개선된 직교분해기법을 사용한 구조의 빠른 복원 및 융합

윤종현[1], 박종승[2], 이상락[3], 노성렬[4]
인천대학교 컴퓨터공학과[1 2 3], 시리우스[4]
{jhyoon[1],jong[2],srlee[3]}@incheon.ac.kr, noh@sirius.co.kr[4]

Fast Structure Recovery and Integration using Scaled Orthographic Factorization

Jong-Hyun Yoon[1], Jong-Seung Park [2] , Sang-Rak Lee[3], Sung-Ryul Noh[4]
University of Incheon, Department of Computer Science and Engineering[1 2], SIRIUS Co., Ltd.[3]

요 약

본 논문에서는 비디오에서의 특징점 추적을 통해 얻은 2D 좌표를 이용한 3D 구조를 추정하는 방법과 네 점 이상의 공통점을 이용한 융합 방법을 제안한다. 영상의 각 프레임에서 공통되는 특징점을 이용하여 형상을 추정한다. 영상의 각 프레임에 대한 특징점의 추적은 Lucas-Kanade 방법을 사용하였다. 3D 좌표 추정 방법으로 개선된 직교분해기법을 사용하였다. 개선된 직교분해기법에서는 3D 좌표를 복원함과 동시에 카메라의 위치와 방향을 계산할 수 있다. 복원된 부분 데이터들은 전체를 이루는 일부분이므로, 융합을 통해 완성된 모습을 만들 수 있다. 복원된 부분 데이터들의 서로 다른 좌표계를 기준 좌표계로 변환함으로써 융합할 수 있다. 융합은 카메라의 모션에 해당하는 카메라의 위치와 방향에 의존된다. 융합 과정은 모두 선형으로 평균 0.5 초 이하의 수행 속도를 보이며 융합의 오차는 평균 0.1cm 이하의 오차를 보였다.

Keywords : Orthographic Factorization, 3D Structure Integration, Registration, Camera Pose Estimation
[1]

1. 서론

가상 세계의 효과적인 구현을 위해서는 실 세계 물체의 정확한 기하 데이터를 3 차원 가상세계에서 표현할 수 있는 물체의 3 차원 모델링 절차가 필요하다. 실물의 3D 모델을 획득하는 방법은 여러 가지가 있다. 3D 스캐너를 통해 간단히 얻을 수도 있으나, 데이터 획득 시 실내라는 공간적인 제약이 따를 뿐만 아니라 고가의 장비이기에 대중화가 될 수 없다는 것이 가장 큰 단점이다.

이를 보완하는 방법으로 본 논문에서는 특수한 3D 카메라가 아닌 범용 디지털 카메라를 통해 얻은 2 차원 영상을 통해 3 차원 복원을 하는 방법을 제시한다. 복원 방법으로 개선된 직교분해기법을 사용한다. 모양 복원과 함께 카메라의 모션 정보를 얻을 수 있는 방법으로 개선된 직교분해기법을 채택하였다. 개선된 직교분해기법은 비디오 영상 및 이미지 시퀀스의 2 차원 정보를 이용한 3 차원 형상을 추정하는 방법이다. 특징점 좌표를 이용한 분해기법을 통해 특징점에 대응되는 3 차원 모델의 형상과 카메라의 이동경로를 추정한다[1].

분해기법을 통해 추정된 데이터는 복원하려는 대상 객체의 일부분이다. 전체를 포함하는 형상 추정을 위해서는 부분 데이터를 융합하는 과정이 필요하다. 획득된 부분 데이터들은 서로 다른 좌표계를 갖기 때문에 모든 데이터를 기준이 되는 좌표계로 이동함으로써 융합할 수 있다.

2. 선행 연구

영상의 2 차원 정보를 사용해서 3 차원 추정을 하는 방법은 크게 보정(calibrated) 방법과 비보정

¶ 본 연구는 산업자원부 (MOCIE)의 지원으로
인천정보산업진흥원 (IIT)을 통하여 수행되었음.

(uncalibrated) 방법의 두 가지로 분류할 수 있다[2]. 보정 방법은 획득한 영상으로부터 Euclidean 좌표까지 형상 복원이 가능하다. 그러나 카메라 파라메터 정보가 없으면 복원에 적용할 수 없다. 비보정 방법은 카메라의 파라메터 정보 없이도 3 차원 모델링이 가능한 방법이다. 카메라 파라메터가 불변이라는 가정하에 카메라 파라메터를 고정시키고 촬영한 이미지를 사용한다. 분해기법에는 직교분해기법(orthographic factorization), 스케일된 직교분해기법(scaled orthographic factorization), 근접 원근분해기법(paraperspective factorization)이 있다[3][4]. 직교분해기법에서는 카메라의 모션정보 추정에서 깊이값 계산이 불가능하고, 복원된 모습이 *x*-축으로 넓게 스케일된 결과가 나오는 단점이 있다. 근접 원근 분해기법은 가장 정확한 형상 추정결과를 보이나 노이즈에 민감하여 형상 추정이 불안정하며, 계산 과정이 어렵다는 단점이 있다. 이런 단점 때문에 근접원근분해기법의 적용이 실질적으로 곤란한 경우가 많다[5][6].

