

2 차원 영상에서 패턴매칭을 이용한 3 차원 물체의 변환정보 추정

조택동#, 이호영*, 양상민**

The Estimation of the Transform Parameters Using the Pattern Matching with 2D Images

Taik Dong Cho[#], Ho Young Lee^{*} and Sang Min Yang^{**}

ABSTRACT

The determination of camera position and orientation from known correspondences of 3D reference points and their images is known as pose estimation in computer vision or space resection in photogrammetry. This paper discusses estimation of transform parameters using the pattern matching method with 2D images only. In general, the 3D reference points or lines are needed to find out the 3D transform parameters, but this method is applied without the 3D reference points or lines. It uses only two images to find out the transform parameters between two image. The algorithm is simulated using Visual C++ on Windows 98.

Key Words : Pattern matching (패턴 매칭), Point matching (점 매칭), Transform parameters (변환정보), Pose estimation (자세 추정)

기호설명

- $p(u, v, w)$ = point coordinate in 3D space
- $q(x, y)$ = point coordinate in image plane
- S = scale factor
- r_{ij} = element of rotation matrix
- t_x, t_y, t_z = translation parameter to each axis
- α, β, γ = rotation parameter to each axis
- c_1, c_2, c_3 = $\cos\alpha, \cos\beta, \cos\gamma$
- s_1, s_2, s_3 = $\sin\alpha, \sin\beta, \sin\gamma$
- (x^*, y^*) = point in image with noise

- m = noise level
- ξ = uniformly distributed noise with zero mean

1. 서론

2 차원 영상과 3 차원 공간상에 대응되는 점들을 이용하여 카메라의 위치를 결정하는 것은 영상 처리 분야에서는 매우 일반적인 문제이며, 위치 정보를 알고있는 3 차원 점 또는 직선과 이것이 카메라에 맺히는 2 차원 정보간의 관계를 이용하여 카메라의 위치 또는 물체 변환 정보를 추출하

접수일: 2003 년 8 월 4 일; 게재승인일: 2004 년 4 월 9 일

교신저자: 충남대학교 기계설계공학과
E-mail: tdcho@cnu.ac.kr Tel. (042) 821-6648

* 충남대학교 기계설계공학과 대학원

** (주)한빛파워서비스 기술연구소

게 된다. 이러한 경우 2 차원 영상에 나타나는 점들의 공간상의 3 차원 정보를 알고 있어야만 3 차원 변환 정보를 추출할 수가 있다. 이러한 방법은 항공기의 위치 추적, 로봇의 보정 등에 사용되고 있는 방법이기도 하다.

Liu¹ 등은 2 차원 영상과 대응되는 3 차원 점 또는 직선을 이용하여 카메라의 위치를 결정하는 것에 대한 방법을 제시하고 컴퓨터 시뮬레이션으로 검증하였으며, Quan 과 Lan²은 n 개의 점을 이용하여 카메라의 위치를 결정하는 방법을 다차원 다항식의 구속조건을 이용하는 방법에 대하여 연구하였다. 또한 Lu³ 등은 비디오 영상을 이용하여 위치 결정을 빠르게 수행할 수 있도록 하는 방법에 대하여 연구하였으며, Ansar 과 Daniilidis⁴는 점과 직선을 이용한 자세 추정에 있어서 반복적이지 않은 추정방법에 대하여 연구하였으며, 오차에 대한 민감도를 해석하였다. 그 외에도⁵⁻⁹ 2 차원 영상과 3 차원 점 또는 직선의 대응관계를 이용하여 카메라의 자세를 결정하는 것과 물체의 자세를 결정하는 것에 대한 연구가 많이 진행되어 왔다.

본 연구에서는 2 차원 영상에 맺히는 점들에 대한 3 차원 정보 없이 점 패턴을 포함하는 두 개의 영상만을 가지고 3 차원 변환 정보를 추출하는 방법에 대하여 제시하고 시뮬레이션의 결과를 제시한다. 여기서는 몇 가지 가정이 존재하며, 이 가정은 먼저 영상을 얻는 카메라를 핀홀 카메라로 가정하였으며, 패턴은 2 차원 평면 패턴으로 가정한다. 이러한 방법은 영상에 대응되는 3 차원 점을 알 수 없는 경우에 대략적인 물체의 변환에 대한 정보를 추정하여야 할 경우에 이용될 수 있을 것이다.

2. 원근변화법 및 3 차원 변환 정보

2.1 원근변화법

본 연구에서는 서로 다른 방향에서 맺혀지는 한 개의 패턴 영상을 이용하여 물체의 변환 정보를 추출하게 된다. 그러므로 먼저 3 차원 물체가 카메라 영상에 맺히는 원리에 대하여 먼저 살펴본다. 카메라를 핀홀형으로 가정하면, 다음의 Fig. 1 과 같은 3 차원 공간상의 점과 영상에서의 위치에 대한 관계를 찾을 수 있다.¹⁰ 3 차원 공간상의 점이 카메라를 통하여 영상면으로 맺히게 되면 3 차원 정보 중 Z 축 방향의 거리에 대한 정보를 소실하

