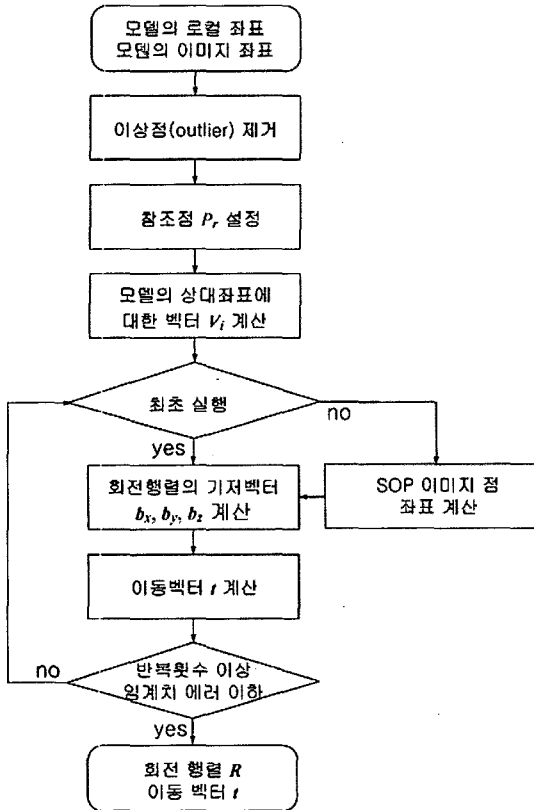
$$V_i \cdot b'_x = x_i(1 + \epsilon_i) - x_0, \quad V_i \cdot b'_y = y_i(1 + \epsilon_i) - y_0 \quad (5)$$

정사영 투영모델에서 카메라의 자세를 계산하기 위해 ϵ_i 가 정확하게 주어지지 않으면 근사값을 구하게 된다. POSIT (POS with Iterations) 알고리즘[16]은 반복법을 통하여 더 정확한 값에 가까워지도록 하고 있다. 그러나 POSIT 알고리즘은 두 가지 제한점을 가진다. 첫 번째는 반드시 네 개 이상의 객체의 3D 로컬 좌표와 단일 프레임에서의 대응되는 영상 좌표가 주어져야 한다는 것이다. 두 번째는 주어진 점들이 모두 동일 평면상에 존재하지 않는 non-coplanar이어야 한다.

POSIT 방법에서는 정사영 투영 모델에 기반하기 때문에 참조점의 설정에 따라 정확도가 크게 떨어질 수도 있다. 본 논문에서의 제안 방법은 마커에서의 특징점 중에서 가장 최적의 특징점을 참조점으로 설정하도록 하여 다양한 마커에 대해서 오차를 최소로 줄이도록 하는 기법을 제시한다. 각 특징점에 대해서 마커를 대표할 수 있는 특징점을 선정하고 이를 참조점으로 지정하여 카메라 자세를 계산하도록 한다. 다음 절에서 참조점을 지정하고 카메라 자세를 계산하는 전체적인 알고리즘을 설명한다.

3.2 제안된 카메라 자세 계산 알고리즘

본 논문에서는 POSIT 알고리즘의 문제점인 참조점의 설정 방법에 대해 오차의 최소화를 보장하면서 마커의 형태에 융통성있는 방법을 제안한다. 카메라 자세 계산의 전체 흐름도는 (그림 2)와 같다. 모델의 로컬 좌표는 이미 알려져 있다고 가정한다. 각 영상에서 모델의 특징점들을 탐색하여 모델의 영상 좌표를 획득한다. 이들 모델의 특징점 중에서 이상점이 있을 경우 정확한 카메라 자세 추정을 저해하는 요소가 되기 때문에 이 이상점을 제거한다. 다음으로 제거되지 않은 특징점 중에서 오차를 최소화할 수 있는 참조점을 선정한다. 선정된 참조점에 상대적인 모델의 상대좌표를 계산한다. 이제 기저벡터와 이동벡터를 반복적으로 계산하여 오차를 최소화시킨다. 지정된 에러의 임계치 이하가 되거나 지정된 반복횟수 이상이 되거나 이전 수행과 현재 수행의 에러가 같게 되는 카메라 이동벡터와 회전행렬을 추정값으로 결정한다.



(그림 2) 카메라 자세계산을 위한 시스템 흐름도.

