

연속적인 이미지를 이용한 3차원 물체의 복원

김성진*, 김강현**, 전희성*

*울산대학교 대학원 컴퓨터정보통신공학과

**부산디지털정보기능대학 정보통신시스템과

3D Reconstruction from an Image Sequence

Seong-jin Kim*, Kang-hyun Kim**, Heesung Jun*

Graduate School of Computer Engineering & Information Technology, University of Ulsan
Dept. of Information & Communication System, Busan Information Polytechnic College

요약

이미지를 기반으로 3차원 모델을 생성하기 위한 방법은 많은 사람들의 연구의 대상이 되어 왔다. 본 연구에서는 연속적으로 획득된 여러 장의 이미지로부터 특징점을 추출한 후, 사영복원과 유클리디언 복원을 이용하여 특징점에 대응되는 3차원 데이터를 계산하는 방법을 구현하였고, 이렇게 얻은 3차원 데이터에 텍스쳐 매핑을 결합하여, 보다 사실적인 3차원 모델을 생성할 수 있는 시스템을 구현하였다.

1. 서론

순차적인 이미지로부터 물체의 형상과 운동 파라미터를 복원하는 연구는 많은 관심의 대상이 되어 왔다. Tomasi와 Kanade에 의해 처음으로 factorization 방법이 개발된 후 많은 연구가 진행되어 왔다[1,2]. 그러나 대부분의 factorization 방법이 카메라의 보정이 필요하다는 단점이 있었고, 최근 연구의 방향은 제약이 적고 일반적인 상황에 적용될 수 있는 사전에 보정되지 아니한 카메라를 사용하는 방법이다. 이 방법은 실험실에서 카메라에 대한 특성을 미리 파악하지 않고도 찍힌 사진 또는 연속영상을 이용하여 3차원 데이터를 복원할 수 있다.

보정되지 않은 카메라로부터 획득된 연속적인 이미지에 대하여 사영복원과 유클리디언 복원을 수행하여 3차원 모델 데이터를 구성하고, 여기서 얻어진 데이터를 보다 사실적으로 표현하기 위해 텍스쳐 매핑을 사용하여 다른 시점의 이미지를 재구성한다.

본 논문의 구성은 다음과 같다. 2장에는 3차원 데이터의 복원 방법에 대해 기술하고, 3장에는 시스템의 구현 및 구현된 시스템을 이용해 얻은 실험결과에 대해 상술하였다. 마지막으로 4장의 결론 부분에는 구현 성과 및 앞으로 해결해야 할 과제에 대해 언급하였다.

2. 3차원 데이터의 복원

2.1 사영복원(Projective Reconstruction)

m 개의 카메라(또는 관측점)에서 모두 관측 가능한 n 개의 3차원 점이 있을 때, 이러한 영상좌표의 측정값 (u_{ij}, v_{ij}) 만을 이용하여, 각 카메라의 원근투영행렬 P_i 와 각 영상점의 3차원 위치 X_j 를 복원하려는 것이 사영복원 문제이다. 이상의 내용을 동차좌표를 사용하여 식으로 표현하면 다음과 같다.

$$\lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = P_i X_j \quad (1)$$

λ_{ij} 는 사영깊이(projective depth) 또는 배율(scale factor)이라 부른다. 위의 식을 모든 카메라와 모든 점에 대해 종합하면 다음과 같다.

$$W_s = \begin{bmatrix} \lambda_{11} \begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \cdots & \lambda_{1n} \begin{bmatrix} u_{1n} \\ v_{1n} \\ 1 \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \lambda_{m1} \begin{bmatrix} u_{m1} \\ v_{m1} \\ 1 \end{bmatrix} & \cdots & \lambda_{mn} \begin{bmatrix} u_{mn} \\ v_{mn} \\ 1 \end{bmatrix} \end{bmatrix} \quad (2)$$

$$= \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} [X_1 \ X_2 \ \cdots \ X_n]$$

2.2 반복 사영복원

앞에서 언급한 사영복원 방법은 여러 논문에서 언급되었고 잘 알려져 있는 방법이다. 본 연구에서는 Chen과 Medioni의 사영복원 알고리듬[3,4]을 바탕으로 다음과 같이 일부 수정한 방법을 사용하였다.