게 되어 3 차원 변환 정보를 찾기가 어렵게 되는 것이다. 이러한 거리 정보는 카메라 영상에서 영상의 크기를 변화시키게 된다. 영상에 맺히는 2 차원 정보 또한 거리 정보에 의해 영향을 받게 된다. 일반적으로 알려진 3 차원 위치의 2 차원 영상으로의 매핑 관계는 다음의 식(1)과 같이 나타난다. 여기서 λ는 카메라 렌즈에서부터 영상면까지의 거리이다.

$$\begin{aligned} u &= \frac{-x(w-\lambda)}{\lambda} \\ v &= \frac{-y(w-\lambda)}{\lambda} \end{aligned} \quad (1)$$

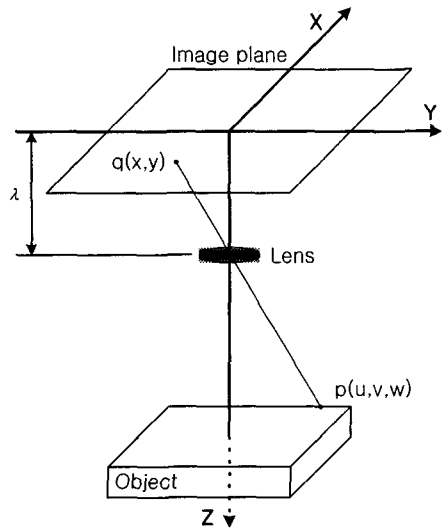


Fig. 1 The basic principle of the perspective projection

2.2 3 차원 변환정보

일반적으로 3 차원 점의 회전 및 이동변환은 다음의 식(2)와 같은 행렬로 나타낼 수 있다. 이는 3 차원 공간 상에서 물체의 회전과 이동을 나타내는 것이다. 첨자 b 는 회전변환 전(첫번째 매칭점들)을 의미하며, 첨자 a 는 회전변환 후(두 번째 매칭점들)을 의미한다.

$$\begin{bmatrix} u_a \\ v_a \\ w_a \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_b \\ v_b \\ w_b \\ 1 \end{bmatrix} \quad (2)$$

카메라에 의하여 얻어지는 영상은 3 차원 좌표 상의 점을 2 차원 평면으로 원근투영하게 되므로 3 차원의 Z 축에 대한 정보를 잃어버리게 된다. 본 논문에서는 Z 축에 대한 정보가 없더라도 패턴의 모양변화를 이용하여 3 차원 회전 및 이동정보를 찾아내는 방법에 대하여 제안한다. 식(2)에 원근투영에 대한 식(1)을 적용하면 다음의 식(3)을 얻을 수 있다.

$$\begin{bmatrix} x_a(\lambda - w_a) \\ \lambda \\ y_a(\lambda - w_a) \\ \lambda \\ w_a \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_b(\lambda - w_b) \\ \lambda \\ y_b(\lambda - w_b) \\ \lambda \\ w_b \\ 1 \end{bmatrix} \quad (3)$$

위의 식(3)을 이용하여 회전과 이동에 대한 변환정보를 추출하게 되는데 실제적으로 영상은 원근변화법에 의하여 Z 축 방향에 대한 정보를 잃어버리게 된다. 즉, w_a 와 w_b 값을 알 수가 없어서 위의 식을 풀 수가 없게 되는 것이다. 이 식을 풀기 위하여 몇 가지 가정을 하였다. 먼저 본 논문에서 제안한 방법은 첫 번째 영상을 기준으로 하여 두 번째 영상의 변화로부터 변환정보를 추출하므로 첫 번째 영상이 기준이 된다고 할 수 있다. 그래서 첫 번째 패턴 영상에서 추출된 점들은 3 차원 영상에서 동일한 높이 값을 가진 점들에 의해서 얻어진 점이라는 가정이 필요하다. 변환정보는 첫 번째 패턴과 두 번째 패턴 사이의 상대적인 변환을 찾는 것이므로 첫 번째 패턴에서의 높이 값은 일정하다는 가정에 의하여 w_b 는 영(zero)으로 볼 수 있다. 그러나 두 번째 패턴에서는 변환 후의 패턴이므로 w_a 의 값은 추출된 점마다 서로 틀리게 된다. 여기서 w_a 는 원근변화법에서 Z 축 방향의 값이 영상의 X, Y 축의 값에 영향을 미치는 것으로 볼 수 있으며 $\lambda/(\lambda - w_a)$ 는 영상에 크기변환 요소에 해당한다고 볼 수 있다. 그리하여 식(3)에서 $\lambda/(\lambda - w_a)$ 를 크기요소(scale factor)인 S 로 놓고 $w_b=0$ 을 적용하면 다음의 식(4)가 얻어진다.

$$\begin{aligned} x_a &= S \cdot r_{11}x_b + S \cdot r_{12}y_b + S \cdot t_x \\ y_a &= S \cdot r_{21}x_b + S \cdot r_{22}y_b + S \cdot t_y \end{aligned} \quad (4)$$

여기서

$$\begin{aligned} r_{11} &= c_2c_3 \\ r_{12} &= -c_2s_3 \\ r_{21} &= s_1s_2c_3 + c_1s_3 \\ r_{22} &= c_1c_3 - s_1s_2s_3 \end{aligned} \quad (5)$$

3. 3 차원 변환정보의 추출

3.1 회전변환 정보의 추정

먼저 3 차원 변환정보 중 회전정보에 대한 것을 추정하는 다음 이동변환 정보를 추정하게 된다. 식(4)를 변형하여 행렬형태로 나타내면 식(6)과 같이 나타낼 수 있다.