입력 : f (초점거리), P (모델의 로컬 좌표), p (모델의 영상 좌표).
 출력 : R (카메라 회전 행렬), t (카메라 이동 벡터).

단계 1 : $i = 0, r = 0, d_r = 0, cnt_i = 0$;
 단계 2 : 각 점이 다른 점들로부터 가장 멀리 떨어져 있는 경우의 수 계산.
 · $cnt_i = cnt_i + 1$ (여기서 $i = \arg \max_j |P_i - P_j|, j \in [0, N-1], N$ 은 점의 수)
 단계 3 : cnt_i 의 값에 따라 특징점들을 오름차순으로 $\{P_0, P_1, \dots, P_{N-1}\}$ 로 정렬
 단계 4 : 이상점을 제거. $M=N$.
 · for $i = 1$ to $\lfloor N/2 \rfloor$
 $t = \arg \min_k \sum_{j=0}^{N-i-1} |P_k - P_j|$
 $d = \max_j \sum_{j=0}^{N-i-1} |P_i - P_j|$
 if $d < |P_i - P_{N-i}|$ then
 $P_{N-i}, P_{N-i+1}, \dots, P_{N-1}$ 의 점을 제거;
 $M=N-i$;
 반복 종료;
 end
 end
 단계 5 : 각 점에 대해서 ($i=0$ 에서 $i=M-1$ 일 때까지) 아래를 반복함.
 · 모델의 특징점 P_i 와 다른 특징점들과의 거리 합
 $d_i = \sum_{j=0, j \neq i}^{M-1} (P_i - P_j)$ 을 계산.
 · d_i 값으로부터 참조점의 인덱스 r 를 다음과 같이 결정함
 if $d_i < d_r$ then $r \leftarrow i, d_r \leftarrow d_i$
 단계 6 : 참조점 P_r 을 이용하여 행렬 A 를 (식 7)로 계산함.

· $V_i = P_i - P_r$ 으로 벡터 $V_i (i = 0, \dots, M-1)$ 를 형성함.
 · 벡터 V_i 들을 행렬의 행 벡터들로 구성하여 크기 $M \times 3$ 의 행렬 A 를 형성함.
 단계 7 : A 의 의사역행렬인 $3 \times M$ 크기의 행렬 B 를(식 8)로 계산함
 단계 8 : b_x, b_y, s 를 (식 10)로 계산함.
 · (식 5)로 $x_i', y_i' (i = 0, \dots, M-1)$ 를 계산하고, 이로부터 M 크기의 벡터 x' 와 y' 를 형성함.
 · $b_x' = Bx', b_y' = By'$ 으로 b_x', b_y' 를 계산함.
 · b_x', b_y' 로부터 s_1, s_2 를 계산하고, b_x', b_y', s_1, s_2 로부터 b_x, b_y 를 계산함.
 단계 9 : 단위벡터 b_x 와 Z_0 를 다음과 같이 계산함.
 · $b_x = b_x \times b_y, Z_0 = f/s$
 단계 10 : 회전행렬과 이동벡터를 다음과 같이 계산함.
 · $t = -(1/s)V_0, R = [b_x^T \ b_y^T \ b_z^T]^T$

(그림 3) M개의 점을 사용한 제안된 카메라 자세 추정 알고리즘.

(그림 3)은 정사영 투영모델에서의 분해기법을 개선하여 본 논문에서 제안한 카메라 자세를 계산하는 알고리즘이다. 이상점은 정확성을 떨어뜨리기 때문에 먼저 이상점을 제거하는 방법에 대해 설명하고, 참조점의 설정에 따라 정확도가 달라지기 때문에 정확도를 높이는 방법에 대해 설명한다. 정확한 카메라 자세를 계산하기 위해서는 먼저 이상점을 제거하고 제거되지 않은 객체의 모든 점들에 대한 무게중심을 구한다. 객체의 점들 중에서 객체의 특징점들의 평균값인 무게중심에 가장 가까운 점을 참조점으로 설정할 경우 정사영 투영모델이 객체의 모든 점이 한 평면에 존재한다는 가정에서 발생하는 오차를 최소화할 수 있기 때문에 가장 정확한 카메라 자세를 계산할 수 있다.