1. 영상좌표의 정규화(좌표의 행, 열의 값이 -0.5 ~ 0.5 범위에 오도록 계산, 안정적인 계산에 필수적임[1])
2. scale factor의 계산(맨 처음에는 $\lambda_{ij} = 1$ 로 계산함)
3. 식 (2)을 이용하여 측정 행렬 W_s 를 계산
4. 특이값 분해(SVD - singular value decomposition) 방법을 이용하여 W_s 의 rank4 factorization을 행함.
5. 사영복원을 행함.
6. W_s 를 특이값 분해했을 때 4번째 특이값과 5번째 특이값의 차이가 미리 정한 범위에 들어올 때까지 단계 2로 돌아가 계산을 반복함.

사영복원의 주 목적은 식 (2)을 일관되게 만드는 λ_{ij} 를 구하는 것이라 할 수 있다.

$$W_s = UDV^T \quad (3)$$

위와 같이 분리한 행렬로부터 사영 운동행렬과 형상벡터를 구하기 위해 다음 세 가지 방법 중 하나를 사용한다.

$$P = UD, \quad X = V^T \quad (4)$$

$$P = U\sqrt{D}, \quad X = \sqrt{D}V^T \quad (5)$$

$$P = U, \quad X = DV^T \quad (6)$$

2.2 유클리디언 복원(Euclidean Reconstruction)

사영복원(Projective Reconstruction)을 통해 운동(motion)과 형상(shape)에 대한 \hat{P} 와 \hat{X} 를 구하였다. 여기서 우리가 원하는 실질적인 유클리디언 형태의 P와 X를 구하기 위해 $P = \hat{P}H$, $X = H^{-1}\hat{X}$ 형식으로 식을 놓고 특정 제약조건에 의해 H를 구함으로써 유클리디언 복원을 수행한다[2].

$$W_s = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = PX \quad (7)$$

사영 운동 행렬(Projective motion matrix) P 는

$$P = [M \mid T] \quad (8)$$

이고, 여기서

$$M = [\mathbf{m}_{x1} \ \mathbf{m}_{y1} \ \mathbf{m}_{z1} \ \dots \ \mathbf{m}_{xn} \ \mathbf{m}_{yn} \ \mathbf{m}_{zn}]^T$$

$$T = [T_{x1} \ T_{y1} \ T_{z1} \ \dots \ T_{xn} \ T_{yn} \ T_{zn}]^T$$

그리고,

$$\mathbf{m}_{xi} = \mu_i f_i \ \mathbf{i}_i + \mu_i u_{0i} \ \mathbf{k}_i$$

$$\mathbf{m}_{yi} = \mu_i a f_i \ \mathbf{j}_i + \mu_i v_{0i} \ \mathbf{k}_i$$

$$\mathbf{m}_{zi} = \mu_i \ \mathbf{k}_i$$

$$\begin{aligned} T_{xi} &= \mu_i f_i t_{xi} + \mu_i u_{0i} t_{zi} \\ T_{yi} &= \mu_i a f_i t_{yi} + \mu_i v_{0i} t_{zi} \\ T_{zi} &= \mu_i t_{zi} \end{aligned} \quad (9)$$

이다. 다시 정리하면

$$P = \hat{P}H = K[R]t_i = \hat{P}[A|B] = [M|T] \quad (10)$$

$$\hat{P}A = M, \quad \hat{P}B = T \quad (10)$$

$$T_{xi} = \hat{p}_{xi} B, \quad T_{yi} = \hat{p}_{yi} B, \quad T_{zi} = \hat{p}_{zi} B \quad (11)$$

가 된다. 식 (10)에서 B를 구하기 위해서는 먼저 복원될 3차원 좌표점의 기준점을 정해야 한다.