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = \begin{bmatrix} S \cdot r_{11} & S \cdot r_{12} \\ S \cdot r_{21} & S \cdot r_{22} \end{bmatrix} \begin{bmatrix} x_b \\ y_b \end{bmatrix} + \begin{bmatrix} S \cdot t_x \\ S \cdot t_y \end{bmatrix} \quad (6)$$

여기서 첫 번째 및 두 번째 영상에서 얻어진 n 개의 매칭점들을 $P_{bi}=[x_{bi}, y_{bi}]^T$, $P_{ai}=[x_{ai}, y_{ai}]^T$, ($i=1, \dots, n$)라 하고, 구하고자 하는 것이 $S t_x$, $S t_y$, $S r_{11}$, $S r_{12}$, $S r_{21}$, $S r_{22}$ 이므로 이들을 열행렬로 식(7)과 같이 나타내면, 식(8)을 얻을 수 있다.

$$\begin{aligned} T_r &= [S t_x \quad S t_y \quad S r_{11} \quad S r_{12} \quad S r_{21} \quad S r_{22}]^T \\ &= [t_x' \quad t_y' \quad r_{11}' \quad r_{12}' \quad r_{21}' \quad r_{22}']^T \end{aligned} \quad (7)$$

$$P_{ai} = C_i \cdot T_r \quad (8)$$

여기서

$$C_i = \begin{bmatrix} 1 & 0 & x_{bi} & y_{bi} & 0 & 0 \\ 0 & 1 & 0 & 0 & x_{bi} & y_{bi} \end{bmatrix}$$

이다. 식(8)을 오차식(E)로 나타내면, 식(9)를 얻을 수 있으며 이 식을 n 개의 매칭 쌍에 대하여 적용하면 식(10)을 얻을 수 있다. 식(10)을 보면, $n \geq 3$ 인 경우 $\text{rank}(C)=12$ 가 되므로 T_r 행렬을 계산할 수 있게 된다. 오차식(E)를 최소가 되도록 최소자승 근사법을 적용하여 식(11)과 같은 방법으로 계산하게 된다.¹¹

$$E_i = C_i \cdot T_r - P_{ai} \quad (9)$$

$$E = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix} = \begin{pmatrix} C_1 T_r - P_{a1} \\ C_2 T_r - P_{a2} \\ \vdots \\ C_n T_r - P_{an} \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix} T_r - \begin{pmatrix} P_{a1} \\ P_{a2} \\ \vdots \\ P_{an} \end{pmatrix} \quad (10)$$

$$= C \cdot T_r - P_a$$

$$T_r = [C^T C]^{-1} C^T P_a \quad (11)$$

먼저 식(11)에서 구한 파라미터 중 r_{11}' , r_{12}' , r_{21}' , r_{22}' 와 식(5)를 이용하여 정리하면 다음과 같은 식(12)와 식(13)을 얻을 수 있다. 식(12)를 이용하여 Z 축 방향 회전값인 γ 를 계산할 수 있다.

$$\gamma = \tan^{-1} \left(-\frac{r_{12}'}{r_{11}'} \right) \quad (12)$$

$$\frac{s_1 s_2}{c_1} = \frac{r_{21}' c_3 - r_{22}' s_3}{r_{21}' s_3 + r_{22}' c_3} = A$$

$$S \cdot c_1 = \frac{r_{22}' s_3 + r_{21}' c_3}{c_3} = B \quad (13)$$

$$S \cdot c_2 = \frac{r_{11}'}{c_3} = -\frac{r_{12}'}{s_3} = C$$

식(13)을 정리하면 식(14)와 같이 크기요소(S)를 계산할 수 있으며 여기서 계산된 S를 이용하여 X와 Y 축 중심 회전 값인 α 와 β 를 식(15)와 같이 계산할 수 있다.

$$S = \sqrt{\frac{(B^2 + C^2 \cdot A^2 \cdot B^2) \pm \sqrt{D}}{2}} \quad (14)$$

$$D = (B^2 + C^2 \cdot A^2 \cdot B^2)^2 - 4(B^2 \cdot C^2)$$

$$\alpha = \cos^{-1} \left(\frac{B}{S} \right), \quad \beta = \cos^{-1} \left(\frac{C}{S} \right) \quad (15)$$

여기서 크기요소는 항상 양의 값을 가지게 되며, α 와 β 는 코사인 값에 의하여 구해진 것이기 때문에 항상 양의 값을 가지게 된다. 즉, 항상 양의 방향으로만 회전되는 것처럼 생각된다는 것이다. 이를 해결하기 위해서 α 와 β 의 회전방향을 찾

는 것이 필요하다. 또한 여기서 구한 회전변환 α , β , γ 는 w_a 값을 정확히 몰라서 크기요소로 단순히 놓고 계산하였으므로 오차를 수반하게 된다. 이러한 오차를 줄이기 위하여 계산되어진 회전변환 값을 이용하여 첫 번째 매칭점들을 회전 변환하여 발생하는 높이 값으로 두 번째 매칭점들을 원근변화법에 의하여 발생하는 오차를 보정해준 다음 같은 방법으로 다시 회전변환을 계산하였다. 이러한 방법으로 회전변환 정보를 추정해나가는 방법을 적용하였다.

