

3차원 데이터상에 영상등록을 위한 카메라 외부표정 계산 Camera Exterior Orientation for Image Registration onto 3D Data

전재춘¹⁾ · Min Ding²⁾ · Sastry Shankar³⁾
Chon, Jaechoon · Min Ding · Sastry Shankar

Abstract

A novel method to register images onto 3D data, such as 3D point cloud, 3D vectors, and 3D surfaces, is proposed. The proposed method estimates the exterior orientation of a camera with respect to the 3D data through fitting pairs of the normal vectors of two planes passing a focal point and 2D and 3D lines extracted from an image and the 3D data, respectively. The fitting condition is that the angle between each pair of the normal vectors has to be zero. This condition can be represented as a numerical formula using the inner product of the normal vectors. This paper demonstrates the proposed method can estimate the exterior orientation for the image registration as simulation tests.

Keywords : Image mapping, 3D point cloud, Camera exterior orientation, Normal vector, 2D/3D line

초 록

본 논문에서는 3차원 점군, 3차원 벡터 또는 3차원 곡면에 영상등록하는 새로운 방법을 제안 하였다. 제안한 방법은 카메라 위치와 3차원 직선, 2차원 영상 직선을 각각 지나는 평면의 법선벡터의 일치화를 통하여 카메라 외부표정을 추정하는 것이다. 법선벡터 일치화의 조건은 각 법선벡터 쌍의 사잇각이 제로가 되는 것이다. 이 조건은 벡터내적인 수학적식으로 표현 된다. 시뮬레이션을 통하여 제안한 방법이 영상등록을 위한 외부표정 추정을 강인하게 하는 것을 증명 하였다.

핵심어 : 영상매핑, 3차원 점군, 카메라 외부표정, 법선벡터, 에지라인, 3차원 라인

1. 서 론

3차원 GIS, 가상현실에서 사용되는 DEM(Digital Elevation Model) 또는 도시 건물등의 3차원 데이터에 동일지역이 촬영된 영상을 매핑하게 된다면, 현실감을 주게된다. 매핑과정에서 중요한 요소는 카메라 외부표정이다. 일반적으로 카메라 외부표정은 지상기준점을 이용하여 계산된다. 위성 정사영상 또는 항공 정사영상 제작시, DLT(Direct Linear Transform)의 상수 값을 계산하는 것 또한 지상기준점을 이용하고 있다(Ahn, et al., 2001). 지상 기준점을 측량하기 위한 시간과 경비를 최소화하기 3차원 데이터에 카메라 영상을 직접 매핑하는 연구가 진행

되고 있다.

3차원 데이터에 카메라 영상을 직접 매핑하는 연구는 1) 카메라와 라이다/레이저 스캐너가 일체형인 시스템을 이용(Sequiera et al., 1999), 2) 영상과 3차원 점군 데이터 상의 여러 점의 쌍을 이용한 카메라 외부표정 추정(Oberkampf et al., 1996; Quan와 Lan, 1999), 3) 스테레오 카메라 시스템의 추출된 3차원 데이터와 3차원 점군 데이터 상의 여러 점의 쌍을 이용한 카메라 외부표정 추정(Zhao et al., 2005)과 4) 카메라와 3차원 점군 데이터 상에서 추출된 2차원 직선과 3차원 직선간 일치화를 통한 카메라 외부표정 추정으로 나누어진다. 1의 경우, 라이다/레이저 스캐너로부터 취득된 데이터에 다른 카메라

1) 연결저자 · Electrical Engineering and Computer Sciences, University of California at Berkeley(E-mail:jcchon@eecs.berkeley.edu)

2) Electrical Engineering and Computer Sciences, University of California at Berkeley(E-mail:dingm@eecs.berkeley.edu)

3) Electrical Engineering and Computer Sciences, University of California at Berkeley(E-mail:sastry@eecs.berkeley.edu)

로부터 촬영된 영상을 매핑 할 수 없기에 2, 3, 4의 기술이 필요로 한다. 2, 3, 4의 경우는 다른 카메라로부터 촬영된 영상을 매핑 할 수 있는 기술이다. 2와 3의 경우는 3차원 점군 데이터에서 영상의 한 점과 일치하는 점 추출이 매우 어렵기 때문에, 일치하는 점을 찾기 위한 어려움이 없는 4번의 방법이 많이 연구 되고 있다. 4번의 방법은 3차원 점군 데이터에서 자동 3차원 직선 추출(Fleishman et al., 2005; Frueh와 Zakhor, 2003), 추출된 3차원 직선과 영상에서 추출된 2차원 직선 정합(Neumann et al., 2003; Frueh와 Zakhor, 2003)과 정합된 직선을 이용한 카메라 외부표정 추정(Lowe, 1991; Christy와 Horaud, 1999; Stamos와 Allen, 2002)으로 나누어진다. 3차원 점군 데이터를 그레이 영상화 하면, 수동으로 3차원 직선 추출과 영상에서 추출된 직선과 쌍을 만들 수 있다. 이런 이유로 본 논문에서는 3차원 직선 추출과 정합이 되어 있다고 가정한다.

