

## 컴퓨터비전 기법을 이용한 물체의 실물 모델링 방법

전희성

컴퓨터·정보통신공학부

### <요 약>

이미지를 이용하여 물체의 형상과 운동 파라미터를 복원하는 문제는 많은 연구자들이 관심을 갖고 있는 매우 중요한 문제이다. 본 연구에서는 3차원 물체를 촬영한 여러 장의 이미지를 이용하여 그 이미지에 포함된 3차원 물체의 형상을 복원하는 유클리디언 복원방법을 구현하였다. 또한 구현된 방법을 이용하여 계산된 물체의 3차원 데이터와 텍스처 맵핑 방법을 결합하여 사실적인 3차원 모델을 만들 수 있음을 보였다.

## Realistic Object Modeling Method by Computer Vision Technique

Heesung Jun

School of Computer Engineering & Information Technology

### <Abstract>

Many researchers have interests in recovering 3D metric structure and motion parameters from 2D images. An Euclidean reconstruction system has been implemented to recover the 3D metric structure and motion parameters from several 2D images. We show that realistic 3D models can be obtained by combining the Euclidean reconstruction system and texture mapping.

## 1. 서론

형상과 운동 파라미터의 복원은 매우 중요한 문제로 많은 연구자들이 관심을 갖고 있다. 최근 가상 환경에 대한 관심이 높아지면서 이러한 가상 세계를 구축하기 위한 기술로써 실제 세계에 존재하는 물체들을 가상 세계에 구축하기 위해 3차원 모델을 만들어내는 작업에 대한 연구가 활발하게 진행되고 있다. 임의의 물체에 대한 3차원 모델을 만들어 내는 방법은 크게 다음과 같은 세 가지로 나눌 수 있다.

- 첫번째, 그래픽 전문가가 컴퓨터를 이용해 그려내는 방법으로 가장 원시적인 방법이고 비용과 노력이 많이 드는 작업이다.
- 두번째, 3차원 디지털라이저 또는 레이저 스캐너를 이용하여 정밀하고 완성도 높은 모델을 만들어 낼 수 있으나 역시 비용이 많이 드는 작업이고 공간적인 제약도 따른다.
- 세번째 방법은 물체를 촬영한 사진들로부터 3차원 모델을 구성해내는 방법이다. 그 대표적인 방법으로 스테레오(stereo), 모션(motion)등이 연구되어 왔다. 스테레오 방법은 대응되는 두 이미지 사이의 차이를 이용하는 방법으로 두 이미지에서 일치점을 찾는 것이 쉽지 않은 문제이다. 모션 방법은 대응되는 이미지들의 기하학적인 특성을 이용하는 방법으로 각 사진에서의 특정 부분을 일치시켜주어야 하는 작업이 필요하다. 그래서 시스템의 효율성을 위해 다소간의 사용자 입력을 이용하는 방법이 제안되었다.

이렇게 재구성된 구조를 가상 시점에서 실제 이미지처럼 만들기 위해 여러 가지 방법이 사용되고 있으나 가장 널리 사용되고 있는 것이 텍스처 맵핑이다. 여기에는 영역 분할과 시점 가중치 방법이 있는데 전자의 경우는 영역을 삼각형 또는 이미지에 포함되어 있는 물체들의 구성 특성에 따라 나누어 하나의 이미지로부터 맵핑하는 방법이고, 후자의 방법은 여러 장의 이미지를 이용해 만들고자 하는 시점에 가까운 이미지에 가중치를 두어 맵핑하는 것이다.

본 연구의 목적은 모션 방법을 사용하여 여러 장의 서로 다른 관점의 사진을 이용하여 물체의 3차원 구조를 재구성하고, 여기에 텍스처 맵핑 기법을 사용하여 다른 시점의 이미지를 실제에 가깝게 재구성하는데 있다.

본문의 구성은 다음과 같다. 2장에는 프로젝티브 복원 방법에 대해 기술하였고, 3장에는 유클리디언 복원 방법에 대해 기술하였으며, 4장에는 시스템의 설계와 구현에 대해 기술하였다. 마지막으로 결론 부분에는 구현 결과 및 성과와 앞으로 해결해야 할 과제에 대해 언급하였다.