3.2 회전정보의 회전방향 결정

앞에서 언급했듯이 계산되어진 α 와 β 는 코사인 값에 의하여 구해졌기 때문에 방향성을 갖지 못한다. 일반적으로 Fig. 2에서 보는 것처럼 물체를 회전시키더라도 평면투사의 경우에는 회전 방향을 구분할 수가 없다. 그러나 카메라 영상은 원근투사이므로 영상에 변화가 나타나게 된다. 이를 이용하여 α 와 β 의 회전 방향을 결정할 수 있으며, 본 논문에서는 방향의 결정은 모두 4가지의 경우가 발생할 수 있으며 이 4가지의 경우를 앞에서 구해진 α , β , γ 를 적용하여 첫 번째 패턴점을 회전 변환한 후 두 번째 패턴점과의 거리오차를 계산하여 오차가 최저로 나타나는 방향을 α 와 β 의 회전 방향으로 결정하였다. Fig. 3은 회전방향을 결정하는 알고리즘을 나타낸 것이다.

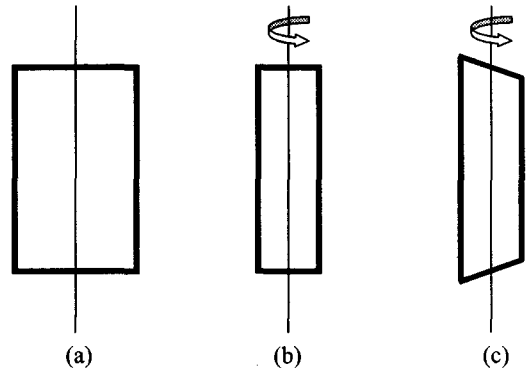


Fig. 2 Image deformation by rotation : (a) camera image before transformation, (b) projection image on plane after transformation and (c) camera image after transformation

1. case $\alpha > 0$ & $\beta > 0$,
estimate α, β, γ and calculate distance error e_1
2. case $\alpha < 0$ & $\beta < 0$,
estimate α, β, γ and calculate distance error e_2
3. case $\alpha > 0$ & $\beta < 0$,
estimate α, β, γ and calculate distance error e_3
4. case $\alpha < 0$ & $\beta > 0$,
estimate α, β, γ and calculate distance error e_4
5. find out minimum error e
6. determine the α & β rotation direction

Fig. 3 The algorithm to determine rotation direction

3.3 이동변환 정보의 추정

앞에서 추정되어진 α, β, γ 를 이용하여 이동변환에 대한 파라미터를 추정하게 된다. 식(2)에 식(1)을 적용하고, $w_b = 0$ 이라고 가정하였으므로 다음의 식(16)을 얻을 수 있다.

$$\begin{aligned} x_a &= \frac{\lambda(r_{11}x_b + r_{12}y_b + t_x)}{\lambda - r_{31}x_b - r_{32}y_b - t_z} \\ y_a &= \frac{\lambda(r_{21}x_b + r_{22}y_b + t_y)}{\lambda - r_{31}x_b - r_{32}y_b - t_z} \end{aligned} \quad (16)$$

위의 식(16)을 정리하여 행렬로 나타내면 식(17)과 같이 되고, 이를 오차식(E_i)로 나타내면 식(18)과 같이 된다.

$$\begin{aligned} B_i &= A_i \cdot T_i \\ B_i &= \begin{bmatrix} (r_{31}x_{bi} + r_{32}y_{bi})x_{ai} \\ (r_{31}x_{bi} + r_{32}y_{bi})y_{ai} \end{bmatrix} \\ A_i &= \begin{bmatrix} x_{ai} - r_{11}x_{bi} - r_{12}y_{bi} & -1 & 0 & -x_{ai} \\ y_{ai} - r_{21}x_{bi} - r_{22}y_{bi} & 0 & -1 & -y_{ai} \end{bmatrix} \\ T_i &= \begin{bmatrix} \lambda & \lambda \cdot t_x & \lambda \cdot t_y & t_z \end{bmatrix}^T \\ E_i &= A_i \cdot T_i - B_i \end{aligned} \quad (17) \quad (18)$$

앞에서 회전변환 정보를 추정했던 것과 마찬가지로 n 개의 매칭점에 대하여 적용하면 식(19)를 얻을 수 있으며, 이를 최소자승 근사법으로 식(20)과 같이 계산하면 T_i 행렬의 값을 구할 수 있다.

$$E = A \cdot T_i - B \quad (19)$$

$$T_i = [A^T A]^{-1} A^T B \quad (20)$$

이렇게 계산되어진 T_i 행렬에 의하여 이동변환 파라미터 t_x, t_y, t_z 에 대한 값을 구할 수 있다.

4. 실험 및 결과

제안된 알고리즘을 시뮬레이션을 통하여 실험하였다. 모든 시뮬레이션은 Windows 98 상에서 Visual C++를 이용하였다.