Christy와 Horaud(1999)의 알고리즘은 추출된 3차원 라인 간의 관계를 인식해야 하는 조건이 있으며, Stamos와 Allen(2002)의 알고리즘은 추출된 3차원 라인을 이용하여 여러 개의 소멸점(vanishing point)을 찾아야 하는 단점이 있다. Lowe(1991)은 추출된 3차원 직선을 영상에 투영한 후, 투영된 라인과 영상에서 추출된 라인의 일치화에 대한 연구를 하였다. 이 논문은 최소 3개의 쌍으로 카메라 외부표정을 추정할 수 있는 장점이 있는 반면에, 추정을 위해 주어지는 초기 카메라 외부표정이 실제 외부표정요소와 근접해야 하는 단점이 있다.

본 논문은 추출된 3차원 직선들간의 아무런 제약 조건

이 없이 카메라 외부표정을 추정할 수 있고, 주어진 초기 카메라 외부표정에 상관없이 강인하게 카메라 외부표정을 찾는 새로운 방법을 제안 하였다. 본 논문에서 제안한 방법은 공간상에 위치하는 카메라 초점과 추출된 3차원 라인을 지나는 평면의 법선벡터와 카메라 초점과 추출된 영상 에지직선을 지나는 평면의 법선벡터를 일치화 하는 것이다. 일치화를 위한 수식은 두 법선벡터 기반한 벡터 내적을 사용했다. 조건은 두 벡터간의 사잇각이 영이 되는 것이다. 주어진 초기 카메라 외부표정과 상관없이 신뢰성 높은 카메라 외부표정을 추정하는 것을 시뮬레이션으로 증명하였다.

2. 법선벡터 조건식

라이다/레이저 스캐너로 취득된 3차원 점군 데이터를 기준좌표계로 설정한 경우, 카메라 초점 f 의 위치 $P_C = [X_C \ Y_C \ Z_C]^T$ 와 카메라의 자세 $\Omega = [\omega \ \phi \ \kappa]^T$ 는 그림 1과 같이 표현 된다.

그림 1에서 P_{O_1} 과 P_{O_2} 는 3차원 점군 데이터에서 추출된 3차원 직선상의 점이며, P_{E_1} 과 P_{E_2} 는 영상에서 추출한 에지 직선상의 점들이다. P_C 와 3차원 직선상의 P_{O_1} 과 P_{O_2} 을 지나는 평면과 P_C 와 영상 에지 직선상의 P_{E_1} 과 P_{E_2} 을 지나는 평면은 그림 2와 같이 표현된다. 두 평면의 법선벡터는 각각 N_O 와 N_E 이며 다음 식으로 계산된다.

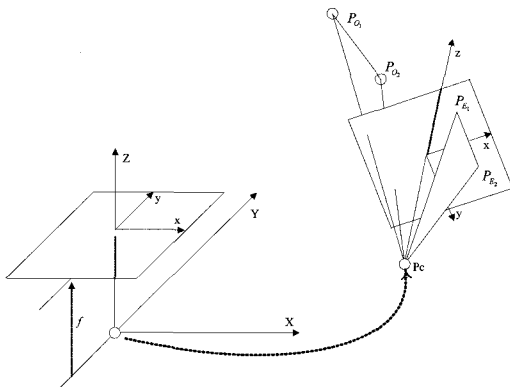


그림 1. Camera exterior orientation referenced in the coordinate system of 3D point cloud data.

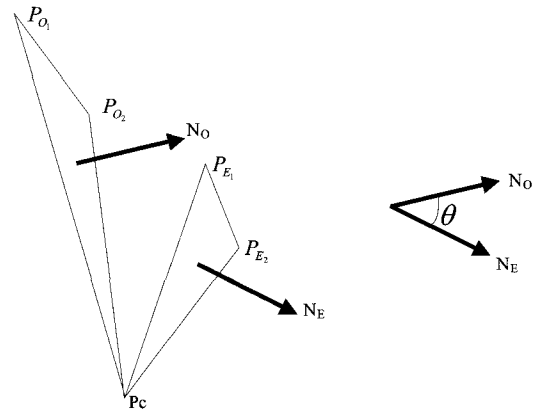


그림 2. Two planes passing a focal point and 2D and 3D lines.