3. 개선된 직교분해기법을 이용한 복원

본 논문에서 제안하는 개선된 직교분해 기법은 직교분해 기법에 기반한 방법이나 스케일 인자를 고려하여 기준이 되는 참조위치에 대한 깊이정보도 추정하도록 한다. 선형시스템으로 구현할 수 있도록 하여 초기치가 필요 없으며 물체 추적이 실패하더라도 즉시 재시작될 수 있다. 각 프레임에 대해서 장면의 공간 정보 및 카메라 자세가 복원되며 그 처리시간은 직교분해 기법에 기반하므로 매우 빠르게 수행된다.

3.1 모양 복원

범용 디지털 카메라를 통해 얻은 F 개의 이미지에 대하여 N 개의 특징점이 주어져 있다고 가정한다. 이때 p 번째 이미지에서 f 번째 특징점을 식 (1)과 같이 나타내기로 한다.

$$\left\{(x_{fp}, y_{fp}) \mid f = 1,...,F, p = 1,...,N\right\} \quad (1)$$

특징점들로부터 개선된 분해기법의 입력으로 사용될 2F×N 인 **W** 행렬을 다음과 같이 구성한다.

$$\mathbf{W} = [\mathbf{X}^T \mid \mathbf{Y}^T]^T$$

X는 특징점들의 x좌표의 열벡터로 구성된 F×N 행렬이고, **Y**는 특징점들의 y 좌표로 열벡터로 구성된 F×N 행렬이다. 분해기법에 사용되기 위한 형식으로 **W'**=[**X'**|**Y'**]의 형태로 변형한다. **X'**과 **Y'**은 **X'**=[x'_{fp}], **Y'**=[y'_{fp}]으로 구성되고, x'_{fp}와 y'_{fp}는 식 (2)와 같이 계산된다.

$$x'_{fp} = x_{fp} - \frac{1}{N}\sum_{p=1}^{N} x_{fp} \quad , \quad y'_{fp} = y_{fp} - \frac{1}{N}\sum_{p=1}^{N} y_{fp} \quad (2)$$

변형된 **W'**은 Singular Value Decomposition(SVD)을 사용하여 **W'**=**UDV** 와 같이 분해한다[7]. **U** 의 첫 번째 세 개의 열로부터 **U'**을, **D** 의 첫 번째 세 개의 열중 처음 세 개의 행으로부터 3×3 인 **D'**을, **V** 의 첫 번째 세 개의 행으로부터 **V'**을 만든다. 이렇게 뽑아낸 **U**, **D**, **V** 의 변형으로 **W'**을 재구성한다. 이렇게 **W'**을 재구성할 수 있는 것은 **U**, **D**, **V** 의 앞쪽 세 개만 취하더라도 노이즈 레벨을 초과하는 모든 특이값(singular value)을 포함할 수 있기 때문이다.

R'과 **S'**을 각각 가상의 회전과 3 차원 형상 행렬이라고 가정하고 이를 식 (3)과 같이 정의하면

$$\mathbf{R}' = \mathbf{U}'[\mathbf{D}']^{1/2} \quad , \quad \mathbf{S}' = [\mathbf{D}']^{1/2}\mathbf{V}' \quad (3)$$

W'=**R'S'**과 같이 만들 수 있다. **R'**은 2F×3 의 크기를 갖는 실제 회전행렬 **R** 의 선형변환이고, **S'**은 3×P 의 크기를 갖는 실제 형상행렬 **S** 의 선형변환이다. **R'**과 **S'**로부터 식 (4)를 통해 **R** 과 **S** 를 구할 수 있다[1].

$$\begin{aligned} \mathbf{W}' &= \mathbf{R}'\mathbf{S}' = (\mathbf{R}'\mathbf{Q})(\mathbf{Q}^{-1}\mathbf{S}') \\ \mathbf{R} &= \mathbf{R}'\mathbf{Q} \quad , \quad \mathbf{S} = \mathbf{Q}^{-1}\mathbf{S}' \end{aligned} \quad (4)$$

3.2 카메라 모션 정보

카메라의 이동 경로와 방향을 표시함으로써 모션정보를 나타낼 수 있다. 이동 경로는 각 이미지 프레임의 특징점들의 평균값으로서 나타낼 수 있다.

$$c_x = \frac{1}{N}\sum_{p=1}^{N} x_{fp} \quad , \quad c_y = \frac{1}{N}\sum_{p=1}^{N} y_{fp} \quad (5)$$

식 (5)는 카메라 이동경로의 x, y 좌표를 나타낸다.