4.1 시뮬레이션

무작위로 생성된 점을 이용하여 변환파라미터 값을 변화시켜가면서 시뮬레이션을 수행하였다. 가상 카메라는 핀홀형으로 가정하고 λ 는 1000으로 하였다. 각각의 영상에서 서로 대응되는 점들은 아는 것으로 가정하였으며, 점의 개수를 4개에서부터 10개까지 변화시켜가면서 수행하였다. 또한 회전변환 파라미터 값은 α 와 β 는 $-60^\circ, -30^\circ, 30^\circ, 60^\circ$ 의 4가지 γ 는 $-75^\circ, -30^\circ, 30^\circ, 75^\circ$ 의 4가지 각도에 대하여 이동변환 파라미터 값은 $-30, 0, 30$ 의 3가지에 대하여 수행하였다. 영상은 일반적으로 잡음을 가지게 되므로 잡음의 영향을 살펴보기 위하여 픽셀단위의 균일분포 잡음을 식(21)과 같이 추가하여 시뮬레이션을 수행하였다.¹

$$\begin{aligned} x^* &= x + m\xi \\ y^* &= y + m\xi \end{aligned} \quad (21)$$

여기서 (x, y) 는 오차가 없는 영상의 점이며, (x^*, y^*) 는 오차가 추가된 영상의 점이다. m 은 픽셀단위의 잡음레벨을 나타내는 값이며 0부터 2까지 0.5 픽셀단위의 값으로 변화시켜 가면서 잡음을 추가하였으며, ξ 는 구간 $[-0.5, 0.5]$ 의 평균이 0인 균일분포 잡음을 나타낸다. 위와 같은 방법으로 총 60480 번의 시뮬레이션을 수행하였다.

4.2 실험

간단한 패턴을 만들고 CCD 카메라를 이용하여 얻어진 영상에서 특징점을 추출하여 변환정보를 추출하는 실험을 수행하였다. 먼저 카메라 보정작업을 통하여 실험에 필요한 λ 를 근사하여 구한 다

음 이를 적용하여 실험을 하였다. 계산된 λ 는 2274.7 이다. 실험은 패턴으로부터 얻어진 10 개의 점을 이용하여 하였으며, 실험은 총 4 가지에 대하여 수행하였다. 본 실험에서 시스템의 특성상 회전 변환을 정확한 값으로 수행할 수 없어서 정확

한 오차를 계산하지는 못하였다. Fig. 4 는 실험에 사용된 패턴을 나타내며 Fig. 4(a)는 패턴의 이동과 회전을 하기 전이며 (b)는 회전 후에 카메라에 나타나는 영상이다.

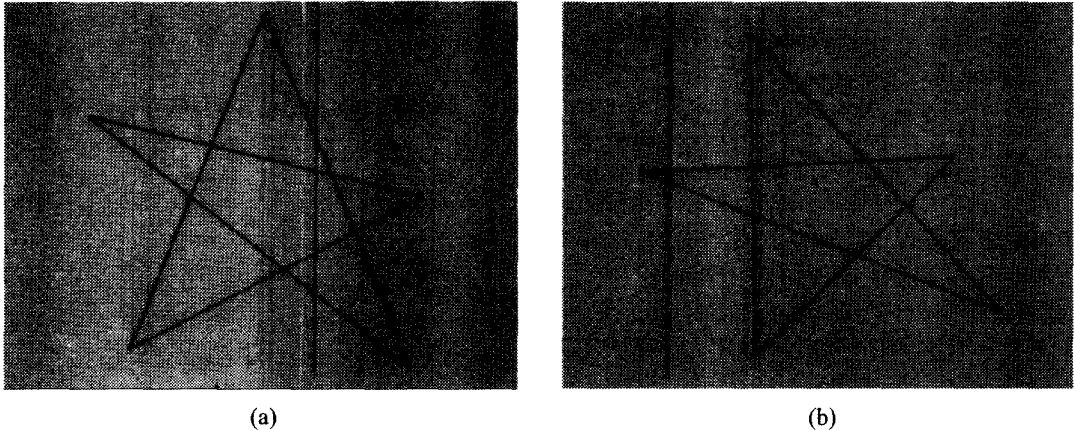


Fig. 4 The camera images for experiment: (a) first pattern image, (b) transformed image of first pattern

4.3 결과 및 고찰

위에서 설명한 방법으로 시뮬레이션을 수행하여 오차를 RMS 오차로 나타내었다. Fig. 5 는 컴퓨터 시뮬레이션을 위하여 무작위로 생성된 10 개의 점을 변환전 후에 따라 나타낸 것이다. Fig. 5(a)는 첫 번째 패턴 즉, 변환전의 영상을 나타내며, (b)는 두 번째 패턴으로 $\alpha=30^\circ$, $\beta=-60^\circ$, $\gamma=75^\circ$, $t_x=0$, $t_y=0$,

$t_z=30$ 로 변환 후 얻어진 영상 데이터이며, (c)는 앞의 두 패턴에 의하여 제시된 알고리즘으로 계산된 변환 파라미터 값을 이용하여 계산된 두 번째 패턴이다. 계산된 파라미터에 의하여 변환된 영상은 두 번째 영상과 거의 일치하는 것을 볼 수 있으며 두 개의 영상이 서로 매칭됨을 알 수 있다.