3. 카메라 외부표정 추정

$$N_O = \begin{bmatrix} X_{N_O} \\ Y_{N_O} \\ Z_{N_O} \end{bmatrix} = (P_{O_1} - P_C) \times (P_{O_2} - P_C),$$

$$N_E = \begin{bmatrix} X_{N_E} \\ Y_{N_E} \\ Z_{N_E} \end{bmatrix} = (P_{E_1} - P_C) \times (P_{E_2} - P_C) \quad (1)$$

여기서, N_O 와 N_E 의 전개식은 부록에 기술 하였다. 식 (1)의 P_E 는 영상좌표에서 추출된 에지 직선상의 점(x_e, y_e)을 공간좌표인 일반적인 투영식을 이용하였다.

$$P_E = \lambda R_\kappa R_\phi R_\omega \begin{bmatrix} x_e - x_0 \\ y_e - y_0 \\ f/\lambda \end{bmatrix} + P_C \quad \text{또는}$$

$$\lambda \begin{bmatrix} x_e - x_0 \\ y_e - y_0 \\ f/\lambda \end{bmatrix} = R_{-\omega} R_{-\kappa} R_{-\phi} [P_E - P_C] \quad (2)$$

여기서, λ 는 CCD 한 픽셀의 크기이며, (x_0, y_0)는 영상의 투영중심이며, $R_\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix}$, $R_\phi = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$, $R_\kappa = \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 는 각각 X, Y, Z축 회전행렬이다.

두 법선벡터의 사이각 θ 가 영이 되면 영상데이터와 3차원 점군 데이터를 일치화 할 수 있다. 두 법선벡터와 θ 를 수학적 조건 맺어 줄 수 있는 것은 벡터내적이다.

$$N_O \cdot N_E = L_O L_E \cos(\theta) \quad (3)$$

여기서, $L_{N_O} = |N_O| = \sqrt{X_{N_O}^2 + Y_{N_O}^2 + Z_{N_O}^2}$, $L_{N_E} = |N_E| = \sqrt{X_{N_E}^2 + Y_{N_E}^2 + Z_{N_E}^2}$ 이다. 두 법선벡터가 일치화 된다면 영상의 에지직선과 3차원 점군 데이터 상에서 추출된 직선의 일치화가 된다. 식 (3)에서 θ 를 영으로 설정하면 직선 일치화를 만족 시킬 수 있다. 식 (4)에서 F 값은 법선벡터의 근접도를 의미하는 것이다.

$$F = \frac{N_O \cdot N_E}{L_O L_E} - 1 = 0$$

$$= \frac{(P_{O_1} - P_C) \times (P_{O_2} - P_C) \cdot (P_{E_1} - P_C) \times (P_{E_2} - P_C)}{L_O L_E} - 1 \quad (4)$$

본 논문에서 추정하고자 하는 카메라 외부표정은 6개의 미지수로서 최소 6쌍의 3차원 직선과 영상 에지직선이 필요로 한다. 식 (4)를 n 개의 쌍으로 표현하면 다음과 같다.

$$F_i = \frac{(P_{O_i} - P_C) \times (P_{O_2} - P_C) \cdot (P_{E_i} - P_C) \times (P_{E_2} - P_C)}{L_{O_i} L_{E_i}} - 1$$

$$i = 1, 2, \dots, n \quad (5)$$

식 (5)는 6개의 미지수가 비선형으로 연관되어 있기 때문에 선형화를 통한 계산이 필요하다. 본 논문에서는 선형화를 Taylor series 방법을 사용하였다. 선형과정에서 1차식까지만 이용하였다.

$$F_i = F_i^0 + \left(\frac{\partial F_i^0}{\partial X_C} \right) dX_C + \left(\frac{\partial F_i^0}{\partial Y_C} \right) dY_C + \left(\frac{\partial F_i^0}{\partial Z_C} \right) dZ_C$$

$$+ \left(\frac{\partial F_i^0}{\partial \omega} \right) d\omega + \left(\frac{\partial F_i^0}{\partial \phi} \right) d\phi + \left(\frac{\partial F_i^0}{\partial \kappa} \right) d\kappa \quad (6)$$

여기서, F_i^0 는 미지수의 초기 입력값에 의한 F 값이며, 식 (6) 각 편미분 전개는 부록에 기술 하였다. n 개의 쌍에 대한 식 (6)은 다음과 같이 표현된다.