## 2. 프로젝티브 복원 방법

### 2.1 프로젝티브 복원문제

$m$ 개의 카메라(또는 관측점)에서 모두 관측가능한  $n$ 개의 3차원 점을 생각하자. 이러한

영상좌표의 측정값  $(u_{ij}, v_{ij})$ 만을 이용하여, 각 카메라의 원근투영행렬(projection matrix)  $P_i$ 와 각 영상점의 3차원 위치  $X_j$ 를 복원하려는 것이 프로젝티브 복원(projective reconstruction) 문제이다 (단  $i=1, m, j=1, n$ ). 이상의 내용을 동차좌표(homogeneous coordinate)를 사용하여 식으로 표현하면 다음과 같다.

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} \sim P_i X_j \tag{1}$$

위의 식의 동등한 표현은 다음과 같이 쓸 수 있다.

$$\lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = P_i X_j \tag{2}$$

여기서  $\lambda_{ij}$ 는 프로젝티브 깊이(projective depth) 또는 배율(scale factor)이라 부른다. 위의 식을 모든 카메라와 모든 점에 대해 종합하면 다음과 같다.

$$W_s = \begin{bmatrix} \lambda_{11} \begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \dots & \lambda_{1n} \begin{bmatrix} u_{1n} \\ v_{1n} \\ 1 \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \lambda_{m1} \begin{bmatrix} u_{m1} \\ v_{m1} \\ 1 \end{bmatrix} & \dots & \lambda_{mn} \begin{bmatrix} u_{mn} \\ v_{mn} \\ 1 \end{bmatrix} \end{bmatrix} \tag{3}$$

또는

$$W_s = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} [X_1 X_2 \dots X_n] \tag{4}$$

위에서 행렬  $W_s$ 의 차원은  $3m \times n$ 이고  $P_i$ 는  $3m \times 4$ ,  $X_j$ 는  $4 \times n$ 이다. 이때 행렬  $W_s$ 의 랭크는 4보다 크지 않다(단  $3m$ 과  $n$ 이 4보다 클 경우).

## 2.2 반복 프로젝티브 복원 알고리즘

앞에서 언급한 프로젝티브 복원 방법은 여러 논문에서 언급되었고 잘 알려져 있는 방법이다. 본 연구에서는 Chen과 Medioni의 프로젝티브 복원 알고리즘[3,4]을 바탕으로 일부

수정한 방법을 사용하였는데 이를 설명한 주요 단계는 다음과 같다.

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

위 알고리즘에서 영상좌표의 정규화는 참고문헌 [1]의 431 페이지에 자세히 언급되어 있듯이 안정적인 계산에 필수적이다. 프로젝티브 복원의 주 목적은 식 (3)을 일관되게 만드는  $\lambda_{ij}$ 를 구하는 것이라 할 수 있다.

$$W_s = UDV^T \quad (5)$$

위와 같이 decompose한 행렬로부터 프로젝티브 운동행렬과 형상벡터를 구할 수 있는 방법은 다음과 같은 세 가지 방법을 고려할 수 있다.

$$P = UD, \quad X = V^T \quad (6)$$

$$P = U\sqrt{D}, \quad X = \sqrt{D}V^T \quad (7)$$

$$P = U, \quad X = DV^T \quad (8)$$

위의 세 가지 방법 중 한 방법을 택해 프로젝티브 복원을 행한다.

### 3. 유클리디안 복원

프로젝티브 복원을 한 후 그것으로부터 유클리디안 복원을 행하기 위해서 Kanade와 Han의 방법[2]을 사용하였다.

$$W_s = \hat{P}\hat{X} = PHH^{-1}X = PX \quad (9)$$

위 식에서  $\hat{P}\hat{X}$ 는 프로젝티브 복원을 의미하며  $PX$ 는 구하려고 하는 유클리디안 복원을 뜻한다. 임의의 4x4 비특이행렬(non-singular matrix) H가 P와 X사이에 들어가더라도 위 식은 성립하며, 이 경우 또 다른 운동과 형상의 쌍이 된다. 카메라의 skew가 0일 때,

앞에서 설명한 프로젝티브 복원방법으로 구한 투영행렬과 형상 행렬에 metric constraints 를 추가하여 카메라 내부변수(intrinsic parameters)와 선형변환 H를 동시에 구하는 것이 Kanade와 Han이 사용한 유클리디안 복원 방법의 개요이다. 자세한 내용은 참고문헌 [2]에 기술되어 있으며 본 연구에서는 초점거리만 미지수인 경우에 대해 구현하였다.