복원될 형상행렬(Shape Matrix)은

$$X \sim \begin{bmatrix} S \\ 1 \end{bmatrix} \quad (12)$$

$$S = [s_1 \ s_2 \ \dots \ s_m], \quad s_j = [x_j \ y_j \ z_j]^T$$

$$\mathbf{x}_j = [v_j \ s_j^T \ v_j]^T$$

이다. 여기서 월드좌표계(world coordinate system)의 원점이 특정 점들의 무게중심으로 되도록 기준점을 정하면,

$$\sum_{j=1}^m v_j \cdot s_j = 0 \quad (13)$$

i) 되고, 이것으로 다음과 같은 결과를 얻을 수 있다.

$$\begin{aligned} \sum_{j=1}^m \lambda_{ij} u_{ij} &= \sum_{j=1}^m (\mathbf{m}_{xi} \cdot v_j \cdot s_j + v_j \cdot T_{xi}) \\ &= \mathbf{m}_{xi} \cdot \sum_{j=1}^m v_j \cdot s_j + T_{xi} \sum_{j=1}^m v_j = T_{xi} \sum_{j=1}^m v_j \end{aligned}$$

마찬가지로

$$\sum_{j=1}^m \lambda_{ij} v_{ij} = T_{yi} \sum_{j=1}^m v_j, \quad \sum_{j=1}^m \lambda_{ij} = T_{zi} \sum_{j=1}^m v_j \quad (14)$$

가 된다. 식(14)에서

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^m \lambda_{ij} u_{ij}}{\sum_{j=1}^m \lambda_{ij}}, \quad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^m \lambda_{ij} v_{ij}}{\sum_{j=1}^m \lambda_{ij}} \quad (15)$$

이고, 식 (11)과 (15)에서 미지수가 4개인 2n개의 선형방정식을 만들 수 있으며, 이것은 선형 최소자승법[1]을 통해 미지수인 B값을 비교적 쉽게 찾을 수 있다.

이제 식 (10)의 $\hat{P}A = M$ 에서 A를 구하기 위해 다음과 같이 놓는다.

$$MM^T = \hat{P}AA^T \hat{P}^T \quad MM^T = \hat{P}Q \hat{P}^T \quad (16)$$

$$\left[\begin{array}{c} \mathbf{m}_{xi} \\ \mathbf{m}_{yi} \\ \mathbf{m}_{zi} \end{array} \right] \left[\begin{array}{ccc} \mathbf{m}_{xi} & \mathbf{m}_{yi} & \mathbf{m}_{zi} \end{array} \right] = \hat{P} \left[\begin{array}{cccc} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{array} \right] \hat{P}^T$$

$$= \begin{bmatrix} |\mathbf{m}_{xi}|^2 & \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} & \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} \\ \mathbf{m}_{yi} \cdot \mathbf{m}_{xi} & |\mathbf{m}_{yi}|^2 & \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} \\ \mathbf{m}_{zi} \cdot \mathbf{m}_{xi} & \mathbf{m}_{zi} \cdot \mathbf{m}_{yi} & |\mathbf{m}_{zi}|^2 \end{bmatrix} \quad (17)$$

식 (17)에서 우리는 각 프레임당 10개의 선형방정식을 만들 수 있다. 여기서 이미지의 중점을 $(0,0)$ 으로, 스케일 펙터 μ_i 를 1이라 가정하면,

$$|\mathbf{m}_{xi}|^2 = |\mathbf{m}_{yi}|^2 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} = 0, \quad \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} = 0, \quad \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} = 0 \quad (18)$$

가 된다. 식 (17)과 식 (18)에서 각 프레임마다 4개의 방정식을 만들 수 있으며, 선형 최소자승법[1]을 적용하면 식(16)의 Q를 구할 수 있다. 그 후 Q를 랭크3 행렬분해(Rank3 Matrix decomposition)를 하여 A를 구할 수 있다. 단 A값이 존재하는 Q값을 구해야 하는데, 이것은 최소자승법으로 구해진 여러 해를 조합하여 조건에 해당되는 답을 선택하여야 한다.

