

실사 객체 복원 및 변형 프레임워크

*김태연, **박정식, *박종일¹

*한양대학교 컴퓨터공학부, **한양대학교 컴퓨터 · 소프트웨어 학과
teayounty@hanyang.ac.kr, nangsik@mr.hanyang.ac.kr, jipark@hanyang.ac.kr

A Framework for Reconstruction and Manipulation of Real Objects

*Taeyeon Kim, **Jungsik Park, and *Jong-Il Park¹

*Division of Computer Science and Engineering, Hanyang University

**Department of Computer and Software, Hanyang University

요 약

본 논문에서는 3 차원 실사 객체를 사용자의 상호작용을 통해 변형시키는 프레임워크를 제안한다. RGB-Depth 카메라로 다각도에서 객체를 촬영하여 3 차원 좌표 및 색상정보를 획득하고 각 영상에서의 3 차원 좌표들을 이용하여 카메라 포즈를 계산한다. 계산한 카메라 포즈와 획득한 3 차원 좌표 및 색상정보를 이용하여 객체의 3 차원 정보를 복원한 후 복원된 객체에 대해 메쉬(Mesh)를 생성한다. 이렇게 실사 객체의 3 차원 정보를 메쉬로 표현한 뒤, 사용자의 상호작용을 통해 객체의 변형을 가하게 되면 메쉬를 변형하여 렌더링 함으로써 사용자가 원하는 모습으로 실사 객체를 변형시킬 수 있다.

1. 서론

3 차원 정보를 이용하는 증강현실과 가상현실 분야는 최근 들어 오컬러스와 같은 HMD(Head-Mounted Display) 장비들이 개발되면서 관심이 급증하고 있고 이에 따른 관련 연구도 활발하게 진행되고 있다. 실세계에서 가상의 물체를 보여주는 증강현실은 사용자에게 시각적 경험과 상호작용을 통해 다양한 경험을 할 수 있게 한다. 예를 들면, [1]과 같이 시각적 경험을 제공할 수도 있고, [2]과 같이 실제 환경에 기반한 음향효과를 제공할 수도 있다.

본 논문에서는 가상의 콘텐츠가 아닌 실사 객체를 사용자의 상호작용을 통해 변형하여 보여주는 프레임워크를 제안한다. 우선 실사 객체의 변형을 위해 실사 객체의 3 차원 정보를 획득하는 작업을 수행한다. 3 차원 정보 획득은 Kinect 와 같은 RGB-Depth 카메라를 이용하거나[3] 일반 웹캠 또는 핸드폰 카메라[4], 3 차원 스캐너[5] 등을 이용한 다양한 기법들이 있으며 본 논문에서는 [3]과 같이 RGB-Depth 카메라를 사용하였다. 이후, RGB-Depth 카메라로부터 획득한 3 차원 정보를 이용하여 실사 객체에 대한 3 차원 포인트 클라우드를 복원한다.

포인트 클라우드 복원은 같은 지점끼리 대응점을 설정하여 객체를 추적하면서 수행되는데, 이때 잘못된 매칭이나 카메라 포즈 오류를 억제하기 위한 여러 방법이 있다. 이러한 방법으로서 주로 여러 이미지에서 보이는 객체에 대한 오류를 계산하는 bundle adjustment 를 통한 기법[4], 반복을 통하여 3 차원상의 모양을 일치시켜 나가는 ICP (Iterative Closest Point) 알고리즘[6]이 많이 사용된다. 본 논문에선 정확도와 속도를 위해 ICP 기법을 이용한다. 마지막으로 복원한 포인트 클라우드로부터 메쉬(mesh)를 생성하고 사용자의 상호작용을

통해 메쉬 단위로 변형을 하면 효과적으로 사용자가 원하는 모습으로 실사객체를 변형시킬 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 실사 객체 복원 및 변형 방법에 대해 설명하고, 3 장에서는 실험 결과를 보이며, 4 장에서는 결론을 맺는다.

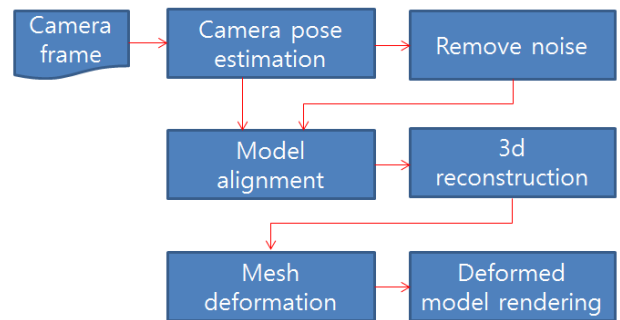


그림 1. 프레임워크 흐름도

2. 실사 객체 복원 및 변형 방법

3 차원으로 복원된 객체는 임의의 방향에서도 객체의 모습을 볼 수 있다. 실세계의 객체에 대하여 여러 각도에서 촬영한 영상을 이용하여 3 차원 복원을 진행하는 경우 한 영상에서 보이는 객체는 사방에서 보이는 면이 아닌 한 곳에서 보이는 면만 보인다. 여러 방향에서 획득한 정보를 하나로 합치면 여러 방향에서 본 객체의 모습을 올바르게 복원할 수 있다. 이때 각 영상에서의 상대적 카메라 포즈를 알아야 두

¹ 교신저자

영상의 3 차원 데이터를 합칠 수 있기 때문에 3 차원 복원을 위해서 카메라 포즈는 반드시 계산되어야 한다.