단계 1에서 참조점 계산을 위한 변수들의 초기화를 수행한다. 단계 2부터 단계 4까지는 이상점을 제거하기 위한 단계이다. 단계 2에서는 이상점을 제거하기 위해 모든 점에 대해 다른 점들과 가장 멀리 떨어져 있는 경우의 수를 계산한다. 단계 3에서는 단계 2에서 구한 경우의 수에 따라 모델의 특징점들을 오름차순으로 정렬한 후에 단계 4에서 이상점을 최대 모델의 특징점 개수의 반까지 제거한다. 단계 5는 가장 정확한 참조점을 설정하기 위해 객체의 특징점의 수만큼 반복 수행하는 단계이다. 객체의 각 점에 대해서 그 점을 제외한 객체의 모든 점들과의 거리 차이를 합산한다. 그 다음, 합산된 값을 검사하여 최소값일 때의 인덱스를 참조점의 인덱스로 저장한다. 다른 점들과의 거리 합이 가장 작은 점에 대한 인덱스를 참조점의 인덱스로 지정한다. 식 (6)은 참조점에 대한 인덱스를 구할 수 있는 식이다.

$$r = \arg_k (\min_i \sum_j |P_k - P_j|) \quad (6)$$

단계 6은 결정된 참조점을 사용하여 행렬 A 를 생성하는 단계로 식 (7)에 의해 모델의 월드 좌표들간의 벡터 V_i 로 이루어진 $M \times 3$ 의 행렬 A 를 생성한다.

$$A = [V_0 \dots V_{M-1}]^T \quad (7)$$

단계 7은 A 의 의사역행렬인 B 행렬을 구하는 단계로 식 (8)에 의해 행렬 B 를 구한다.

$$B = A^+ = (A^T \cdot A)^{-1} \cdot A^T \quad (8)$$

단계 8은 단위벡터 b_x, b_y 와 스케일요소 s 를 구하는 단계로 식 (8)의 행렬 B 를 이용해 식 (9)에 의해 b_x' 와 b_y' 를 구한다. 구해진 b_x' 와 b_y' 로부터 식 (10)과 같이 s_1 과 s_2 를 구한다.

$$b_x' = Bx', b_y' = By' \quad (9)$$

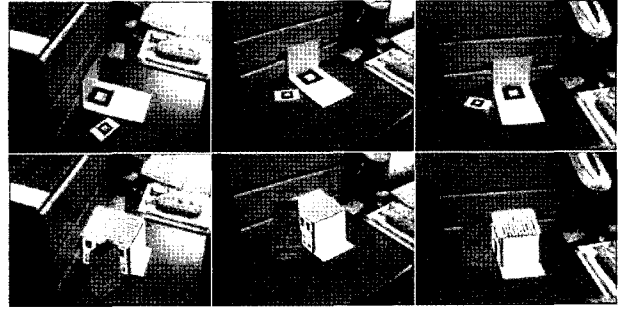
$$s_1 = (b_x' \cdot b_x')^{1/2}, s_2 = (b_y' \cdot b_y')^{1/2} \quad (10)$$

단계 9는 단위벡터 b_x 와 모델 거리 Z_0 를 구한다. b_x 는 b_x' 와 b_y' 의 외적으로 구하고 모델 거리는 $Z_0 = 2f / (s_1 + s_2)$ 에 의해 구하게 된다. 마지막 단계인 단계 10은 위에서 구해진 값들로 객체의 이동벡터와 회전행렬을 구함으로써 카메라 자세 계산을 완료한다.

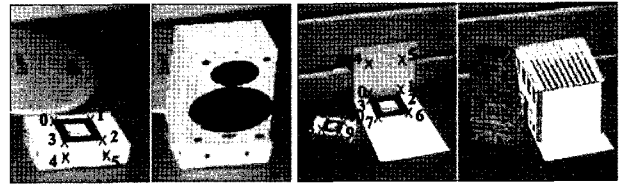
4. 실험 결과

제안된 방법의 유효성을 증명하기 위해 카메라 자세 계산 시스템을 구현하였다. 마커의 모델 좌표는 알고 있다고 가정한다. 각 프레임에 대해 마커의 특징점은 각 영상 프레임에서 빠른 마커 탐지 알고리즘을 적용해 얻는다. 카메라 내부 파라미터는 절대적인 정확한 형태좌표를 얻기 위해서 필요하다. 본 실험에서는 카메라 영상의 획득 시 즉시 적용하여 파라미터를 얻을 수 있는 온라인 카메라 캘리브레이션 모듈을 사용하였다. 캘리브레이션 알고리즘은 체크무늬 패턴 영상을 획득하여 모서리점 좌표를 사용하는 방식[17]으로 일반적으로 잘 동작한다.