모양 복원 시 계산된 **R** 을 통해 카메라의 방향을 계산할 수 있다. R 은 식 (6)과 같은 형태이다.

$$\mathbf{R} = [\mathbf{i}_1 \cdots \mathbf{i}_N \mathbf{j}_1 \cdots \mathbf{j}_N] \quad (6)$$

여기서 $\mathbf{i}_1$과 $\mathbf{j}_1$, $\mathbf{i}_N$과 $\mathbf{j}_N$과 같이 대응되는 행 벡터를 서로 외적 함으로써 각 이미지 프레임에서의 카메라 방향을 계산할 수 있다.

스케일된 직교분해기법에서는 카메라 위치의 깊이 정보를 계산할 수 있다. 하지만 계산된 z-좌표값은 단위가 조절 (normalize)되어 있는 상태이므로 카메라 위치의 x-좌표와 y-좌표도 단위를 조절해야 한다. 또한 각 프레임에서 특징점의 평균으로 계산된 카메라의 이동 경로는 서로 방위(orientation)가 다르다. 방위를 맞춰야 올바른 카메라의 이동경로를 계산할 수 있다. 방위는 프레임별 이동 경로에 각각의 회전 행렬을 적용시킴으로

써 일치할 수 있다. 단위를 조절한 x, y 좌표와 계산된 z 좌표를 함께 프레임 별 회전행렬에 적용시킴으로써 올바른 카메라의 모션 정보를 계산할 수 있다.

4. 3 차원 정보의 융합

연속되는 이미지 시퀀스에서 복원할 수 있는 물체의 정도는 일부분에 해당한다. 좀더 완벽한 복원 모습을 위해서 부분 데이터의 융합이 필요하다 [8].

데이터 융합의 선행 작업으로서 3 차원 데이터 집합 $\mathbf{D}_i$와 $\mathbf{D}_j$가 복원되어 있음을 가정한다. $\mathbf{D}_i$와 $\mathbf{D}_j$는 서로 공통된 부분을 갖고 있어야 한다. 그림 1 은 3 차원 스캐너를 사용한 오브젝트의 복원된 부분데이터의 모습이다. 공통된 부분을 갖는 두 오브젝트의 융합 모습을 볼 수 있다.

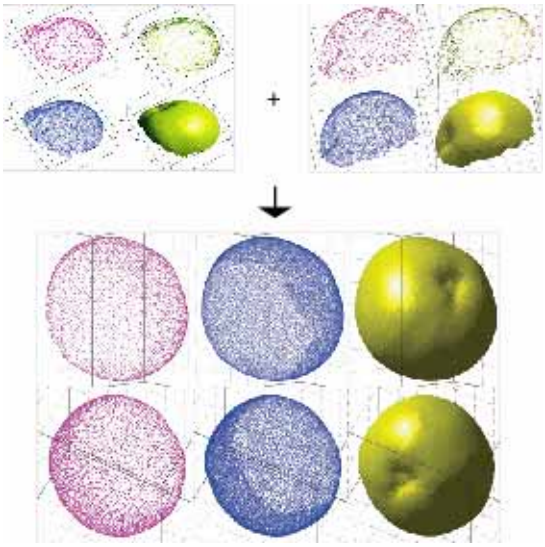

그림 1. 부분데이터의 융합

두 데이터 셋은 서로 다른 좌표계를 갖는 오브젝트로서 두 오브젝트의 융합을 위해서 좌표계의 변환 과정이 필요하다. 이때 변환 행렬을 **T** 라 하면 변환 형식은 식 (7)과 같다.

$$\mathbf{X}_i = \mathbf{T}_{ij}\mathbf{X}_j \tag{7}$$

여기서

$$\mathbf{X}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \mathbf{X}_j = \begin{bmatrix} x_j \\ y_j \\ z_j \\ 1 \end{bmatrix}, \mathbf{T}_{ij} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

이다. 식 (7)은 다음과 같이 나타낼 수 있고,

$$\begin{bmatrix} r_{11}x_j + r_{12}y_j + r_{13}z_j + t_1 - x_i \\ r_{21}x_j + r_{22}y_j + r_{23}z_j + t_2 - y_i \\ r_{31}x_j + r_{32}y_j + r_{33}z_j + t_3 - z_i \\ 0 \end{bmatrix} = \mathbf{0}_4 \tag{8}$$

이를 **Mq=0** 형태인 동차방정식으로 변형하면 **M** 은 3 차원 좌표로 구성되어 있는 알고 있는 정보이므로 SVD 를 사용하여 벡터 **q** 를 계산할 수 있다. 이와 같은 방법으로 식 (8)을 변형하면

$$\mathbf{Mq} = \mathbf{0}$$

$$\mathbf{M} = \left[\begin{array}{ccc|c|c} \mathbf{p}_j^T & \mathbf{0}_3^T & \mathbf{0}_3^T & & \\ \mathbf{0}_3^T & \mathbf{p}_j^T & \mathbf{0}_3^T & \mathbf{I}_{3\times3} & -\mathbf{p}_i \\ \mathbf{0}_3^T & \mathbf{0}_3^T & \mathbf{p}_j^T & & \end{array}\right] \tag{9}$$

