

연속적인 이미지를 이용한 3차원 장면의 사실적인 복원

전 희 성[†]

요 약

본 연구에서는 여러 이미지를 이용하여 사실적인 3차원 장면의 모델을 얻는 방법이 구현되었다. 이미지는 파라미터를 모르는 카메라를 이용하여 여러 위치에서 획득한 것을 사용하였다. 먼저 특징점 추출 및 추적 방법을 사용하여 모든 이미지에 대한 대응점들을 구하고 이 점들을 사용하여 사영복원을 구한다. 그 다음 사영 복원된 값에 여러 제약조건을 사용하여 유클리디언 복원을 하면 특징점들의 3차원 좌표값이 계산된다. 이 좌표값을 이용하여 삼각형 매쉬를 구한 후 이 면에 텍스처 맵핑을 하면 사실적인 복원이 완성된다. 전체 시스템은 C++언어로 구현하였으며, 사용자 인터페이스는 Qt 라이브러리로, 텍스처 맵핑과 모델 가시화 부분은 OpenGL 그래픽스 라이브러리로 구현하였다. 구현된 시스템의 효율성을 보이기 위해 모의 데이터와 실제 이미지 데이터를 이용하여 실험한 결과를 포함하였으며 만족할 만한 복원 결과를 얻을 수 있었다.

키워드 : 영상기반 모델링, 사영 복원, 유클리디언 복원, 움직임에 이용한 복원, 텍스처 맵핑

Realistic 3D Scene Reconstruction from an Image Sequence

Jun, Heesung[†]

ABSTRACT

A factorization-based 3D reconstruction system is realized to recover 3D scene from an image sequence. The image sequence is captured from uncalibrated perspective camera from several views. Many matched feature points over all images are obtained by feature tracking method. Then, these data are supplied to the 3D reconstruction module to obtain the projective reconstruction. Projective reconstruction is converted to Euclidean reconstruction by enforcing several metric constraints. After many triangular meshes are obtained, realistic reconstruction of 3D models are finished by texture mapping. The developed system is implemented in C++, and Qt library is used to implement the system user interface. OpenGL graphics library is used to realize the texture mapping routine and the model visualization program. Experimental results using synthetic and real image data are included to demonstrate the effectiveness of the developed system.

Keywords : Image-based Modeling, Projective Reconstruction, Euclidean Reconstruction, Structure From Motion, Texture Mapping

1. 서 론

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 방

법으로 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. 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.

* 이 논문은 2002년 울산대학교의 연구비에 의하여 연구되었음.

† 정회원 : 울산대학교 컴퓨터정보통신공학부 교수

논문접수 : 2009년 11월 14일

수정일 : 1차 2010년 1월 12일, 2차 2010년 3월 15일

심사완료 : 2010년 3월 19일

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.

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]. Han과 Kanade[2] 그리고 Chen과 Medioni[3, 4], 국내에서는 김성용과 한준희[12]가 사전에 보정하지 않은 카메라로 찍은 사진을 이용하여 물체의 형상을 복원하는 방법을 연구하였다. 이 방법은 실험실에서 카메라에 대한 특성을 미리 파악하지 않고도 찍힌 사진 또는 연속영상을 이용하여 3차원 데이터를 복원할 수 있다.

본 연구는 보정되지 않은 카메라로부터 획득된 연속적인 이미지에 대하여 사영 복원과 유클리디언 복원을 수행하여 3차원 모델 데이터를 구성하고, 여기서 얻어진 데이터를 보다 사실적으로 표현하기 위해 텍스처 맵핑을 사용하여 물체의 사실적인 3차원 모델을 구현하였다.

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

물체의 3차원 데이터 복원은 아래에서 설명하는 사영 복원(Projective Reconstruction)과 유클리디언 복원(Euclidean Reconstruction)을 통해 이루어진다.

사영 복원이란 운동(Motion)과 형상(Shape)을 구하는 복원으로, 그 좌표계가 단순히 직교좌표계에 있지는 않으므로 그 결과물로는 모델링이 불가능하다. 그러나, 특정 조건을 이용하여 유클리디언 복원을 수행함으로써 그 좌표계를 우리가 흔히 생각할 수 있는 x, y, z축으로 구성된 3차원 직교좌표계(유클리디언 좌표계)로 전환시킴으로 그 복원을 완성할 수 있다.