(그림 4)에서 첫 번째 줄의 영상들은 마커가 놓여 있는 원본 영상이다. 두 번째 줄의 영상들은 첫 번째 줄의 영상들에서 마커를 추적하고 카메라의 자세를 계산하여 마커의 위치에 가상 그래픽 객체인 스피커를 삽입한 결과이다. 세 번째 줄의 영상들은 두 개의 마커에 대한 원본 영상이다. 네 번째 줄은 세 번째 줄의 영상들에서 두 마커를 추적하여 카메라 자세를 계산하고 두 마커의 위치에 각각 굴뚝과 건물을 삽입한 결과이다.



(그림 4) 두 실사 비디오에 대한 원본 프레임과 가상객체가 삽입된 프레임: Test Set #1 (위), Test Set #2 (아래).



(그림 5) 두 실사 비디오에 대한 삽입 전의 마커의 모습과 삽입 후의 가상객체 모습: Test Set #1 (왼쪽), Test Set #2 (오른쪽).

<표 1> Test Set #1의 6개의 영상 점에 대한 참조점 선택 에러.

| 특징점 | 점 0 | 점 1 | 점 2 | 점 3 | 점 4 | 점 5 |
|------------------|---------|---------|---------|---------|---------|---------|
| 평균 에러 (pixel 단위) | 2.33763 | 2.46799 | 1.66688 | 1.75333 | 1.96703 | 1.95909 |

(그림 5)의 영상들은 (그림 4)의 프레임에서 확대한 영상으로 각 마커에 가상객체가 자연스럽게 삽입된 모습을 확대하여 보여주고 있다.

<표 1>은 (그림 5)의 왼쪽의 마커에서 6개의 점을 차례로 참조점으로 하였을 경우의 에러이다. <표 1>에서 점 2와 점 3을 참조점으로 했을 경우 가장 적은 에러가 나타난다. 식 (6)으로부터 구해진 참조점은 참조점과 그 외의 모델 점들 사이의 거리가 가장 작은 참조점이고 이때 가장 정확한 카메라 자세 계산이 가능하다.

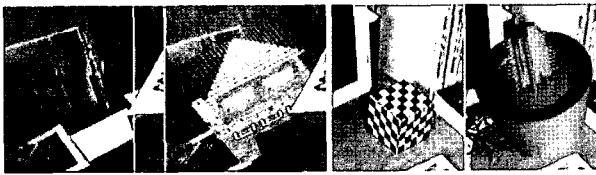
(그림 6)의 상단 그림은 가운데에 노란 육면체가 마커로 사용되는 Test Set #3의 비디오 프레임이다. 하단 그림은 노란 육면체를 마커로 하여 특징점을 추적하고 카메라 자세를 계산하여 집을 삽입한 결과를 보여준다. (그림 7)의 상단 그림은 72개의 특징점을 가진 큐브박스를 마커로 사용한 Test



(그림 6) 가상객체의 비디오 삽입 결과 : Test Set #3의 원본 프레임 (위), 가상객체가 삽입된 프레임 (아래).

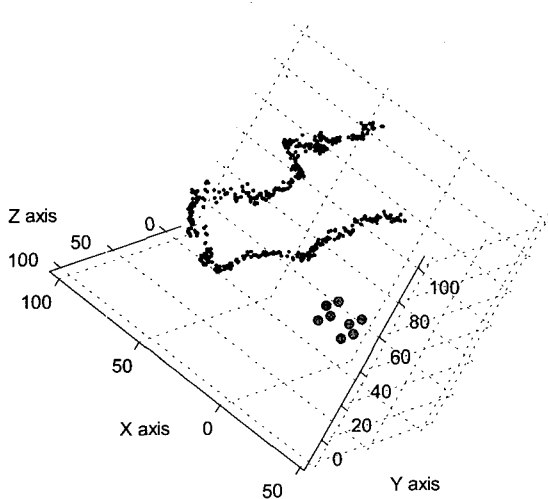


(그림 7) 가상객체의 비디오 삽입 : Test Set #4의 원본 프레임(위), 가상객체가 삽입된 프레임(아래).



(그림 8) 마커와 삽입된 가상객체: Test Set #3 (왼쪽), Test Set #4 (오른쪽).