식 (9)와 같고 $\mathbf{p}_i$, $\mathbf{p}_j$, **q**는 다음과 같다.

$$\mathbf{p}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T, \mathbf{p}_j = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^T$$
$$\mathbf{q} = \begin{bmatrix} r_{11}, ..., r_{33}, t_1, t_2, t_3, 1 \end{bmatrix}^T$$

13×1 의 벡터 **q** 는 12 개의 미지수를 포함하고 있다. **q** 를 구하기 위하여 공통점의 쌍이 5 쌍 이상 필요하다. 해를 구하기 위한 최소 점의 수인 5 쌍으로 표현된 동차방정식은 다음과 같다.

$$\mathbf{Aq} = \mathbf{0}, \mathbf{A} = \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_5 \end{bmatrix} \tag{10}$$

식 (10)에서 $\boldsymbol{M}_i$는 식 (9)와 같이 표현된다. 15×13 행렬 **A**의 값은 알고 있는 3 차원 좌표이므로 **Q**를 구하기에 충분하다. 이와 같은 방법으로 **T**행렬을 구성하여 식 (7)을 통해 서로 다른 부분 오브젝트를 하나의 오브젝트로 융합할 수 있다.

5. 부분 데이터의 복원 실험

5.1 마커를 사용한 복원

3 차원 부분 데이터를 복원하는 방법으로 개선된 직교분해기법을 사용하였다. 본 실험에서는 직각 모양으로 생긴 마커를 사용하여 모양 복원과 카메라 모션정보 복원 실험을 하였다. 실험에는 32 개의 이미지 시퀀스가 사용되었고, 각 이미지 프레임에서 12 개의 특징점을 모양 복원에 사용하였다. 그림 2 는 복원에 사용된 마커의 이미지 프레임이다.

그림 2. 마커 이미지 프레임

그림 3 에서 복원된 마커의 모양과 카메라의 모션정보를 볼 수 있다. 12 개의 특징점이 3 차원적으

로 복원되었다. 32 개의 카메라의 위치를 표시하고, 화살표 방향으로 카메라의 방향을 표시하였다.

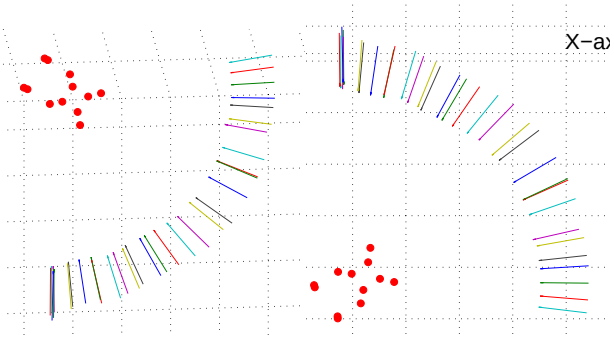

그림 3. 복원된 마커의 모양과 카메라 모션 정보

마커와 같이 구조적 모양을 알고 있는 오브젝트는 3 차원 좌표를 추정할 수 있다. 정의된 3 차원 좌표와 비교한 복원된 좌표의 에러는 표 1 과 같다. 마커 안과 밖의 사각형, 세워져 있는 4 개의 점으로 이루어진 사각형을 구성하는 12 개의 선분의 길이를 비교하였다. 단위는 cm 를 사용하였고, 복원된 좌표값의 단위가 다르므로 단위조절 후 비교하였다. 표 1 에서 d1 에서 d4 는 사각형을 구성하는 선분길이를 의미한다. 표에서 알 수 있듯이 최대 0.3cm 의 오차가 발생하는 결과를 얻을 수 있었다.

표 1. 정의된 3 차원 좌표를 기준으로 한 복원된 3 차원좌표의 에러(단위:cm)

	d1	d2	d3	d4
inner square	0.2225	0.1199	-0.3137	-0.0287
outer square	0.0212	0.0270	-0.0735	0.0253
points	-0.1167	0.0846	-0.0274	0.0596

5.2 패턴 상자를 사용한 복원

표면에 패턴을 붙인 상자를 사용하여 융합을 위한 3 차원 부분 데이터를 복원하였다. 박스의 앞면, 윗면, 양 옆면을 포함한 형상의 복원을 위해 3 개의 부분 데이터를 복원하였다. 복원에 사용된 이미지 시퀀스는 부분 데이터 1 에 10 개, 부분 데이터 2 에 7 개, 부분 데이터 3 에 14 개이다. 부분 데이터를 복원하는데 사용한 특징점의 개수는 각각 61 개, 61 개, 43 개 이다. 그림 4 는 복원에 사용된 패턴 상자의 이미지 프레임이다.