$$B = J \Delta \quad (7)$$

여기서, $\Delta_{6 \times 1} = [dX_C \ dY_C \ dZ_C \ d\omega \ d\phi \ d\kappa]^T$ 는 관측대상의 미세조정 값이며, $B_{n \times 1} = \begin{bmatrix} \vdots \\ -F^0(t) \\ \vdots \end{bmatrix}$, $J_{n \times 6} =$

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_i^0}{\partial X_C} & \frac{\partial F_i^0}{\partial Y_C} & \frac{\partial F_i^0}{\partial Z_C} & \frac{\partial F_i^0}{\partial \omega} & \frac{\partial F_i^0}{\partial \phi} & \frac{\partial F_i^0}{\partial \kappa} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

이다. Δ 는 최소자승

법으로 계산 가능하며, 15개의 미지수는 다음 식을 반복하여 계측된다.

$$\Delta = (J^T J)^{-1} J^T B$$

$$[X_C \ Y_C \ Z_C \ \omega \ \phi \ \kappa]_j^T = [X_C \ Y_C \ Z_C \ \omega \ \phi \ \kappa]_{j-1}^T + \Delta \quad (8)$$

여기서, j 는 반복회수이다.

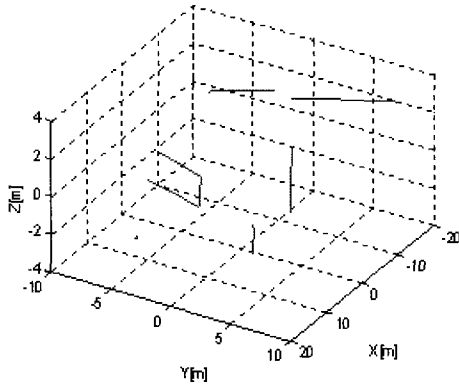
4. 시뮬레이션

본 논문에서 제안한 방법을 검증하기 위해, MATLAB으로 카메라 외부표정을 추정 하였다. 시뮬레이션을 위한 카메라 제원은 표 1과 같다. 카메라 내부표정과 렌즈왜곡은 처리되어 있는 것으로 가정한다.

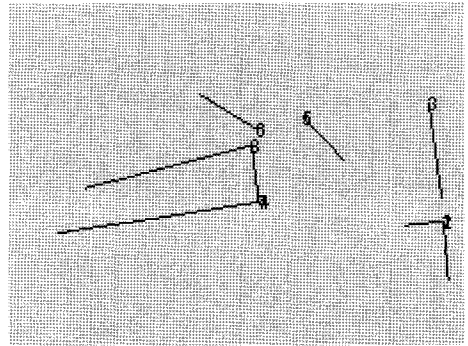
그림 3(a)에서 직선들은 3차원 점군 데이터로부터 3차원 직선을 추출 했다고 가정했을 때, 주어진 직선들이며, 점은 카메라 초점의 위치이다. 그림 3(b)에서 직선은 주어진 3차원 직선 데이터를 (2)식을 통하여 영상에 투영했을 때 결과이다.

표 1. Camera reference.

Image Resolution: 1280 pixel × 960 pixel
CCD Size: 5.953 mm × 4.464 mm
Each Cell Size (λ): 4.65 μm × 4.65 μm
Focal length: 4.5 mm (= 967.74 pixel)

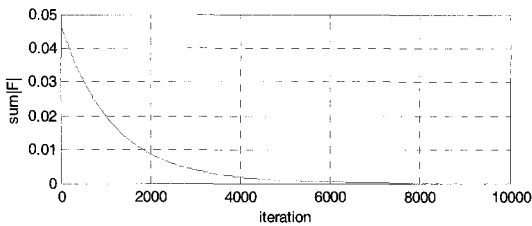


(a) 3D space

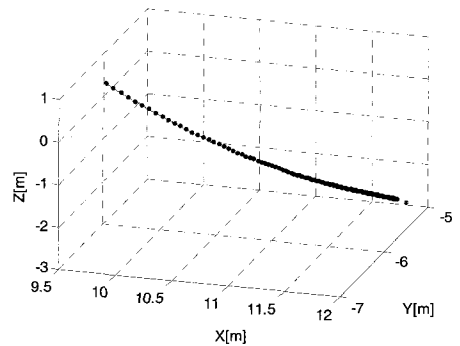


(b) Image space

그림 3. 3D lines projection on an image plane.



(a) F variation



(b) Camera position variation

그림 4. F variation when fixing the camera pose and estimating camera position.

을 때 결과이다.