## 4. 구현 및 실험

### 4.1 구현

본 시스템은 PC에서 C 언어로 구현하였으며 많이 사용되는 특이값 분해는 참고문헌[6]을 참조하였다. 단 참고 문헌 [6]에 제시된 프로그램 코드가 항상 특이값이 큰 것부터 작은 것 순서로 정렬된 결과를 출력하지는 않기 때문에 수정이 필요한 것이 주의해야 할 점이다. 그래픽 출력 프로그램 부분은 필요에 따라 VRML 코드 또는 OpenGL을 이용하여 프로그래밍 하였다.

### 4.2 실험

개발된 시스템의 효용성을 검증하기 위해 모의데이터 및 실제데이터를 사용하여 실험하였다.

#### 4.2.1 모의 데이터

OpenGL을 이용하여 카메라의 위치를 변경시키면서 10개의 다른 관측점에서 바라본 정육면체의 8 꼭지점의 화면좌표를 이용하여 각 꼭지점의 3차원 좌표를 복원하였다.(실제로는 정육면체의 7 꼭지점만 관측가능하나 컴퓨터 시뮬레이션 데이터이기 때문에 8 꼭지점이 가능하다.) 계산된 3차원 좌표는 다음과 같고, 좌표계의 원점은 임의로 계산되어 나온다. 물론 그 중 임의의 한점을 기준 좌표계의 원점으로 삼아 그것에 대한 좌표값의 형태로 계산할 수도 있다.

```
-0.070323 -0.069735 -0.081805
-0.070749 0.069038 -0.082861
0.068780 0.069424 -0.085179
0.068436 -0.070120 -0.084115
-0.068413 -0.069071 0.056579
-0.068830 0.069722 0.056290
0.070727 0.070112 0.054747
0.070372 -0.069453 0.055036
```

복원된 데이터를 이용하여 계산한 정육면체 12변의 각각의 길이는 다음과 같다.

```
0.138778, 0.139549, 0.139548, 0.138779, 0.138794, 0.139566, 0.139566, 0.138794, 0.138399,
```

0.139166, 0.139941, 0.139166

복원된 변 길이의 참값은 알 수 없으므로 평균값을 구하여 그 값에 대한 오차를 계산하였다. 표1에 평균 오차와 최대, 최소 오차를 기록하였다.

표 1. 계산된 변의 오차 (%)

평균 오차	0.278
최대 오차	0.555
최소 오차	0.004

다음은 복원된 좌표를 바탕으로 각 정점에서 두 개의 변이 이루는 총 24개 각도의 값을 계산하였다. 계산된 각도의 값은 다음과 같다.

89.989867, 89.693044, 89.989546, 90.327543, 89.989292, 89.692853, 89.989354, 90.328500, 90.163481, 90.153222, 89.840057, 89.843240, 90.160885, 89.846203, 89.835485, 90.157427, 90.160245, 90.158005, 89.834918, 89.846833, 89.842595, 89.840640, 90.153014, 90.163751

이 경우에는 참값은 90도이므로 그 값에 대한 오차를 계산하여 표 2에 평균 오차와 최대, 최소 오차를 기록하였다. 이 표에서 알 수 있듯이 계산된 결과가 상당히 정확함을 알 수 있다.

표 2. 계산된 각의 오차(단위: 도)

평균오차	0.160506
최대오차	0.327543
최소오차	0.010133

#### 4.2.2 터미널 데이터

실제 이미지를 이용한 실험 데이터 중 첫번째는 Chen의 박사학위 논문[3]에서 사용된 컴퓨터 터미널 데이터(그림 1 참조)를 사용하였다. 이 데이터는 총 8 프레임이며, 18개의 특징점을 수작업으로 정합시켜 3차원 데이터의 복원을 시도하였다.