2.1 사영 복원(Projective Reconstruction)

m개의 카메라(또는 관측점)에서 모두 관측 가능한 n개의 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 사영 복원 문제이다. 이 문제를 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.

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

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

위의 식 (2)에서 W_s 가 $3m \times 4$ 행렬과 $4 \times n$ 행렬의 곱이기 때문에 W_s 의 rank는 기껏해야 4임을 주목해야 한다.

2.2 반복 사영 복원

앞에서 언급한 사영 복원 방법은 여러 논문에서 언급되었고 잘 알려져 있는 방법이다. 본 연구에서는 Han과 Kanade[2]의 방법과 Chen과 Medioni의 사영 복원 알고리즘[3, 4]을 바탕으로 다음과 같이 일부 수정한 방법을 사용하였다. 수정된 부분은 단계 1의 영상좌표 정규화 부분과 단계 4의 계산 반복 결정 부분이다. 영상좌표의 정규화는 참고문헌 [1]의 17장 3절에 자세히 언급되어 있듯이 안정적인 계산에 필수적이다.

1. 영상좌표를 정규화(좌표의 행과 열의 값을 -0.5~0.5로 함)하여 초기 행렬을 계산
2. 식 (2)을 이용하여 초기의 행렬에 배율 λ_{ij} 를 곱하여 측정행렬 W_s 를 계산 (초기값은 $\lambda_{ij} = 1$)
3. 특이값 분해방법(SVD, singular value decomposition) [6]을 이용하여 W_s 의 Rank4 Factorization을 행함
4. W_s 를 특이값 분해했을 때 4번째 특이값과 5번째 특이값의 차이가 미리 정한 범위에 들어올 때까지 단계 2로 돌아가 계산을 반복함.
5. W 를 특이값 분해한 결과인 UDV^T 를 이용하여 P와 X를 복원한다.

사영 복원의 목적은 식 (2)의 행렬 W_s 의 Rank를 4가 되도록 만드는 λ_{ij} 를 구하는 것이다.

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

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

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

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

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

2.3 유클리디언(Euclidean) 복원

사영 복원을 통해 운동(motion)과 형상(shape)에 대한 P와 X를 구하였다. 우리가 원하는 유클리디언 형태의 P와 X를 구하기 위해 $P = PH$, $X = H^{-1}X$ 로 치환한 후 특정 제약조건을 만족시키는 H를 구함으로써 유클리디언 복원을 수행할 수 있다. (본 절의 대부분은 참고문헌 [2]의 내용에서 전제한 것이다.)

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

사영 운동 행렬(Projective motion matrix) \underline{P} 는

$$\underline{P} = [M | T] \quad (8)$$

이고, 여기서

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

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

그리고,

$$m_{xi} = \mu_i f_i i_i + \mu_i u_{0i} k_i$$

$$m_{yi} = \mu_i f_i j_i + \mu_i v_{0i} k_i$$

$$m_{zi} = \mu_i k_i \quad (9)$$

$$T_{xi} = \mu_i f_i t_{xi} + \mu_i u_{0i} t_{zi}$$

$$T_{yi} = \mu_i f_i t_{yi} + \mu_i v_{0i} t_{zi}$$

$$T_{zi} = \mu_i t_{zi}$$

이다. 다시 정리하면

$$P = \underline{P}H = K_i [R_i | t_i] = \underline{P}[A|B] = [M|T] \quad (10)$$

$$\underline{P}A = M, \quad \underline{P}B = T \quad (11)$$

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

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

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

$$x_j = [v_j s_j^T v_j]^T$$

이다. 여기서 세계좌표계(world coordinate system)의 원점이 특징점들의 무게중심으로 되도록 기준점을 정하면,

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

이 되며, 이것으로 다음과 같은 결과를 얻을 수 있다.

$$\sum_{j=1}^m \lambda_{ij} u_{ij} = \sum_{j=1}^m (m_{xi} \cdot v_j s_j + v_j T_{xi})$$

$$= m_{xi} \cdot \sum_{j=1}^m v_j s_j + T_{xi} \sum_{j=1}^m v_j = T_{xi} \sum_{j=1}^m v_j$$

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 구할 수 있다.