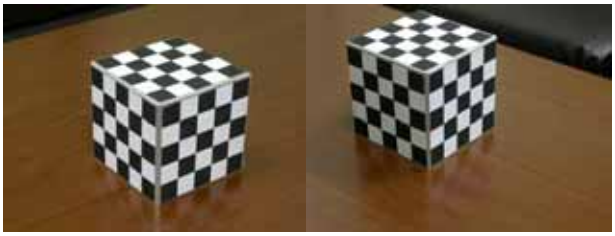

그림 4. 패턴 상자의 이미지 프레임

그림 5 에서 부분 데이터 1 과 부분 데이터 2 의 복원된 형상의 모습과 추정된 카메라의 모션을 볼 수 있다.

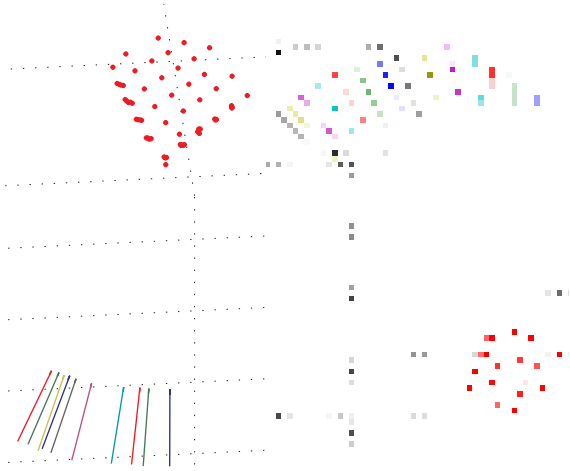

그림 5. 패턴 상자의 두 부분 데이터의 복원된 모양과 카메라 모션 정보

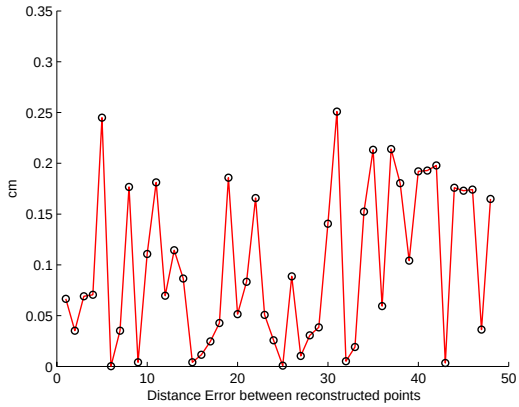

그림 6. 부분 데이터 1 의 복원된 3 차원 점들 사이의 실측과의 거리 오차

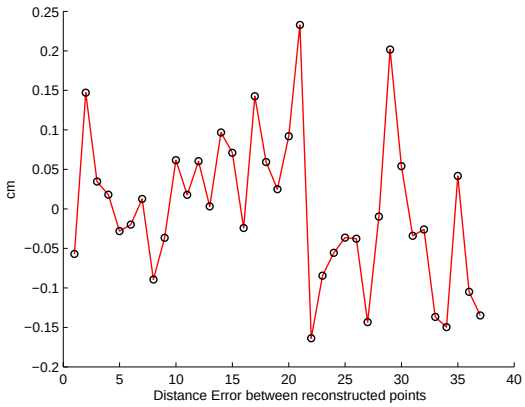

그림 7. 부분 데이터 2 의 복원된 3 차원 점들 사이의 실측과의 거리 오차

그림 6 은 부분 데이터 1 의 복원된 두 점 사이의 길이를 실측과 비교하여 오차를 측정한 것이다. 48 개의 길이에 대한 오차는 최대 0.25cm 정도이고, 절대값을 취한 오차의 평균은 0.0986cm 로 상당히

작은 오차를 갖는 결과가 나오는 것을 알 수 있다.

그림 7 은 그림 6 에서의 실험과 같은 방법으로 부분 데이터 2 에 대하여 오차를 측정한 결과이다. 이 실험에서는 37 개의 길이를 측정하여 실측과 비교하였으며 오차 평균값은 0.0742cm 로서 그림 6 의 실험보다도 정확한 결과가 나오는 것을 알 수 있다.

5.3 양(sheep) 인형을 사용한 복원 실험

특정한 패턴을 사용하지 않은 불규칙한 모형을 갖는 대상에 대한 실험을 해보았다. 복원 대상으로 인형을 사용하였다. 융합을 위한 부분데이터를 생성하였으며, 패턴 상자와 마찬가지로 3 개의 부분 데이터를 복원하였다. 생성된 부분 데이터는 인형의 앞쪽, 왼쪽, 오른쪽에 해당한다. 그림 8 은 실험에 사용된 이미지 시퀀스이며, 특징점을 추적하는 모습을 보여주고 있다.

그림 8. 인형의 이미지(부분 데이터 3) 시퀀스 및 특징점 추적

그림 9 에서 각 부분 데이터의 복원된 모습과 카메라의 모션 정보를 확인할 수 있다.