Set #4의 비디오 프레임이다. 하단 그림은 그 마커의 특징점을 추적하고 카메라 자세를 계산하여 화분과 말을 합성한 결과이다. 여기서 삽입된 말은 처음 프레임에서 왼쪽의 화분에서 출발하여 마지막 프레임에 오른쪽 화분으로 이동하는 결과를 보여준다. (그림 8)은 (그림 6)에 보이는 Test Set #3의 마커와 마커위에 집이 합성된 영상을 확대하고 (그림 7)에 보이는 Test Set #4의 마커와 마커에 화분이 합성된 결과를 보여준다. (그림 9)는 (그림 6)에 보이는 Test Set #3의 507개의 프레임에 대한 카메라 궤적을 표현한 그림이다. 각 프레임에서의 카메라의 위치와 카메라의 방향을 표시하였다.



(그림 9) Test Set #3의 비디오 프레임에 대한 카메라 궤적

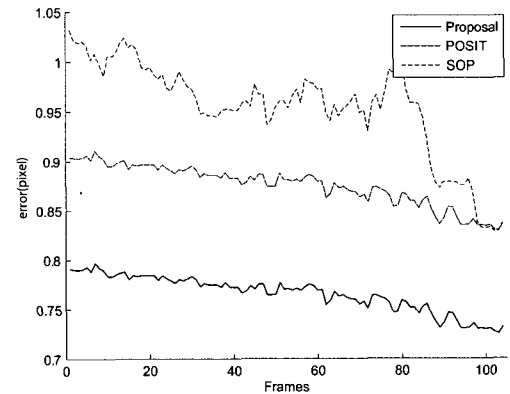
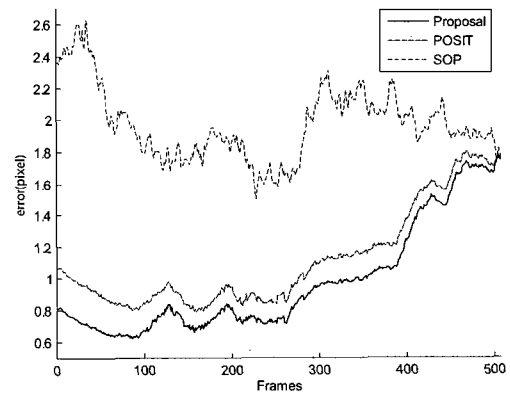
(그림 10)은 (그림 6)과 (그림 7)의 Test Set #3과 Test Set #4의 비디오에 대한 SOP와 POSIT 알고리즘과 본 논문에서 제안한 방법을 프레임별로 비교한 그래프이다. 본 논문에서 제안한 방법이 가장 정확한 것을 볼 수 있다. 에러를 구하는 식은 식 (11)과 같다. m_i 는 마커의 특징점이고

m_i' 은 카메라 자세를 계산하여 얻어진 회전행렬과 이동벡터로 재 투사하여 얻어진 점이다.

$$e = \frac{1}{n} \sum_{i=0}^{n-1} |m_i - m_i'| \tag{11}$$

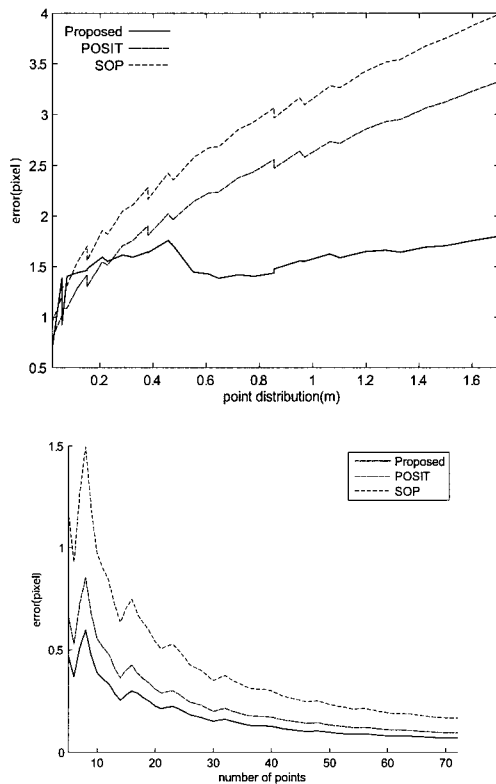
(그림 11)의 왼쪽 그림은 참조점을 기준으로 마커의 특징점의 분산에 따른 에러를 나타내는 그래프이다. 참조점과의 거리차이에 대한 분산이 커질수록 SOP와 POSIT은 에러가 크게 증가하지만 본 논문에서 제안한 방법은 거의 비슷한 수준을 유지하는 것을 볼 수 있다. 오른쪽 그림은 점의 수에 따른 에러 그래프로서 점의 수를 많이 사용할수록 더욱 정확한 결과를 얻을 수 있다는 것을 보여준다.