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

$$MM^T = PAA^T P^T, \quad (16)$$

$$MM^T = PQ P^T$$

$$\begin{bmatrix} m_{xi} \\ m_{yi} \\ m_{zi} \end{bmatrix} [m_{xi} m_{yi} m_{zi}] = P \begin{bmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{bmatrix} P^T$$

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

식 (17)에서 우리는 각 관측점 당 10개의 선형방정식을 만들 수 있다. 여기서 이미지의 중점을 (0, 0)으로, 배율(scale factor) μ_i 를 1이라 가정하면,

$$|m_{xi}|^2 = |m_{yi}|^2$$

$$m_{xi} \cdot m_{yi} = 0,$$

$$m_{xi} \cdot m_{zi} = 0,$$

$$m_{yi} \cdot m_{zi} = 0 \quad (18)$$

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 랭크 3 행렬분해를 하여 A를 구할 수 있다. 단 A값이 존재하는 Q값을 구해야 하는데, 이것은 최소자승법으로 구해진 여러 해를 조합하여 조건에 해당되는 답을 선택하여야 한다.

최소자승법으로 구한 A가 식 $Q = AA^T$ 를 항상 만족하지는 않기 때문에, 여기에 가장 적절한 해를 찾아야 한다. Q의 행렬식(determinant) 값이 0이 되기 때문에, 최소자승법에 의한 최적의 두 개의 해 Q_1 과 Q_2 에서 $Q = Q_1 + bQ_2$ 로 놓고 $|Q| = 0$ 을 만족하는 b값을 구하면 $Q = Q_1 + bQ_2$ 가 된다. 이렇게 하여 최적의 Q값을 구할 수 있다. 우리가 구하고자 하는 A값은 Q에 대해서 랭크3 행렬분해를 수행하여 얻을 수 있다. 최종적으로 3차원 좌표값들은 다음 식에서 구한다.

$$X = H^{-1}X$$

3. 시스템 실험 및 구현

개발된 시스템의 효용성을 검증하기 위해 여러 차례의 모의데이터를 이용한 실험을 거쳐 3차원 복원시스템을 완성한 후, 실제 이미지 데이터를 사용하여 검증하였으며, 그다음 사실적인 복원을 위한 텍스처 맵핑을 구현하고 실제 영상데이터를 사용하여 실험하였다.

3.1 시스템의 실험 및 단계적 구현

3.1.1 정육면체에 대한 모의실험

OpenGL을 이용해 카메라의 위치를 물체를 중심으로 일정한 반경으로 회전시키면서 10개의 서로 다른 관측점에서 바라본 정육면체의 8 꼭지점의 화면좌표(유효소수점 6자리)를 이용하여 각 꼭지점의 3차원 좌표를 계산하였다. 실제 정육면체는 최대 7 꼭지점만 관측가능하나 컴퓨터 시뮬레이션 데이터이기 때문에 8 꼭지점이 가능하다. 스케일의 복원은 불가능하므로, 계산된 각 변의 평균값에 대한 오차를 계산한 결과를 <표 1>에 제시하였다. 최대 오차가 0.56%로 매우 정확한 결과를 얻었다. 또한 복원된 변을 이용하여 각 꼭지점의 각도를 계산하였다. 각도도 최대 오차가 0.33도로 매우 정확한 결과를 얻었다.

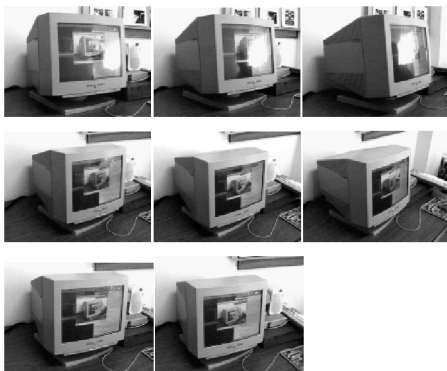
<표 1> 정육면체에 대한 복원결과와 오차
<Table 1> Reconstruction error for cube data