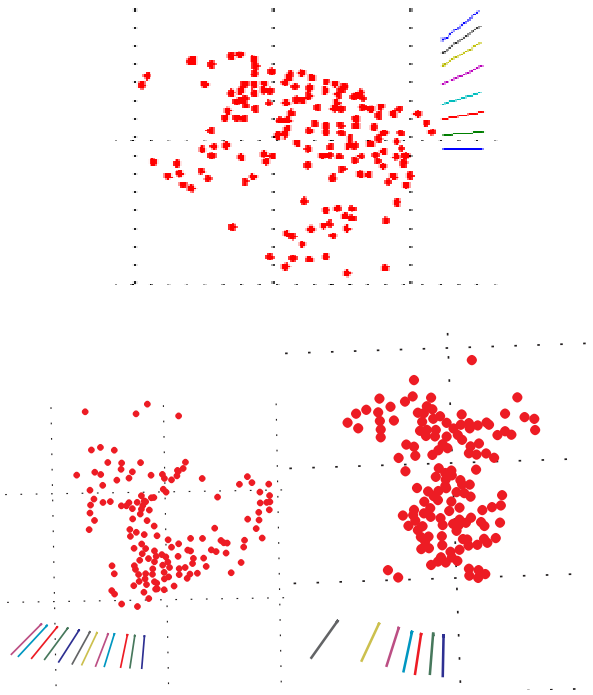

그림 9. 인형의 왼쪽(아래 왼쪽), 오른쪽(아래 오른쪽), 앞면에(위) 대한 복원된 모습과 추정된 카메라의 모션정보

이미지 시퀀스는 부분 데이터 별로 7 개, 8 개, 12 개를 사용하였으며, 특징점의 개수는 정면 143 개, 왼쪽 155 개, 오른쪽 146 개를 사용하였다.

5.4 복원 처리 시간

부분 데이터의 복원 처리 시간은 표 2 와 같다.

표 2. 부분 데이터의 복원 처리 시간 (단위:초)

인형	0.5000	0.5000	0.5150	0.5150
패턴상자	0.7820	0.8280	0.7500	0.9380

복원 처리 시간은 대략 0.5 초 정도의 시간이 걸리는 결과를 볼 수 있다. 이와 같은 빠른 처리 시간으로 실 시간적 복원에 대한 적용이 가능할 것으로 보인다.

6. 복원된 부분 데이터의 융합 실험

6.1 패턴 상자 데이터의 융합

패턴상자의 정면, 위, 양 옆의 4 면을 복원하는데 3 개의 복원된 부분 데이터를 사용하였다. 이전 실험과 마찬가지로 부분 데이터의 복원이 선행되었다. 3 개의 부분 데이터 중 앞면을 가장 잘 나타내는 데이터를 기준으로 나머지 두 개의 부분 데이터의 좌표계를 변환하여 융합을 하였다.

복원 데이터 융합 시 카메라의 이동경로와 방향도 좌표계를 기준이 되는 부분데이터에 일치함으로써 올바른 카메라 모션정보를 표현할 수 있다. 카메라의 이동경로와 방향을 점 좌표로 생각하고 형상 변환과 동일한 변환행렬을 모션정보에 적용하였다. 그림 10 에서 패턴상자의 융합된 결과를 볼 수 있다. 상자 표면에 바둑판모양 패턴을 붙여 실험한 결과 특징점 추적의 효율성 때문에 비교적 정확한 형상 복원이 가능하였다.

그림 11 은 패턴 상자의 부분 데이터 두 개를 융합할 때 지정한 공통점들의 변환 오차이다. 두 부분 데이터 사이의 변환 행렬을 구할 때 사용한 10 개의 공통점에 대한 변환 오차를 계산하였다. 오차 계산시 복원된 좌표의 값을 실 물체에 대하여 스케일을 맞추었다. x-좌표와 y-좌표의 경우는 오차가 0.1cm 을 넘지 않는 결과를 보였으나 z 좌표의 경우 최대 0.5cm 에 가까운 오차가 발생하였다.