제안한 알고리즘은 특징점의 수 N 에 대해 $O(N^2)$ 만큼의 연산이 필요하다. 그러나 첫 번째 프레임에서 한번만 수행되는 참조점 설정을 위한 단계에서만 $O(N^2)$ 복잡도를 가지며 그 외의 단계에서는 $O(N)$ 의 복잡도를 가지므로 연산량의 큰 증가는 없다. 큰 연산량의 증가가 없는 것을 (그림 12)를 보면 알 수 있다. (그림 12)는 점의 수에 따른 연산 시간을 세 가지 방법에 대하여 비교한 그래프이다. 영상의 획득 및 마커에서의 특징점 탐색 시간을 제외한 카메라 자세의 추정시간만 표시하였다. 이 그래프에서 보면 SOP는

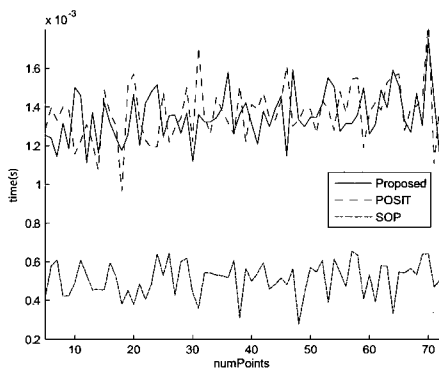


(그림 10) Test Set #3의 비디오 데이터에 대한 투사 에러 (위), Test Set #4의 비디오 데이터에 대한 투사 에러 (아래).

한 번의 연산으로 구하고 POSIT과 제안된 방법은 반복하여 계산을 하기 때문에 SOP보다 0.8 ms 정도 더 걸리는 것을 볼 수 있다. 제안된 방법이 POSIT 알고리즘과 같은 반복을 하지만 POSIT 알고리즘과 거의 같은 수행속도를 가진다. 기존의 방법보다 정확성이나 안정성 면에서 우수하면서 추가적인 계산시간이 거의 없음을 알 수 있다. (그림 10)과 (그림 11)에서 제안된 방법은 POSIT 알고리즘보다 정확성에서 우수한 것을 볼 수 있고 (그림 12)에서 POSIT 알고리즘과 제안된 방법의 연산시간이 거의 유사하게 나타나는 것을 볼 수 있다. 제안된 방법은 선형근사를 사용하여 POSIT 알고리즘만큼 빠르면서 안정된다는 것을 알 수 있다.



(그림 11) 에러 비교 결과 : 점의 분산에 대한 에러(왼쪽), 점의 수에 대한 에러(오른쪽).



(그림 12) 한 프레임에서의 점의 수에 따른 연산 속도 비교 결과.

5. 결론

본 논문은 모델의 기하구조를 가정하고 모델을 프레임에서 탐지하고 탐지된 특징점 위치를 사용하여 빠르게 카메라 자세를 추정하는 기법을 제안하였다. 단순한 형태의 선명한 패턴의 모델을 사용하면 특징점을 빠르게 탐색하고 추적할 수 있다. 단일 프레임에서의 탐색된 특징점으로부터 카메라의 위치와 방향을 초기치가 필요없는 선형 기법으로 빠르게 추정한다. 자세 추정 절차의 모든 추정식이 선형이고 초기 설정치가 필요하지 않기 때문에 자세 추정 과정은 실시간 시스템에 응용하기에 충분히 빠르다. 이전의 빠른 카메라 자세 추정 방법들과 비교하여 시간을 더 소비하지 않고도 카메라 자세의 정확성이 크게 증가하였다.

제안된 방법의 응용으로 실사 비디오 스트림에 가상 객체를 삽입하는 증강현실 응용을 구현하였다. 계산된 카메라 자세 파라미터를 사용하여 영상 프레임에서 탐지된 모델 위치에 가상 객체를 삽입하는 응용시스템을 구현하고 객체 삽입을 수행하였다. 실험 결과들은 제안된 카메라 자세 추정 방법이 뛰어나고 매우 빠르다는 것을 보여주었다.