변(12개)	평균오차	0.278 %
	최대오차	0.555 %
각(24개)	평균오차	0.16°
	최대오차	0.33°
	최소오차	0.01°

3.1.2 터미널 데이터

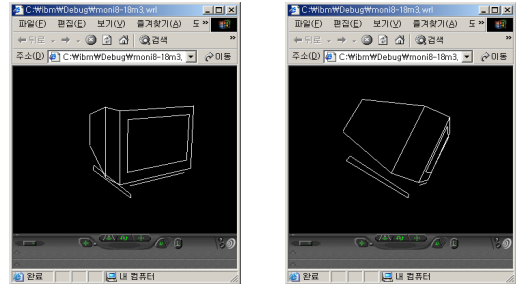
Chen의 박사학위 논문[3]에서 사용된 컴퓨터 터미널 이미지(그림 1 참조)를 사용하여 8 프레임의 18개의 특징점을 정합시켜 3차원 데이터를 복원하였다.

복원된 18개의 점만을 도시해서는 물체의 형상을 알아보



(그림 1) Chen의 컴퓨터 터미널 이미지
(Fig. 1) Chen's computer terminal images

기가 어렵기 때문에 각 점간의 상호관계를 알아보기 쉽도록 VRML을 이용하여 각 점들간의 연결관계를 나타내는 선분을 추가하였다. 서로 다른 두 관측점에서 바라본 복원된 터미널을 (그림 2)에 나타내었다. 터미널 물체의 3차원 형상이 실제와 매우 가깝게 복원된 것을 육안으로 확인할 수 있다.



(그림 2) 서로 다른 곳에서 본 복원된 터미널 데이터
(Fig. 2) Recovered terminal data from two views

3.1.3 실제 영상데이터를 이용한 실험

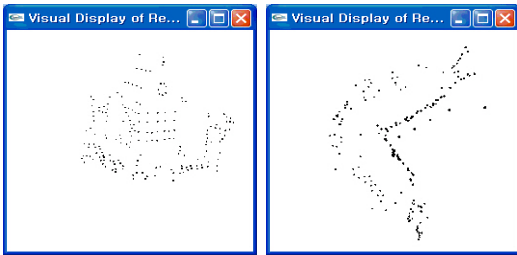
구현된 방법이 특징점이 매우 많은 영상데이터에 대해서도 잘 적용되는 지 확인하기 위해 여러 논문에서 사용된 (그림 3)과 같은 집 모양의 데이터[5] 다섯 프레임을 이용하였다. 영상데이터에 수작업으로 많은 특징점들을 추적하고 정합시키는 일은 시간이 많이 걸리고 오차가 생길 확률이 크다. 따라서 특징점의 자동 추출 및 추적이 필요하며, 프레임 간 운동이 작을 때 사용되는 방법인 Kanade, Lucas, Tomasi에 의해 고안된 KLT 트랙커[8, 9]를 사용하였다. 본 프로그램에서 트랙커의 구현은 [11]에 제시된 프로그램을 변형하여 사용하였다.

각 프레임에 대해 KLT 트랙커를 이용하여 400개의 특징점을 추출하여 첫 프레임부터 마지막 프레임까지 모두 추적된 점 191개의 3차원 좌표값을 구할 수 있었다. 이 데이터들을 OpenGL 라이브러리를 이용하여 서로 다른 두 곳에서 가시화한 것을 (그림 4)에 나타내었으며 오른쪽 그림은 지붕의 윗 부분에서 바라본 것이다. 집의 두 벽면의 각도가 90도에 근접함을 확인할 수 있다.

앞에서 제시한 (그림 2) 또는 (그림 4)의 가시화 방법은 복원된 데이터의 3차원 형상을 유추할 수는 있으나, 사실적인 복원과는 거리가 있기 때문에 텍스처 맵핑까지 할 수 있는 프로그램을 개발하였다.



(그림 3) Movi 집 데이터 중 한 프레임
(Fig. 3) One frame of Movi house data



(그림 4) 서로 다른 곳에서 복원된 Movi 모형 집 데이터
(Fig. 4) Recovered Movi house data from two views

3.2 사실적인 복원시스템의 구현

3.2.1 3차원 복원 프로그램의 구현

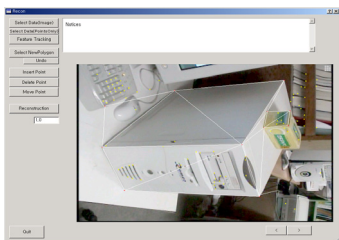
본 시스템은 C++언어를 사용하여 구현하였으며. 많이 사용되는 특이값 분해는 참고문헌[6]을 참조하였고, 사용자 인터페이스 부분은 Qt 라이브러리[10], 그래픽 부분은 OpenGL[7]을 이용하여 작성하였다. (그림 5)에 3차원 복원 프로그램의 사용자 인터페이스를 나타내었다.