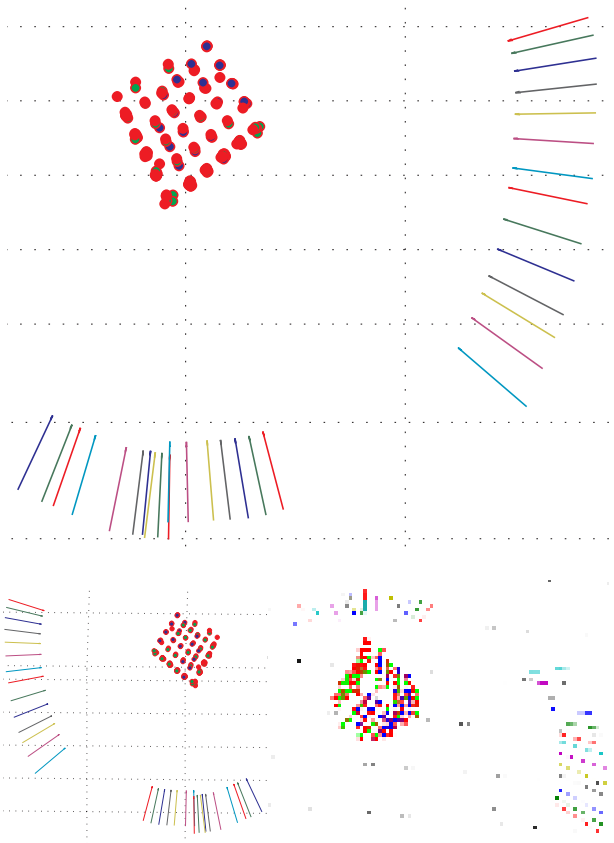

그림 10. 패턴 상자 데이터의 융합

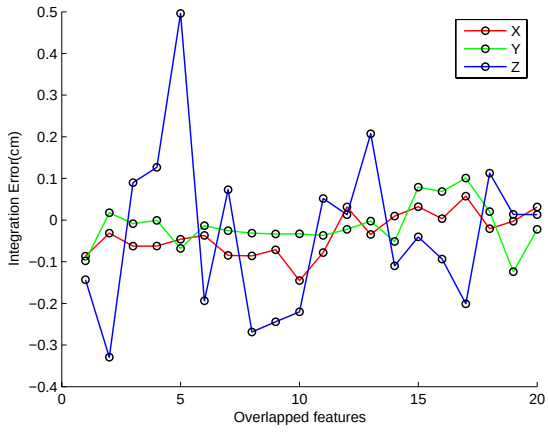

그림 11. 패턴 상자의 두 부분 데이터의 융합오차

6.2 인형의 융합 실험

인형의 정면과 양 옆면의 복원을 위하여 정면, 왼쪽, 오른쪽의 부분 데이터를 융합하였다. 실험내용은 패턴 상자와 같다. 그림 12 에서 연속되지 않는 세 방향에서 촬영한 영상을 이용해서 복원된 3 차원 구조를 확인할 수 있다.

그림 13 은 연속되는 네 개의 부분 데이터를 융합한 3 차원 구조를 나타내고 있다. 인형의 오른쪽 모습부터 정면까지의 모습이다.

그림 14 에서 그림 13 실험에서의 융합 에러를 나타내었다. 네 개의 부분 데이터가 가지는 세 부분의 공통 영역에 존재하는 공통점에 대한 융합 에러로서 최대 0.5cm 정도의 오차를 가짐을 알 수 있다.