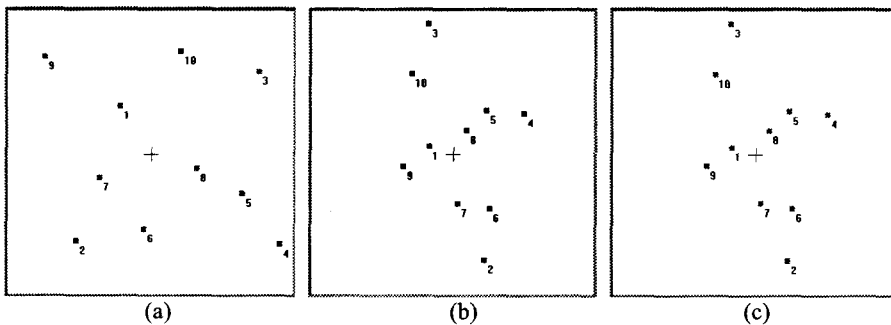


Fig. 5 The generated points for simulation: (a) first pattern, (b) second pattern, and (c) transformed image of first pattern with estimated transformation parameters($\alpha=30^\circ$, $\beta=-60^\circ$, $\gamma=75^\circ$, $t_x=0$, $t_y=0$, $t_z=30$)

Fig. 6 은 잡음의 정도에 따른 변환 파라미터의 RMS 오차를 그래프로 나타낸 것이다. Fig. 6(a)~(f) 는 각각 5 개부터 10 개의 점을 이용하여 시뮬레이

션한 결과로 잡음이 증가함에 따라 오차가 증가하는 것을 볼 수 있다. 전체적인 시뮬레이션의 결과를 보면 Z 축 중심 회전값인 γ 는 다른 오차에 비

해서 적게 나타나는 것을 볼 수 있다. 이는 다른 파라미터를 구하기 전에 가장 먼저 γ 의 값을 먼저 구하게 되므로 다른 파라미터는 γ 의 오차가 반영되기 때문에 그러한 것으로 생각된다. 또한 γ 는 회전방향을 따로 결정하지 않더라도 계산에 의하여 구할 수 있는 반면 α 와 β 는 회전 방향을 결정해 주어야 한다. 또한 이동변환 값은 회전변환 값을 추출한 다음에 계산이 되며, 특히 Z 축 방향으로의 이동변환은 α 와 β 의 회전에 의해서도 발생

할 수 있는 관련된 값이므로 α 와 β 의 오차가 영향을 미치게 되어 Z 축 방향으로의 이동인 t_z 의 오차가 다른 파라미터의 오차보다 크게 나타나는 것으로 생각된다. 원근변화법의 특성상 Z 축 방향으로의 작은 이동은 2 차원 영상에 큰 변화를 발생시키지 않기 때문에 이를 찾아내기가 힘든 것이다. Table 1 은 시뮬레이션에 의하여 계산되어진 RMS 오차 데이터의 일부를 표로 나타낸 것이다.

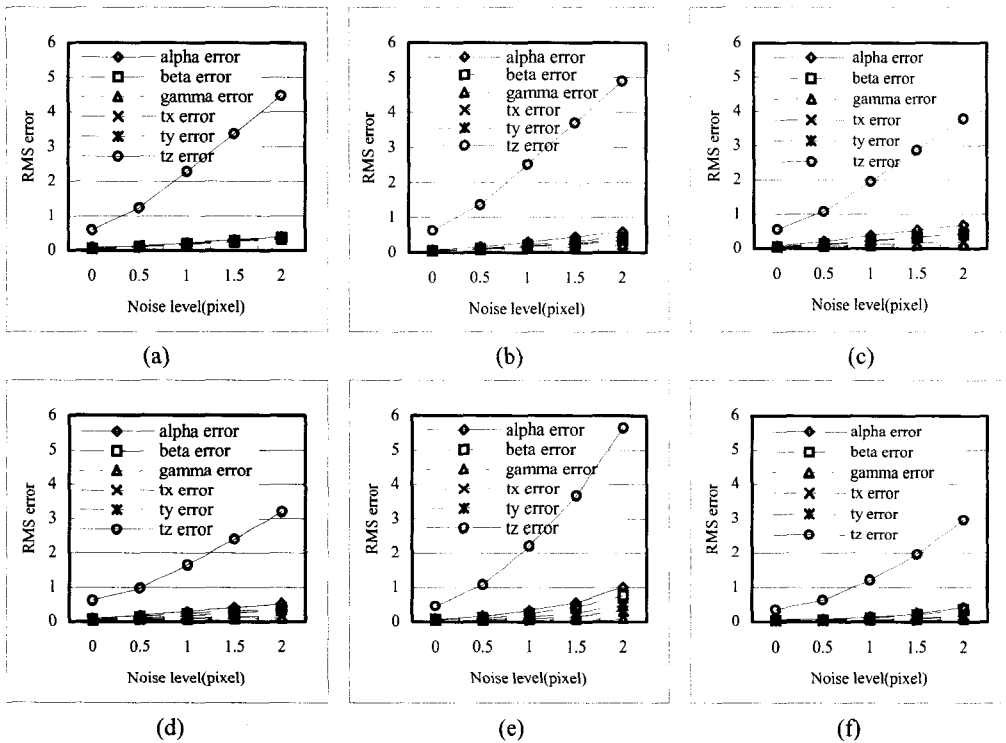


Fig. 6 RMS error according to noise level: (a) with 5 points, (b) with 6 points, (c) with 7 points, (d) with 8 points, (e) with 9 points, and (f) with 10 points