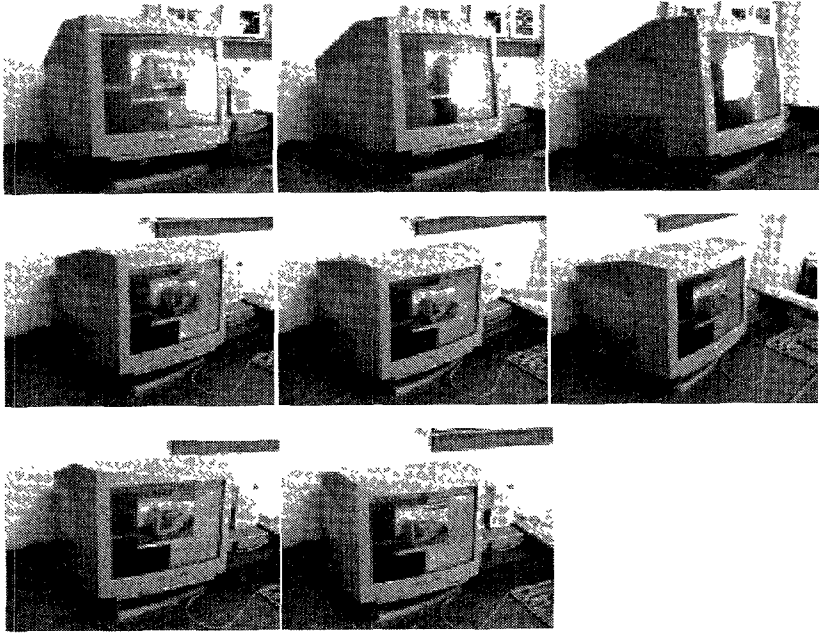


그림 1. Chen의 터미널 데이터

복원된 18개의 점만을 도시해서는 물체의 형상을 알아보기가 쉽지 않기 때문에 각 점간의 상호관계를 알아보기 쉽도록 VRML code를 이용하여 각 점들간의 연결관계를 나타내는 선분을 추가하였다. 그림 2, 3에는 서로 다른 두 관측점에서 바라본 복원된 터미널을 나타내었다. 터미널 물체의 3차원 형상이 실제와 매우 가깝게 복원된 것을 육안으로 확인할 수 있다.

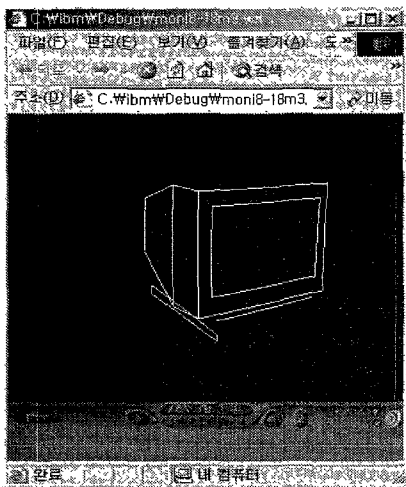


그림 2. 복원된 터미널 데이터

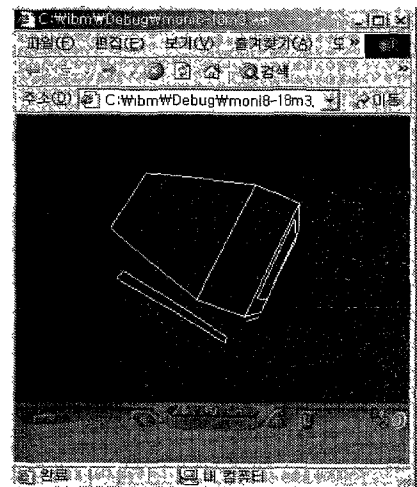


그림 3. 다른 각도에서의 터미널 데이터

### 4.2.3 MOVI 하우스 데이터

구현된 방법이 특징점이 매우 많은 데이터에 대해서도 잘 적용되는 지 확인하기 위해 여러 논문에서 사용된 그림 4와 같은 집 모양의 하우스 데이터[5]를 이용하였다. 이러한 종류의 데이터에 앞서와 같이 수작업으로 많은 특징점들을 추적하고 정합시키는 일은 시간이 많이 걸리고 오차가 생길 확률이 크다. 따라서 특징점의 자동 추출 및 추적기가 필요하다. 현재 알려진 방법 중 가장 많이 사용되고 안정적인 방법이 KLT 트랙커[8-10]이다. KLT 트랙커를 사용하기 위해서는 프레임 간 운동이 적어야 한다. 사용된 프레임 수는 총 6개이며 여섯개의 프레임에 대해 KLT 트랙커를 이용하여 200개의 특징점을 추출하여 6개 프레임 처음부터 마지막 프레임까지 모두 추적된 총 191개의 점에 대해 유클리디언 복원을 시도하였다.