3. 시스템 구현 및 실험

3.1 시스템 구현

본 시스템은 PC에서 C++ 언어로 구현하였다. 많이 사용되는 특이값 분해는 참고문헌[6]을 참조하였으며, 사용자 인터페이스 부분은 Qt 라이브러리[10], 그래픽 부분은 OpenGL[7]을 이용하여 프로그래밍 하였다.

3.2 실험

개발된 시스템의 효용성을 검증하기 위해 모의데이터와 실제 영상데이터를 사용하여 실험하였다.

모의데이터를 이용한 실험

OpenGL을 이용해 카메라의 위치를 변경시키면서 10개의 다른 관측점에서 바라본 정육면체의 8 꼭지점의 화면좌표를 이용하여 각 꼭지점의 3차원 좌표를 계산하였다. 변 길이의 참값은 알 수 없으므로 계산된 각 변의 평균값을 구하여 그 값에 대한 오차를 계산한 결과, 평균오차는 0.278%, 최대오차는 0.555%, 최소오차는 0.004%이었다.

또한 복원된 좌표를 바탕으로 각 정점에서 두 개의 변이 이루는 총 24개의 각도(참값은 모두 90도)를 계산하여 오차를 구한 결과, 평균오차는 0.16도, 최대오차는 0.33도, 최소오차는 0.01도이었다. 계산된 결과가 매우 정확함을 알 수 있다.

실제 영상데이터를 이용한 실험

구현된 방법이 특징점이 매우 많은 영상데이터에 대해서도 잘 적용되는지 확인하기 위해 여러 논문에서 사용된 그림 1과 같은 집 모양의 데이터[5]를 이용하였다.

영상데이터에 수작업으로 많은 특징점을 추적하고 정합시키는 일은 시간이 많이 걸리고 오차가 생길 확률이 크다. 따라서 특징점의 자동 추출 및 추적기가 필요하며 현재 알려진 방법 중 프레임 간 운동이 적을 때 많이 사용되고 안정적인 방법인 KLT 트랙커[8-9]를 사용하였다.

사용된 프레임 총 수는 10개이며 각 프레임에 대해 KLT 트랙커를 이용하여 400개의 특징점을 추출하여 첫 프레임부터 마지막까지 모두 추적된 총 279개의 점에 대해 유클리디언 복원을 시도하였다.

구현된 프로그램을 사용하면 특징점에 해당하는 점들이 각 프레임에 동일하게 표시된다. 이들 중 첫 프레임에 특징점을 표시한 것이 그림1에 나타나 있다.

본 논문에서는 KLT 트랙커의 방법을 변형하여 구현하였다. 우선 프로그램에 의해 자동으로 일치점이 찾아지게 되는 데 그 중 정확히 일치하지 않거나 불필요한 점은 제외시킬 수 있으며, 정확하지 않은 특징점은 사용자가 수동으로 이동시킬 수 있게 하였다.



그림 1. 구현된 프로그램에 의해 표시된 특징점

추적된 모든 점에 대해 3차원 좌표값이 필요하다면 앞에서 설명한 사영복원과 유클리디언 복원 프로그램을 이용해 3차원 좌표값을 계산한다. 만약 매우 사실적인 3차원 모델이 필요하다면 몇 개의 특징점에 의해 결정되는 평면에 텍스쳐 매핑을 하는 것이 더욱 효과적일 수 있다. 그림 2에서 보듯이 화면상에서 적절한 세 개의 점의 집합을 선택하면 그에 따라 평면이 화면에 표시되는 데, 이 평면이 포함하는 이미지 부분이 추후 텍스쳐 매핑에 사용된다.

텍스쳐 매핑을 하기 위해 그림 2에 나타낸 것(집: 14점, 14면, 자동차: 8점, 7면, LiFia판: 5점, 3면)과 같

이 복원하였다.