Table 1 RMS error of transform parameter according to noise level and points

No.	4					7					10					
	m	0.0	0.5	1.0	1.5	2.0	0.0	0.5	1.0	1.5	2.0	0.0	0.5	1.0	1.5	2.0
R	α	0.074	0.159	0.292	0.430	0.569	0.077	0.211	0.382	0.542	0.690	0.070	0.089	0.148	0.251	0.424
	β	0.060	0.110	0.194	0.283	0.373	0.042	0.132	0.238	0.334	0.420	0.045	0.065	0.121	0.216	0.368
M	γ	0.066	0.120	0.210	0.306	0.403	0.063	0.067	0.079	0.096	0.115	0.043	0.047	0.056	0.070	0.086
S error	t_x	0.042	0.088	0.161	0.236	0.312	0.041	0.073	0.119	0.163	0.203	0.019	0.028	0.052	0.090	0.156
	t_y	0.029	0.093	0.179	0.267	0.355	0.050	0.116	0.216	0.322	0.436	0.023	0.035	0.064	0.113	0.195
	t_z	0.481	1.422	2.752	4.102	5.453	0.556	1.086	1.954	2.864	3.787	0.345	0.641	1.214	1.962	2.971

Fig. 7 은 카메라 영상정보로부터 추출한 점들을 이용하여 변환 파라미터를 계산하고 첫 번째 패턴을 이 계산된 파라미터를 이용하여 변환하여 두 번째 패턴과 비교하여 본 것으로 이를 이용하여 계산된 변환 파라미터의 검증을 수행하였다. Table 2 는 4 가지 실험에서 얻어진 변환 파라미터의 값을 보여준다. 실험과정에서 패턴을 변환하는 파라미터의 값을 정확하게 적용할 수 없는 관계로 오차의 비교는 할 수 없었으나 비슷한 결과를 얻을 수 있었다. 1 번 실험의 경우는 회전정보는 없이 X 축과 Y 축 방향으로만 이동변환을 주어 본 것으로 이동변환의 값은 거의 유사한 값을 보이고 있으나, 회전변환의 값이 오차가 많이 발생하는 것으로 나타났다. 실험 2 는 X 축 중심회전을 한 경우이며, 실험 3 은 Y 축 중심회전을 한 경우로, 정확한 오차는 계산할 수 없지만, 방향과 그 크기가 비슷하게 나타났다. 이는 계산된 변환 정보를 이용하여 첫 번째 패턴을 변환한 경우 두 번째 패턴과 거의 일치하게 나타나는 것으로 확인하였다. 실험 4 의 경우에도 모든 방향에 대한 회전정보를 준 경우이며 이 경우에도 계산된 값을 이용하여 첫 번째 패턴을 변환한 경우 Fig. 7 (c)에 보는 것

처럼 Fig. 7(b)와 거의 일치하는 모습을 보여준다. 시뮬레이션의 경우보다 실제의 경우 많은 오차를 발생하였으며, 이는 영상으로부터 점을 추출하는 과정에서 발생하는 오차와 카메라 보정 작업에 의하여 구하여진 λ 에 의한 오차에 의한 것으로 생각된다. 실제로 렌즈 카메라의 경우에는 위치에 따라 λ 가 다르게 발생하게 되는데 이를 하나의 근사값으로 근사하여 사용함으로써 발생하는 오차의 영향으로 인하여 변환 파라미터의 계산오차가 발생한 것으로 사료된다.

Table 2 Experiment results

No.	1	2	3	4
α	9.53	0.69	43.89	21.75
β	-1.38	-30.70	0.01	-11.83
γ	0.26	6.71	2.51	20.90
t_x	-38.19	-19.71	-19.27	-8.60
t_y	40.01	6.95	-29.20	-1.10
t_z	-1.21	-79.26	-81.87	-44.33

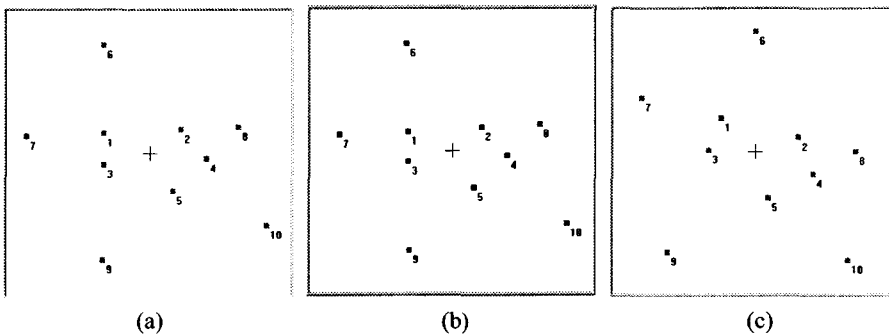


Fig. 7 The extracted points from image: (a) first pattern, (b) second pattern, and (c) transformed image of first pattern with estimated transformation parameters($\alpha=21.75^\circ$, $\beta=-11.83^\circ$, $\gamma=20.90^\circ$, $t_x=-8.6$, $t_y=-1.1$, $t_z=-44.33$)