그림 4는 카메라의 자세는 주어진 경우에 카메라의 위치를 추정한 결과이며, 그림 5는 카메라 위치가 주어진 경우에 카메라의 자세를 추정한 결과이다. 식 (5)에 자세와 위치를 분류하여 추정 한 이유는 각 항목에 식 (5)의 민감도를 측정하기 위한 것이다. 시뮬레이션 조건은 표 2의 Test 1을 사용한 것이다. 그림 4(a)는 F 변화값을 나타낸 것이며, 그림 4(b)는 카메라 위치변화를 나타낸 것이다. 그림 4(b)에서 파란점은 식 (8)을 100번 반복 단위로 나타낸 것이다. 빨간점은 실제 카메라의 위치이다. 그림에서 보듯이 실제위치에 근접할수록 매우 느리게 수렴하는 것을 알 수 있다. 이와 반대로 그림 5에서의 F 의 변화값은 매우 빠르게 수렴하는 것을 알 수 있다. 이 시뮬레이션 결과로서 식 (5)는 카메라의 자세보다 위치에 민감하다는 것을 알 수가 있다.

그림 6(a), 6(b), 6(c)는 각각 표 2에서 Test 1, 2, 3에서

표 2. Errors of the initial given camera exterior orientation and estimation results

	Given Error	Estimation Result
Test 1	Position: [-11.9117 5.2837 2.6052] [m] Pose: [-100.48 -182.46 93.94] [degree]	Position: [0.0995 0.0434 0.1048] [m] Angle: [0.2449 0.1015 0.1426] [degree]
Test 2	Position: [-2.2 -1.2 3.5] [m] Pose: [60.5 -10.9 -50.2] [degree]	Position: [0.0895 0.0419 0.0291] [m] Angle: [0.493 0.613 0.837] [degree]
Test 3	Position: [2.2 1.2 -3.5] [m] Pose: [-60.5 10.9 50.2] [degree]	Position: [0.2306 0.1682 0.1307] [m] Angle: [0.5904 0.3315 0.1582] [degree]

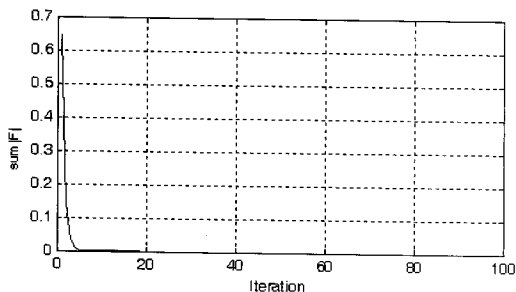


그림 5. F variation when fixing the camera position and estimating camera pose.

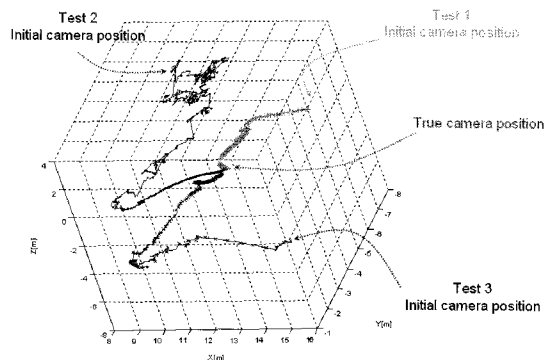
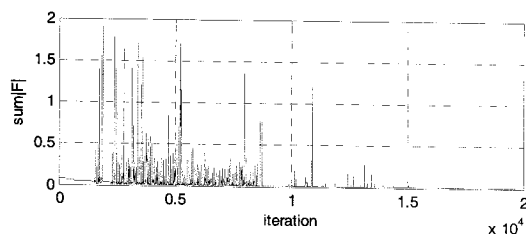
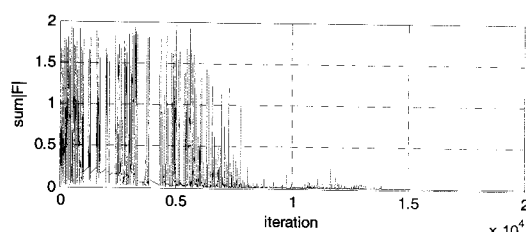


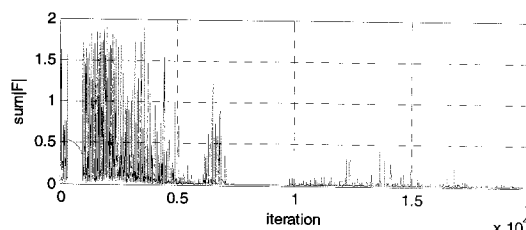
그림 7. Camera position variation under Tests 1, 2, and 3 in Table 2.