구현된 프로그램의 주요 기능은 다음과 같다.

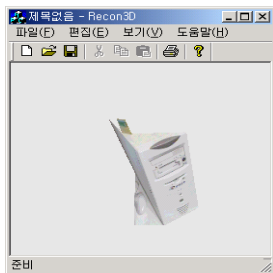
1. 순차 이미지 또는 정합된 수치 데이터를 불러 오
2. 특징점의 추적
3. 특징점의 선택, 조정 및 삼각형 폴리곤 구성
4. 사영 복원 및 유클리디언 복원
5. 텍스처 맵핑을 사용하여 복원된 구조의 시각화

구현된 프로그램의 주요 기능 구현에 대한 상세한 설명은 아래에 기술한다.

또한 임의의 관측점에서 바라본 물체를 표현하기 위해 모델 관측 프로그램을 MFC와 OpenGL을 이용하여 작성하였으며 (그림 6)에 모델 관측 프로그램의 인터페이스를 나타내었다.



(그림 5) 3차원 복원 프로그램 인터페이스
(Fig. 5) Reconstruction program interface



(그림 6) 전용 모델 관측 프로그램
(Fig. 6) Model visualization program

구현된 프로그램을 사용하면 특징점에 해당하는 점들이 각 프레임에 동일하게 표시된다. 이들 중 첫 프레임에 특징점을 표시한 것을 (그림 7)에 나타내었다.

사실적 복원 프로그램에서는 KLT 트래커의 방법을 변형하여 구현하였다. 우선 프로그램에 의해 일치점이 자동으로 찾아지게 되는 데 그 중 정확히 일치하지 않거나 불필요한 점은 제외시킬 수 있으며, 정확하지 않은 특징점은 사용자가 수동으로 몇 픽셀 이동시킬 수 있게 하였다.

추적된 모든 점에 대해 3차원 좌표값이 필요하다면 앞에서 설명한 사영 복원과 유클리디언 복원 프로그램을 이용해 3차원 좌표값을 계산한다. 만약 매우 사실적인 3차원 모델이 필요하다면 몇 개의 특징점에 의해 결정되는 평면에 텍스처 맵핑을 하는 것이 더욱 효과적일 수 있다.

(그림 8)에서 보듯이 화면상에서 적절한 세 개의 점의 집합을 선택하면 그에 따라 삼각평면이 화면에 표시되는 데, 이 평면이 포함하는 이미지 부분이 추후 텍스처 맵핑에 사용된다.

임의의 삼각형 평면에 대응되는 이미지는 서로 다른 관측점에서 관측가능하다면 어느 것이라도 텍스처 맵핑에 사용할 수 있다. 그러나 이미지의 크기가 작은 경우에는 복원 모델의 화질이 저하되므로, 여러 관측점 중 가장 넓은 면적을 가지고 있는 이미지를 선택하여 텍스처 맵핑이 되도록 구현하였다.