5. 결론

본 논문에서는 두 개의 패턴 영상을 이용하여 변환정보를 추정하는 방법을 제시하고 이를 Visual C++을 이용하여 컴퓨터 시뮬레이션을 통하여 검증하여 보았다. 영상은 카메라의 원근변화법에 의하여 일반적인 투사변환과는 다르다는 점을 이용하여 2 차원 영상을 가지고 3 차원 변환정보를 추

정하는 방법을 제안하였다. 카메라는 이상적인 핀홀 카메라로 가정하였으며, 실제의 렌즈형 카메라에서는 카메라 렌즈의 가공오차나 왜곡현상에 의하여 시뮬레이션 결과보다 오차가 심하게 발생할 것으로 생각된다. 그러나 카메라 보정 작업이 정확히 수행되어 λ 에 해당되는 값을 계산할 수 있다면 변환 파라미터의 추정이 가능하리라 생각되며, 어느 정도의 오차에 대해서도 근사한 값을 계산할

수 있다. 시물레이션 결과에서 볼 수 있듯이 최소 4 개의 점 이상에 대한 대응되는 점들에 관한 영상의 정보가 있으면 이들의 변환관계를 추정할 수 있었으며, 잡음을 추가한 경우에도 오차가 있기는 하지만 근사한 값을 추정할 수 있었다. 일반적으로 카메라나 물체의 위치를 결정하는 기존의 방법들은 3 차원 대응점 또는 대응 직선을 이용한 방법들이 주를 이루고 있으나 본 논문에서 제시한 방법은 3 차원 대응점 또는 직선 없이 2 차원 영상의 데이터만을 가지고 변환정보를 추출할 수 있는 방법을 제시하였으며 시물레이션을 통하여 가능성을 보였다. Z 축 방향으로의 이동변환이 값의 오차가 크게 나타나기는 했지만 다른 파라미터에 대해서는 잡음의 영향에도 불구하고 근사한 값이 계산되는 것을 볼 수 있었다.

또한 실제 실험에서도 시물레이션의 경우보다는 큰 오차를 나타내기는 하였으나 정확한 추정이 아닌 근사적인 추정이면 가능한 응용분야에 적용이 가능하리라 생각된다. 향후 연구과제로는 제안된 알고리즘을 충분한 실제 실험을 통하여 수정 보완하여야 할 것이며, 회전변환의 경우 작은 회전각도를 갖는 경우에는 회전 파라미터 계산의 오차가 큰 것으로 나타나므로 이에 대한 민감도를 실험을 통하여 알아보고 개선방법에 대한 연구가 필요할 것이다. 또한 잡음이 심한 경우에는 회전 방향을 잘 못 찾는 경우가 발생할 수 있으므로 좀더 강인한 회전변환 방향 추정 알고리즘에 대한 연구가 필요하리라 생각된다.

참고문헌

1. Liu, Y., Huang, T. S. and Faugeras, O. D., "Determination of Camera Location from 2-D to 3-D Line and Point Correspondences," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp. 28-37, 1990.
2. Quan, L. and Lan, Z., "Linear N-Point Camera Pose Determination," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 8, pp. 774-780, 1999.
3. Lu, C. P., Hager, G. D. and Mjolsness, E., "Fast and Globally Convergent Pose Estimation from Video Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 6, pp. 610-616, 2000.
4. Ansar, A. and Daniilidis, K., "Linear Pose Estimation from Points and Lines," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, pp. 578-589, 2003.
5. Dornaika, F. and Garcia, C., "Pose Estimation using Point and Line Correspondences," Real-Time Imaging, Vol. 5, pp. 215-230, 1999.
6. Joseph, S. H., "Optimal Pose Estimation in Two and Three Dimensions," Computer Vision and Image Understanding, Vol. 73, No. 2, pp. 215-231, 1999.
7. Liu, M. L. and Wong, K. H., "Pose estimation using four corresponding points," Pattern Recognition Letters, Vol. 20, pp. 69-74, 1999.
8. Or, S. H., Luk, W. S., Wong, K. H. and King, I., "An efficient iterative pose estimation algorithm," Image and Vision Computing, Vol. 16, pp. 353-362, 1998.
9. Wlczek, P., Maccato, A. and deFigueiredo, R. J. P., "Pose Estimation of Three-Dimensional Objects from Single Camera Images," Digital Signal Processing, Vol. 5, pp. 176-183, 1995.
10. Lee, H. K., Kim, S. H., Choi, S. K. and Lee, S. R., "The Optimal Grasp Planning by Using a 3-D Computer Vision Technique," J. of the KSPE, Vol. 19, No. 11, pp. 54-64, 2002.
11. Chang, S. H., Cheng, F. H., Hsu, W. H. and Wu, G. Z., "Fast Algorithm for Point Pattern Matching: Invariant to Translations, Rotations and Scale Changes," Pattern Recognition, Vol. 30, No. 2, pp. 311-320, 1997.