(a) Test 1



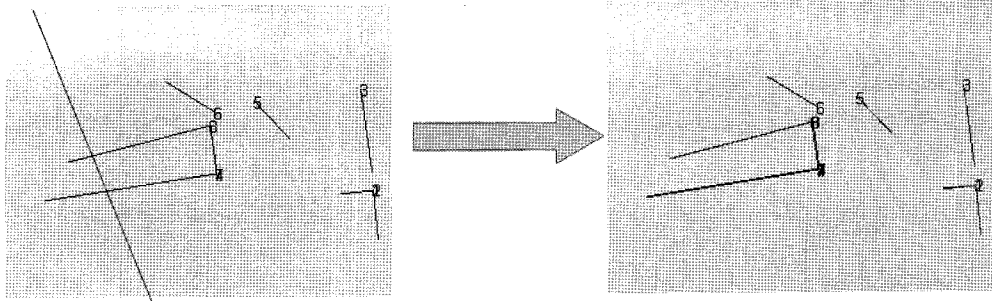
(b) Test 2



(c) Test 3

그림 6. F variation under Tests 1, 2, and 3 in Table 2.

주어진 초기조건을 이용하여 카메라 외부표정을 추정할 때의 F 의 변화 값이다. 여기서 표 2의 값들은 실제 카메라 외부표정으로부터 거리/위치차를 의미한다. 그림 7은 추정과정에서 카메라 위치변화를 표현한 것이다. 그림 8(a)는 주어진 초기조건을 이용하여 3차원 직선을 영상프레임에 투영한 결과이며, 그림 8(b)는 추정결과를 이용하여 투영한 결과이다. 여기서, 빨간 직선은 투영된 3차원 직선, 파란 직선은 영상 에지직선이다. 표 2에서 주어진 초기조건을 이용하여 3차원직선을 영상프레임에 투영했을 때, Test 2의 경우에만 한 개의 투영된 3차원 직선만 가시화 될 정도로 매우 좋지 않은 초기조건을 이용한 것이다. 표 2의 추정결과와 그림 8(b)에서 보듯이, 매우 좋지 않은 초기조건에도 제안한 방법은 모두 실제 카메라 외부표정에 근접 했다는 것을 알 수 있다. 표 2의 초기조건을 Lowe(1991)의 알고리즘에 적용 했을 때, 모두 발산하여 결과 값을 얻지 못하였다. 이 알고리즘의 경우는 초기조건은 투영된 3차원 직선과 영상 에지직선의 매우 근접할 때 추정이 가능한 것으로 알려져 있다. 제안한 알고리즘은 추정반복이 2만 번 이상해야 한다는 단점이 있지만 초기조건과 상관없이 수렴하여 실근을 계산하는 장점을 가지고 있다. 2만 번 추정반복은 MATLAB환경에선 5



Initial (Test 2)

Result (Test 1, 2, and 3)

(a) Using initial orientation

(b) Using estimated orientation

그림 8. Simulation results of 3D lines projected onto an image frame using estimated camera exterior orientations.

분정도 소요 되지만, C언어에서는 1초 이내에 계산이 가능하다.

5. 결 론

본 논문은 3차원 점군 데이터에서 추출된 3차원 직선들간의 아무런 제약 조건이 없이 카메라 외부표정을 추정할 수 있고, 주어진 초기 카메라 외부표정에 상관없이 강인하게 카메라 외부표정을 찾는 새로운 방법을 제안 하였다. 본 논문에서 제안한 방법은 공간상에 위치하는 카메라 초점과 추출된 3차원 라인을 지나는 평면의 법선벡터와 카메라 초점과 추출된 영상 에지직선을 지나는 평면의 법선벡터를 일치화 하는 것이다.

제안한 방법은 DEM 또는 3차원 점군 데이터 등에서 지역적으로 3차원 라인이 취득이 가능하다면 지상기준점을 이용하지 않고 카메라 외부표정을 계산 할 수 있기에, 각종 정사영상 제작과 신속한 데이터 업데이트가 필요로 하는 3차원 GIS분야와 가상현실 분야에 기여 할 것으로 예상된다.

참고문헌

- Ahn, Chung-Hyun, Seong-Ik Cho, and Jaechoon Chon (2001), Ortho-rectification software applicable for IKONOS high resolution images: GeoPixel-Ortho, IGARSS 2001, pp. 555-557, Sydney 2001.
- Christy, S. and R. Horaud (1999), Iterative pose computation from line correspondences. *CVIU*, Vol. 73, No. 1, pp. 137-144.
- Fleishman S., D. C. Tel-Aviv, and C. T. Silva (2005), Robust