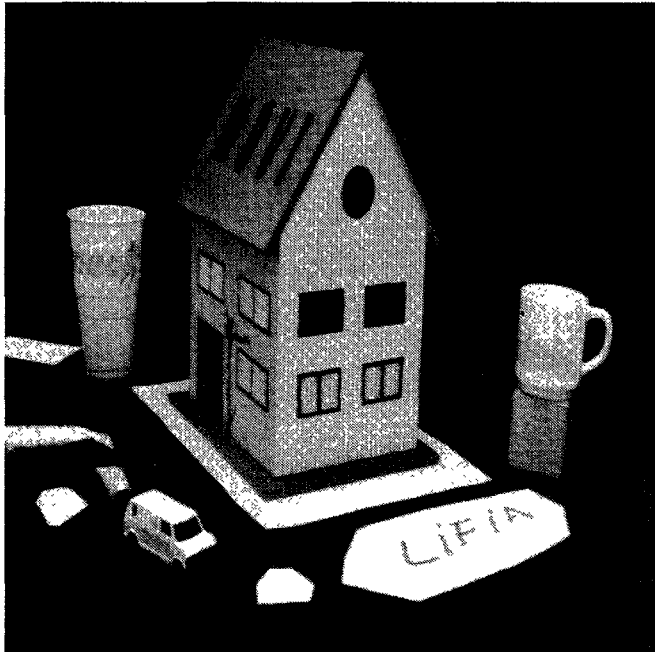


그림 4. 사용된 MOVI House 데이터의 한 예

복원된 3차원 좌표를 OpenGL 프로그램을 이용하여 작은 구의 형태로 표시한 결과가 그림 5와 같다. 사물이 점만으로 구성되어 물체의 구조를 확인하기가 쉽지 않으나 OpenGL로 작성한 대화식 프로그램에서 관측점을 계속 바꾸면서 확인해 본 결과 3차원 형상이 실제와 가깝게 복원된 것을 확인할 수 있었다.



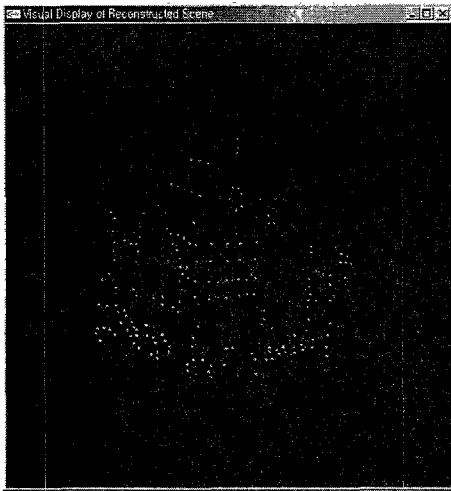


그림 5. 복원된 House 데이터

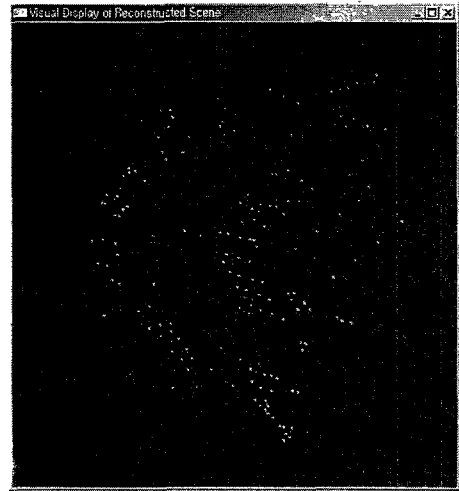


그림 6. 다른 관측점에서의 데이터

그림 6은 관측점을 다르게 하여 바라본 3차원 데이터 값이다. 건물모형의 앞면과 옆면이 이루는 각도가 거의 90도에 가까운 것을 확인할 수 있다.

#### 4.2.4 텍스처 맵핑 - 터미널 데이터

3차원 물체의 사실적인 복원을 위해서는 텍스처 맵핑 방법을 하는 것이 좋다. 단 텍스처 맵핑을 하기 위해서는 다음과 같은 몇 가지 사전 준비 작업이 필요하다.