향후 연구과제로 마커의 형태를 다양화시키고 또한 마커를 사용하지 않고 임의의 물체에 대해서도 카메라 추정이 가능한 방법을 연구할 필요가 있다. 또한 동일 평면상에 위치한 특징점들만을 사용하여 카메라 자세의 안정된 추정이 가능한 기법의 연구도 진행되어야 한다.

참고 문헌

- [1] S. Gibson, J. Cook, T. Howard, R. Hubbold, and D. Oram, "Accurate camera calibration for off-line, video-based augmented reality," Proceedings of International Symposium on Mixed and Augmented Reality, pp.37-46, 2002.
- [2] M. Lourakis and A. Argyros, "Camera Matchmoving in Unprepared, Unknown Environments," Computer Vision and Pattern Recognition, p.1190, 2005.
- [3] J. Heikkila, "Accurate camera calibration and feature based 3-D reconstruction from monocular image sequences," Ph.D. dissertation, University of Oulu, Finland, 1997.
- [4] M. Lourakis and A. Argyros, "Vision-Based Camera Motion Recovery for Augmented Reality," Computer Graphics International, pp.569-576, 2004.
- [5] Y.-K. Jung, K.-W. Lee, and Y.-S. Ho, "Feature-Based Object Tracking with an Active Camera," Proceedings of the Third IEEE Pacific Rim Conference on Multimedia, pp. 1137-1144, 2002.
- [6] G. Simon, A. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," Proceeding of IEEE and ACM International Symposium on Augmented Reality, pp.12-128, 2000.
- [7] S. Yamamoto and S. Hirai, "Robust and Video-frame Rate

Tracking of Planar Motion by Matched Filtering on FPGA," Proceeding of IEEE and International Conference on Robotics & Automation, pp.4707-4712, April, 2004.

[8] M. Pressigout and E. Marchand, "Real Time Planar Structure Tracking for Visual Servoing: a Contour and Texture Approach," IEEE Intelligent Robots and System, pp.251-256, August, 2005.

[9] K. Kutulakos and J. Vallino, "Calibration-free augmented reality," IEEE Transactions on Visualization and Computer Graphic, pp.346-359, October, 2000.

[10] H. Park and J.-I. Park, "Invisible Marker Tracking for AR," Third IEEE and ACM International Symposium on Mixed and Augmented Reality, pp.272-273, 2004.

[11] A. Ali and S. Afghani, "Support Vectors a Way to Adapt for Lane Marker Tracking: A Step towards Intelligent Transportation Systems," Proceeding of IEEE and International Conference on Emerging Technologies, pp. 151-156, September, 2005.

[12] X. Zhang, S. Fronz, and N. Navab, "Visual Marker Detection and Decoding in AR Systems: A Comparative Study," International Symposium on Mixed and Augmented Reality, p.97, 2002.

[13] R. Carceroni and C. Brown, "Numerical methods for model based pose recovery," Technical Report 659, The University of Rochester, New York, 1997.

[14] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization approach," International Journal of Computer Vision, pp.137-154, 1992.

[15] D. Oberkampf, D. Dementhon, and L. Davis, "Iterative pose estimation using coplanar feature points," Comput. Vis. Image Underst., pp.495-511, 1996.

[16] D. Dementhon and L. Davis, "Model-based object pose in 25 lines of code," International Journal of Computer Vision, pp.123-141, 1995.

[17] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.1330-1334, 2000.

[18] C. Poleman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.206-218, 1997.



박종승

e-mail : jong@incheon.ac.kr
 1992년 경북대학교 전자계산학과(이학사)
 1994년 포항공과대학교 컴퓨터공학과(공학석사)
 1999년 포항공과대학교 컴퓨터공학과(공학박사)
 1999년~2001년 한국전자통신연구원
 가상현실연구부 선임연구원

2001년~2003년 디지털아리아 시각정보처리연구소 책임연구원
 2004년~현재 인천대학교 컴퓨터공학과 조교수
 관심분야: 영상처리, 3D 비전, 증강현실, 게임공학



이범종

e-mail : leeyanga@incheon.ac.kr
 2006년 인천대학교 컴퓨터공학과(공학사)
 2006년~현재 인천대학교 컴퓨터공학과
 석사과정

관심분야: 동작기반 인터페이스, 증강현실,
 가상현실