(그림 7) 구현된 프로그램에 의해 표시된 특징점
(Fig. 7) Feature points recovered

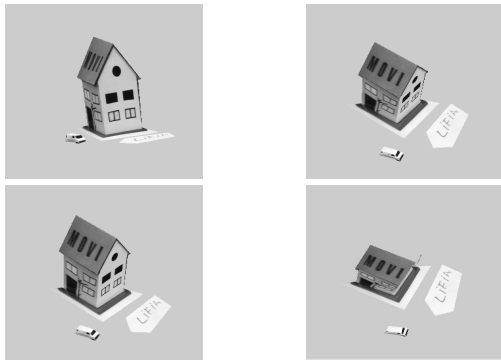


(그림 8) 복원을 위한 특징점과 면 구조의 설정
(Fig. 8) Feature points & faces for reconstruction

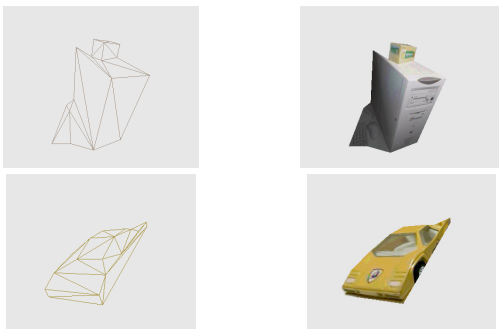
3.2.2 사실적인 복원 결과

(그림 8)에 나타낸 것과 같이 3개의 물체(집, 자동차, LiFiA판)에 대해 텍스처 맵핑을 사용하여 사실적인 복원을 시도하였다. 집, 자동차, LiFiA판에 각각 14개, 7개, 3개 - 합계 24개의 삼각형이 사용되었다.

텍스처 맵핑 결과 3차원 물체의 매우 사실적인 복원 모델을 얻을 수 있었다. (그림 9)에 복원된 3차원 모델을 서로



(그림 9) 여러 각도에서 바라본 복원된 3차원 모델
(Fig. 9) Reconstructed 3D model from different views



(그림 10) 컴퓨터와 장난감 자동차의 복원결과
(Fig. 10) Reconstructed computer and toy car

다른 4곳의 관측점에서 바라본 결과를 나타내었다.

실제 물체를 촬영하여 얻은 이미지 데이터를 이용하여 복원하기 위해 (그림 10)에 보인 것처럼 컴퓨터와 장난감 자동차를 이용하여 복원한 결과를 나타내었다. 매우 사실적인 복원이 이루어진 것을 확인할 수 있다.

4. 결 론

본 연구에서는 3차원 물체를 촬영한 연속적인 영상데이터로부터 3차원 물체의 형상을 복원하는 유클리디언 복원방법을 구현하였으며, 텍스처 맵핑을 사용하여 사실적인 3차원 복원을 수행하는 프로그램을 구현하였다. 구현된 시스템의 주요 특징은 다음과 같다.

첫째, 특징점들을 KLT를 통해 자동으로 추출하고, 사용자가 사실적인 복원에 필요한 몇 개의 특징점들을 대화식으로 선택할 수 있다.

둘째, 텍스처 맵핑이 프로그램에서 자동으로 이루어져, 모델링할 때의 수작업이 많이 줄어들었다.

향후에 수행하고자 하는 연구과제는 다음과 같다.

1. 현재 사용하고 있는 KLT 트래커보다 나은 SIFT 트래커 [13] 등의 방법을 사용하여 특징점의 추출, 정합 및 추적 방법을 보다 간편하고 정확하게 구현할 필요가 있다.
2. 현재까지 구현된 부분은 3차원 물체의 한쪽 부분만을 복원할 수 있다. 3차원 물체의 이면까지를 포함한 모

든 부분을 모델링할 수 있는 방법의 구현이 필요하다.

참 고 문 헌

- [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, pp.674-679, 1981.
- [9] Jianbo Shi and Carlo Tomasi, "Good Features to Track," IEEE Conference on Computer Vision and Pattern Recognition, pp.593-600, 1994.
- [10] Qt Library (<http://www.trolltech.com/>)
- [11] (<http://www.ces.clemson.edu/~stb/klf/>) KLT : An Implementation of the Kanade-Lucas-Tomasi Feature Tracker
- [12] 김성용, 한준희, "계약 조건을 이용한 셀프 캘리브레이션 방법," 정보과학회논문지:소프트웨어 및 응용 제28권 제4호 pp. 358-368, 2001년 4월.
- [13] D. Lowe, "Distinctive image features from scale-invariant key-points," International Journal of Computer Vision, 60(2), 91-110, 2004.



전 희 성

e-mail : hsjun@mail.ulsan.ac.kr

1981년 서울대학교 전기공학과 졸업(학사)

1983년 서울대학교 전기공학과(공학석사)

1992년 미국 Rutgers University 컴퓨터공학과(공학박사)

1983년~1986년 금성반도체(주) 주임연구원

1992년~1993년 삼성전자(주) 통신연구소 수석연구원

2001년 미국 University of California, Irvine 방문교수

1993년~현재 울산대학교 컴퓨터정보통신공학부 교수

관심분야: 컴퓨터비전, 영상처리, 증강현실 등