1. 우선 OpenGL의 텍스처 맵핑에서 필요한 이미지 데이터의 가로와 세로의 크기를 2의 제곱의 형태로 준비한다.
2. 텍스처 맵핑에 필요한 화면좌표값을 알기 위해 카메라 캘리브레이션 행렬도 미리 계산, 준비한다.
3. 텍스처 맵핑을 할 부분에 대해 해당되는 3차원 좌표값과 사용할 텍스처 이미지를 일일이 지정해 주어야 한다.

이와 같은 절차를 거쳐야 하므로 텍스처 맵핑 작업은 아직까지는 매우 번거롭다. 그림 7은 앞에서 계산된 컴퓨터 터미널 데이터 중 앞의 화면부분에 텍스처 맵핑을 한 결과이다. 그림 2 또는 그림 3의 복원 결과와 비교하면 매우 사실적인 재구성이 가능함을 알 수 있다.

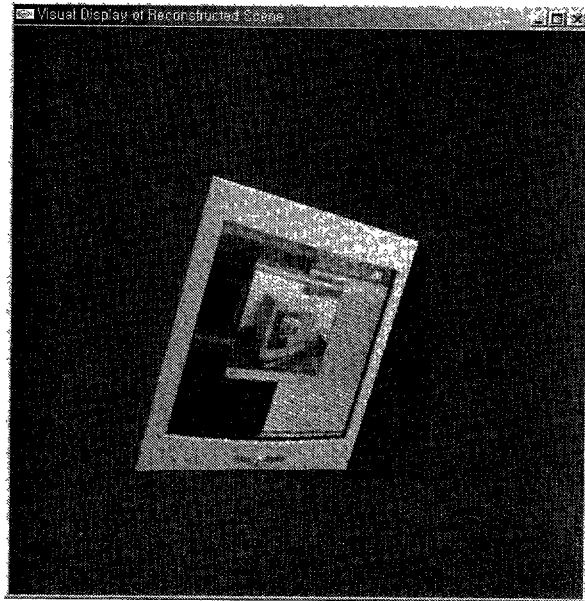


그림 7. 복원된 3차원 터미널 데이터의 화면부분에 텍스처 맵핑을 한 결과

## 5. 결 론

본 연구에서는 컴퓨터 비전 분야의 기술과 컴퓨터 그래픽스 분야의 기술을 결합하여, 3차원 물체를 촬영한 여러 장의 이미지로부터 그 이미지에 포함된 3차원 물체의 형상을 복원하는 유클리디언 복원방법을 구현하였으며 수작업이 많이 요구되기는 하지만 간단한 텍스처 맵핑 또한 구현하였다. 구현된 유클리디언 복원 시스템의 주요 특징은 다음과 같다.

첫째, 초기값에 구애받지 않고 안정적인 해를 구할 수 있다.

둘째, 빠른 계산이 가능하다.

셋째, 텍스처 맵핑 방법이 조금 더 쉽게 사용될 수 있게 구현되면 매우 사실적인 3차원 모델을 복원할 수 있다.

터미널 데이터와 House 데이터의 참값을 알 수 없으므로 다른 논문과의 정량적인 비교가 가능하지 않은 것이 아쉬우나, 다른 방법[3][11]과 거의 유사한 결과를 얻은 것을 성과로 생각한다. 추후 계속하려고 하는 연구 과제는 다음과 같다.

1. 다양한 시뮬레이션 데이터의 확보 및 실험
2. 참값을 알 수 있는 물체의 이미지 데이터를 활용한 유클리디언 복원
3. 보다 간편하고 정확한 특징점의 추출과 정합 및 추적 방법에 대한 연구
4. 손쉬운 텍스처 맵핑 방법에 대한 연구

## 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. [http://www.inrialpes.fr/movi/pub/Images/sequences/g3\\_vp\\_ra\\_s1/index.html](http://www.inrialpes.fr/movi/pub/Images/sequences/g3_vp_ra_s1/index.html) ( MOVI house data)
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. Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.
10. Jianbo Shi and Carlo Tomasi. Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.
11. Peter Sturm and Bill Triggs, "A factorization based algorithm for multi-image projective structure and motion", Proc. ECCV, 1996