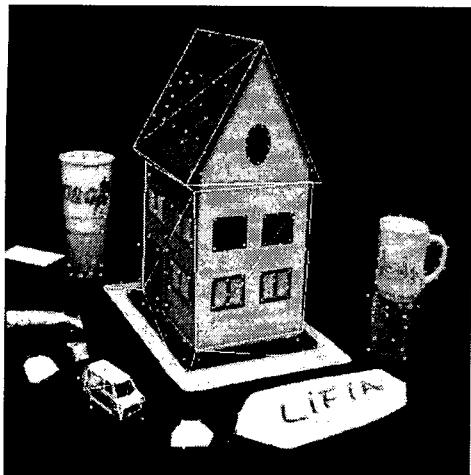


그림 2. 복원을 위한 특징점과 면 구조의 설정

텍스쳐 매핑 결과 매우 사실적인 3차원 모델을 얻을 수 있었다. 그림 3에 서로 다른 두 관측점에서 바라본 복원모델을 나타내었다.

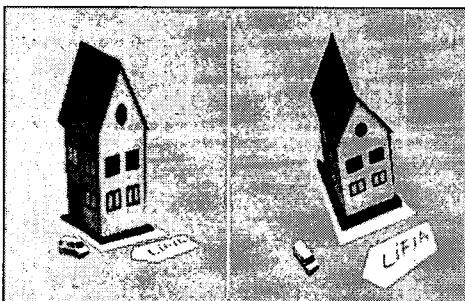


그림 3. 두 다른 각도에서 바라본 복원된 3차원 모델

입력된 데이터에 따라 적절한 해가 존재하지 않아 복원에 실패한 경우도 있으나, 대체로 성공적으로 복원을 수행할 수 있었다.

4. 결론

본 연구에서는 3차원 물체를 촬영한 연속적인 영상 데이터로부터 3차원 물체의 형상을 복원하는 유클리디언 복원방법을 구현하였으며, 텍스쳐 매핑을 통하여 사실적인 3차원 복원을 수행하였다. 구현된 시스템은 다음과 같은 장점을 가지고 있다.

첫째. 특징점을 KLT를 통해 자동으로 추출하고, 사용자가 사실적인 복원에 필요한 몇 개의 특징점을 대화식으로 선택할 수 있게 하였다.

둘째. 구현된 유클리디언 복원시스템은 초기값에 구

애받지 않고 안정된 해를 구할 수 있다.

셋째. 텍스쳐 매핑이 프로그램에서 자동으로 이루어져, 모델링할 때 수작업이 거의 필요하지 않다. 향후의 연구과제는 다음과 같다.

1. 다양한 모의 데이터의 확보 및 실험
2. 참값을 알 수 있는 물체의 이미지를 활용한 유클리디언 복원
3. 보다 간편하고 정확한 특징점의 추출과 정합 및 추적 방법
4. 여러 프레임의 이미지 중 가장 적합한 이미지를 택해 텍스쳐 매핑을 하는 방법

[참고문헌]

- [1] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000
- [2] M. Han and T. Kanade, "Creating 3D Models with Uncalibrated Cameras", Proc. of IEEE Computer Society Workshop on the Application of Computer Vision, Dec. 2000
- [3] Q. Chen, "Multi-view Image-Based Rendering and Modeling", Ph. D. Dissertation, USC, 2000
- [4] Q. Chen and G. Medioni, "Efficient iterative solution to M-view projective reconstruction problem", Proc. IEEE CVPR 1999
- [5] MOVI house data (http://www.inrialpes.fr/movi/pub/Images/sequences/g3_vp_ra_s1/index.html)
- [6] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes in C, Cambridge University Press, 1989
- [7] M. Woo, J. Neider, T. Davis, D. Shreiner, OpenGL Programming Guide, 3rd edition, Addison Wesley, 1999
- [8] Bruce D. Lucas and Takeo Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". International Joint Conference on Artificial Intelligence, pages 674-679, 1981.
- [9] Jianbo Shi and Carlo Tomasi. "Good Features to Track". IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.
- [10] Qt Library (<http://www.trolltech.com/>)