- moving least-squares fitting with sharp features, ACM SIGGRAPH 2005, pp. 544-552, Los Angeles, California.
- Frueh, Christian and Avidesh Zakhor (2003), Constructing 3D City Models by Merging Ground-Based and Airborne Views, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03), Vol. 2, pp. 562-569.
- Lowe, David G. (1991), Fitting parameterized three-dimensional models to images, *IEEE Tr. on PAMI*, Vol. 13, No. 5, pp. 441-450.
- Neumann, Ulrich, Suya You, Jinhui Hu, Bolan Jiang, and JongWeon LeeU (2003), Augmented Virtual Environments (AVE): Dynamic Fusion of Imagery and 3D Models, *Virtual Reality 2003*, pp. 61-67.
- Oberkampff D., D. DeMenthon, and L. Davis (1999), Iterative pose estimation using coplanar feature points. *CVGIP*, 63(3).
- Sequiera, V., K. Ng, E. Wolfart, J. Concalves, and D. Hogg (1999), "Automated reconstruction of 3D models from real environments," *ISPRSJ. Photogrammetry Remote Sensing*, Vol. 54, pp. 1-22.
- Stamos, I. and P. Allen (2002), Geometry and Texture Recovery of Scenes of Large Scale, *CVIU*, Vol. 88, pp. 94-118.
- Quan L. and Z. Lan (1999), Linear N. point camera pose determination. *IEEE Tr. on PAMI*, Vol. 21, No. 8, pp. 774-780.
- Zhao. W., D. Nister, and S. Hsu (2005), Alignment of Continuous Video onto 3D Point Clouds, *IEEE Tr. on PAMI* Vol. 27, No. 8, pp. 1305-18.

부 록

$$N_o = \begin{bmatrix} X_{N_o} \\ Y_{N_o} \\ Z_{N_o} \end{bmatrix} = (P_o - P_c) \times (P_{o_2} - P_c) = \begin{bmatrix} \bar{X} & \bar{Y} & \bar{Z} \\ X_{o_1} - X_c & Y_{o_1} - Y_c & Z_{o_1} - Z_c \\ X_{o_2} - X_c & Y_{o_2} - Y_c & Z_{o_2} - Z_c \end{bmatrix}$$

$$= \begin{bmatrix} (Y_{o_1} - Y_c)(Z_{o_2} - Z_c) - (Z_{o_1} - Z_c)(Y_{o_2} - Y_c) \\ (Z_{o_1} - Z_c)(X_{o_2} - X_c) - (X_{o_1} - X_c)(Z_{o_2} - Z_c) \\ (X_{o_1} - X_c)(Y_{o_2} - Y_c) - (Y_{o_1} - Y_c)(X_{o_2} - X_c) \end{bmatrix}$$

$$N_E = \begin{bmatrix} X_{N_E} \\ Y_{N_E} \\ Z_{N_E} \end{bmatrix} = (P_{E_1} - P_C) \times (P_{E_2} - P_C) = \begin{bmatrix} \bar{X} & \bar{Y} & \bar{Z} \\ X_{E_1} - X_C & Y_{E_1} - Y_C & Z_{E_1} - Z_C \\ X_{E_2} - X_C & Y_{E_2} - Y_C & Z_{E_2} - Z_C \end{bmatrix}$$

$$= \begin{bmatrix} (Y_{E_1} - Y_C)(Z_{E_2} - Z_C) - (Z_{E_1} - Z_C)(Y_{E_2} - Y_C) \\ (Z_{E_1} - Z_C)(X_{E_2} - X_C) - (X_{E_1} - X_C)(Z_{E_2} - Z_C) \\ (X_{E_1} - X_C)(Y_{E_2} - Y_C) - (Y_{E_1} - Y_C)(X_{E_2} - X_C) \end{bmatrix},$$

$$\frac{\partial F}{\partial P_C} = \left(\frac{\partial X_{N_0}}{\partial P_C} X_{N_1} + X_{N_0} \frac{\partial X_{N_1}}{\partial P_C} + \frac{\partial Y_{N_0}}{\partial P_C} Y_{N_1} + Y_{N_0} \frac{\partial Y_{N_1}}{\partial P_C} + \frac{\partial Z_{N_0}}{\partial P_C} Z_{N_1} + Z_{N_0} \frac{\partial Z_{N_1}}{\partial P_C} \right) / (L_{N_0} L_{N_1})$$

$$-(F+1) \left(\frac{\partial L_{N_0}}{\partial P_C} / L_{N_0} + \frac{\partial L_{N_1}}{\partial P_C} / L_{N_1} \right),$$