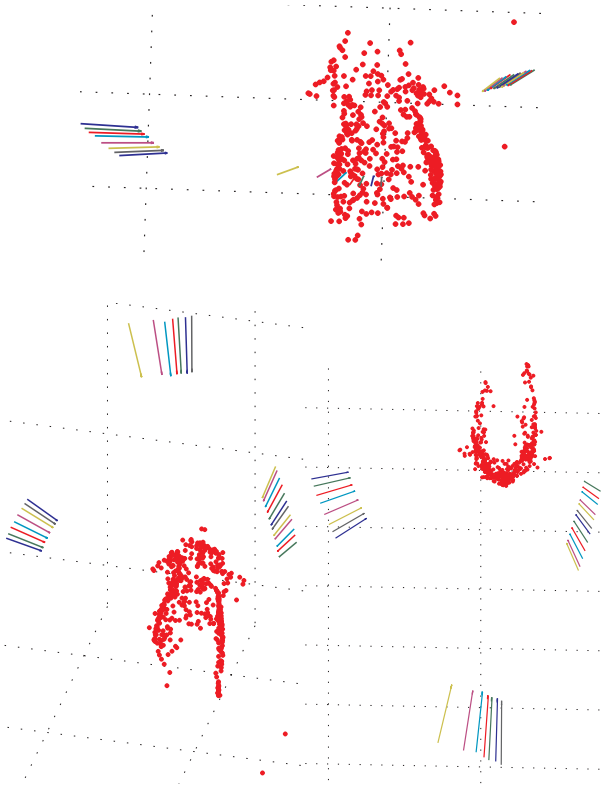

그림 12. 불연속적인 인형 부분 데이터의 융합

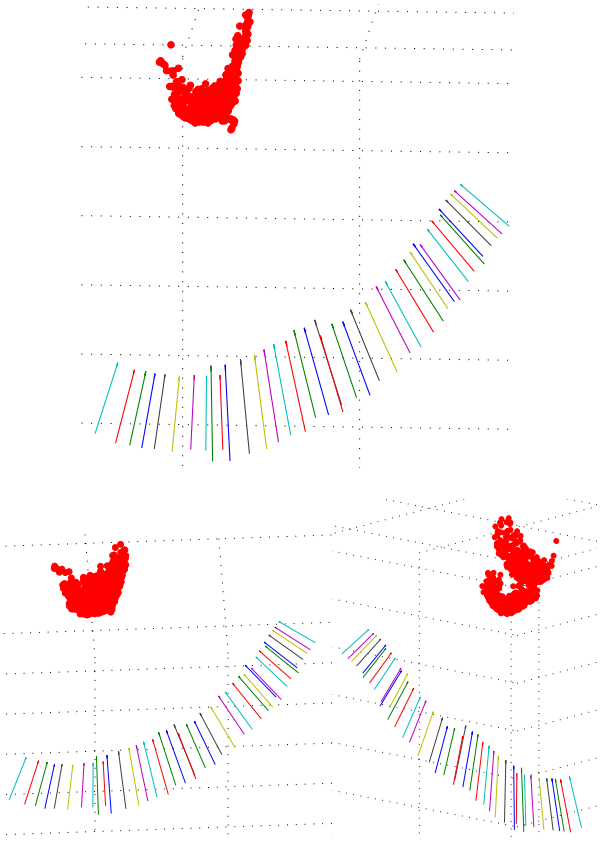

그림 13. 연속되는 인형 부분 데이터의 융합

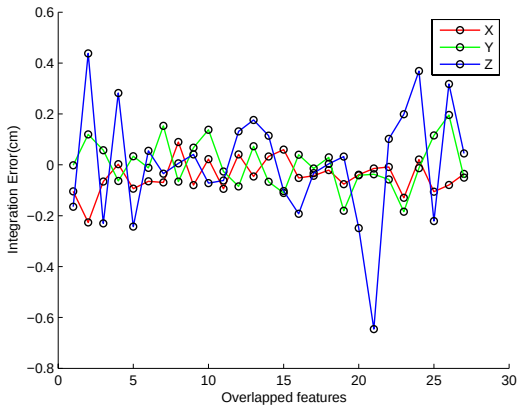

그림 14. 인형의 두 부분 데이터의 융합 오차

7. 결론

본 연구에서는 개선된 직교분해기법을 사용하여 대상의 3 차원 데이터를 부분적으로 획득하고 부분 데이터의 융합을 통해 복원의 완성도를 높이는 방법을 제안하였다. 영상을 사용한 특징점 추출 및 추적을 통해 2D 데이터를 획득하고 이를 통해 3 차원 구조를 복원하였다. 융합 시 부분 데이터들이 일치 하는 영역에서 5 개 이상의 공통점을 지정함으로써 변환행렬을 계산하였다. 변환행렬을 적용하여 모든 데이터를 기준이 되는 데이터의 좌표계로 이동함으로써 융합된 복원 데이터를 획득할 수 있었다.

실험에서 사용한 패턴 상자의 경우 특징점 추적 및 공통점 지정이 용이하므로 비교적 정확한 복원을 할 수 있었고 융합된 모습도 실 대상과 비슷한 결과를 얻을 수 있었다. 추정된 카메라의 모션 정보 또한 실제 측정한 카메라의 위치, 방향과 유사하게 나타나는 것을 볼 수 있었다. 인형을 사용한 실험에서는 패턴 상자보다 에러가 조금 크지만 형상 복원과 카메라 추정 및 융합에서 실물과 유사한 결과를 얻을 수 있었다.

융합 시 공통점을 지정하는 과정에서 많은 수의 공통점을 지정한 경우 지정점에 대한 오차는 조금 커졌으나 전체적인 점에 대한 오차가 줄어드는 것을 알 수 있었다.

본 연구에서는 거의 전체적인 융합을 완성하긴 했으나 복원과 융합 에러에 대한 개선의 여지가 있다. 에러를 줄이기 위해 특징점 추출 및 추적에 대한 추가적인 연구가 우선적으로 필요하다. 좀더 정확한 복원이 이루어졌을 경우 융합한 데이터에 텍스쳐를 입혀 사실적 묘사를 하는 것도 향후 연구 과제이다.

Reference

[1] Christoph Bregler, Aaron Hertzmann, Henning Biermann, “Recovering Non-Rigid 3D Shape from Image Streams,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, pp. 190-696, June 2000.

[2] Mei Han, Takeo Kanade. “Creating 3D Models with Uncalibrated Cameras,” WACV, p. 178, Fifth IEEE Workshop on Applications of Computer Vision, 2000.

[3] Carlo Tomasi, Takeo Kanade, “Shape and Motion from Image Streams: a Factorzation Method,” Technical Report CMU-CS-91-172, Carnegie Mellon University, Pittsburgh, PA, September 1991.

[4] Conrad J, Poelman and Takeo Kanade, “A Paraperspective Factorization Method for Shape and Motion Recovery,” Technical Report CMU-CS-93-219, Pittsburgh PA, December 1993.

[5] L.Matthies, T.Kanade, R.Szeliski, “Kalamn filter-based algorithms for estimating depth from image sequences,” International Journal of Computer Vision, Vol. 3(3), pp. 209-236, September 1989.

[6] C. Tomasi, T. Kanade, “Shape and Motion without depth,” In Proceedings of the DARAPA Image Understanding Workshop, pp. 258-270, Pittsburgh, September 1990.

[7] G.H. Golub, C.Reinisch, “Singular Value Decomposition and Least Squares Solutions,” In Handbook for Automatic Computation, Vol. 2, pp. 134-151, 1971.

[8] Andrew Johnson, Sing Bing Kang, “Registration and Integration of Textures 3-D Data,” Proceedings of 3DIM’97, pp. 234-241, 1997.