$$\frac{\partial F}{\partial \Omega} = \left(\frac{\partial X_{N_0}}{\partial \Omega} X_{N_1} + X_{N_0} \frac{\partial X_{N_1}}{\partial \Omega} + \frac{\partial Y_{N_0}}{\partial \Omega} Y_{N_1} + Y_{N_0} \frac{\partial Y_{N_1}}{\partial \Omega} + \frac{\partial Z_{N_0}}{\partial \Omega} Z_{N_1} + Z_{N_0} \frac{\partial Z_{N_1}}{\partial \Omega} \right) / (L_{N_0} L_{N_1})$$

$$-(F+1) \left(\frac{\partial L_{N_0}}{\partial \Omega} / L_{N_0} + \frac{\partial L_{N_1}}{\partial \Omega} / L_{N_1} \right),$$

$$\frac{\partial}{\partial P_C} \begin{bmatrix} X_{N_0} \\ Y_{N_0} \\ Z_{N_0} \end{bmatrix} = \begin{bmatrix} 0 & -Z_{O2} + Z_{O1} & -Y_{O1} + Y_{O2} \\ -Z_{O1} + Z_{O2} & 0 & -X_{O2} + X_{O1} \\ -Y_{O2} + Y_{O1} & -X_{O1} + X_{O2} & 0 \end{bmatrix}, \quad \frac{\partial}{\partial \omega \phi \kappa} \begin{bmatrix} X_{N_0} \\ Y_{N_0} \\ Z_{N_0} \end{bmatrix} = 0,$$

$$\frac{\partial}{\partial P_C} \begin{bmatrix} X_{N_E} \\ Y_{N_E} \\ Z_{N_E} \end{bmatrix} = \begin{bmatrix} 0 & -Z_{E_2} + Z_{E_1} & -Y_{E_1} + Y_{E_2} \\ -Z_{E_1} + Z_{E_2} & 0 & -X_{E_2} + X_{E_1} \\ -Y_{E_2} + Y_{E_1} & -X_{E_1} + X_{E_2} & 0 \end{bmatrix},$$

$$\frac{\partial}{\partial \Omega} \begin{bmatrix} X_{N_E} \\ Y_{N_E} \\ Z_{N_E} \end{bmatrix} = \begin{bmatrix} \frac{\partial Y_{E_1}}{\partial \omega \phi \kappa} (Z_{E_2} - Z_C) + (Y_{E_1} - Y_C) \frac{\partial Z_{E_2}}{\partial \omega \phi \kappa} - \frac{\partial Z_{E_1}}{\partial \omega \phi \kappa} (Y_{E_2} - Y_C) - (Z_{E_1} - Z_C) \frac{\partial Y_{E_2}}{\partial \omega \phi \kappa} \\ \frac{\partial Z_{E_1}}{\partial \omega \phi \kappa} (X_{E_2} - X_C) + (Z_{E_1} - Z_C) \frac{\partial X_{E_2}}{\partial \omega \phi \kappa} - \frac{\partial X_{E_1}}{\partial \omega \phi \kappa} (Z_{E_2} - Z_C) - (X_{E_1} - X_C) \frac{\partial Z_{E_2}}{\partial \omega \phi \kappa} \\ \frac{\partial X_{E_1}}{\partial \omega \phi \kappa} (Y_{E_2} - Y_C) + (X_{E_1} - X_C) \frac{\partial Y_{E_2}}{\partial \omega \phi \kappa} - \frac{\partial Y_{E_1}}{\partial \omega \phi \kappa} (X_{E_2} - X_C) - (Y_{E_1} - Y_C) \frac{\partial X_{E_2}}{\partial \omega \phi \kappa} \end{bmatrix},$$

$$\frac{\partial P_E}{\partial \Omega} = \lambda \left[R_x R_\phi \frac{\partial R_\omega}{\partial \omega} \quad R_x \frac{\partial R_\phi}{\partial \phi} R_\omega \quad \frac{\partial R_c}{\partial \kappa} R_\phi R_\omega \right] \begin{bmatrix} x_c - x_0 \\ y_c - y_0 \\ f / \lambda \end{bmatrix},$$

$$\frac{\partial R_\omega}{\partial \omega} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin \omega & -\cos \omega \\ 0 & \cos \omega & -\sin \omega \end{bmatrix},$$

$$\frac{\partial R_\phi}{\partial \phi} = \begin{bmatrix} -\sin \phi & 0 & \cos \phi \\ 0 & 0 & 0 \\ -\cos \phi & 0 & -\sin \phi \end{bmatrix},$$

$$\frac{\partial R_\kappa}{\partial \kappa} = \begin{bmatrix} -\sin \kappa & -\cos \kappa & 0 \\ \cos \kappa & -\sin \kappa & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

(접수일 2007. 4. 10, 심사일 2007. 10. 10, 심사완료일 2007. 10. 13)