카메라 포즈를 계산하는 방법은 대표적으로 특징점 매칭과 ICP 방법이 있다. 특징점 매칭 방법은 처리속도가 빠르나 정확도가 떨어지는 문제가 있고, ICP 방법은 속도는 느리지만 정확도가 높다. 따라서 본 논문에서는 이 두 방법의 장점을 이용하여 특징점 매칭 방법으로 초기 카메라 포즈를 계산하고 ICP 알고리즘을 적용하여 카메라 포즈의 정확도를 높인다. 카메라 포즈 계산을 위해 두 영상에 대한 3 차원 정보가 필요하며, 이는 RGB-Depth 카메라로부터 얻은 데이터와 카메라 내부 파라미터를 통해 계산할 수 있다. 그림 1 은 제안한 프레임워크의 흐름도를 나타낸다.

2.1. 특징점 매칭을 통한 카메라 포즈 계산

특징점 매칭을 수행하기 위해 기준 영상(Reference image)에서 해리스 코너(Harris corner)를 찾은 후 이를 옵티컬 플로우(optical flow)로 목표 영상(Target image)에서의 매칭점을 찾는다. 이렇게 찾은 매칭쌍을 3 차원 좌표로 변환하여 카메라 포즈를 계산한다. 영상 I_1 의 특징점들을 3 차원 좌표로 나타낸 집합을 f_1 이라 하고, 영상 I_2 의 특징점들을 3 차원 좌표로 나타낸 집합을 f_2 라 할 때, f_1 과 f_2 는 3 차원 상에서 회전변환과 이동변환의 관계를 갖는다. 회전변환의 관계만 남기기 위해, f_1 과 f_2 의 중심 c_1 과 c_2 를 계산하여 c_1, c_2 를 원점으로 이동시키는 이동변환을 f_1 과 f_2 에 적용한다. 이때의 이동변환을 각각 t_1, t_2 라 하자.

$$t_1 = -c_1 \quad (1)$$

$$t_2 = -c_2 \quad (2)$$

t_1 과 t_2 가 적용된 후의 점들의 집합을 각각 p_1, p_2 라 하면 p_1 과 p_2 사이에는 회전변환의 관계만 남게 된다.

$$p_1 = f_1 - c_1 \quad (3)$$

$$p_2 = f_2 - c_2 \quad (4)$$

따라서 p_1 과 p_2 사이의 회전변환은 SVD 를 이용하여 구할 수 있다. 이때의 회전변환을 R , p_1 과 p_2 의 크기를 N 이라 하면, 다음과 같이 계산할 수 있다.

$$\hat{R} = \underset{R}{\operatorname{argmin}} \sum_{i=0}^N \|p_{2i} - R * p_{1i}\|^2 \quad (5)$$

t_1 과 t_2 를 적용하기 전의 f_1 과 f_2 사이의 이동 변환 T 는 R 을 이용하여 계산할 수 있다.

$$T = R * t_1 - t_2 \quad (6)$$

계산한 R 과 T 를 이용하여 I_1 의 좌표계를 I_2 의 좌표계로 변환할 수 있다. 그러나 영상에서 특징점을 찾을 때의 오류와 깊이(depth) 데이터의 잡음으로 인해 정확한 카메라 포즈를 계산하기 어렵다. 이런 문제를 해결하기 위해 ICP 알고리즘으로 카메라 포즈의 오류를 보정한다.

2.2. ICP 를 통한 카메라 포즈 계산

ICP 알고리즘이란 기준 포인트 클라우드의 각 점과 가장 가까운 목표 포인트 클라우드의 점을 대응점으로 설정하여 오류를 최소화하는 회전변환과 이동변환을 계산하고, 계산된 변환을 적용하여 변환된 포인트 클라우드에 대하여 같은 과정을 반복하며 기준 포인트 클라우드를 목표 포인트 클라우드에 3 차원 상으로 최대한 맞게 이동시키는 알고리즘이다. ICP 는 반복을 통해 계산되므로, 많은 시간이 걸리지만 두 포인트 클라우드가 충분히 가깝다면 빠르게 수렴한다. 따라서 특징점 매칭을 통해 계산된 초기 포즈를 사용하여 빠르게 카메라 포즈의 정확도를 향상시킬 수 있다.

특징점 매칭과 ICP 를 수행하였어도 잡음의 영향으로 카메라 포즈가 부정확할 수 있다. 이를 보정하기 위해 카메라 포즈를 계산한 후 잡음을 제거하고 최종 카메라 포즈를 계산한다. 잡음을 제거한 뒤에 한번 더 카메라 포즈를 계산하는 것이므로 더 정확한 카메라 포즈를 결과로 얻을 수 있다.

그림 2 는 각 영상간의 포인트 클라우드 매칭 결과로 기준 영상의 포인트 클라우드(흰색)가 목표 영상의 포인트 클라우드(빨간색)에 3 차원 상으로 잘 맞게 이동된 것(초록색)을 확인할 수 있다.

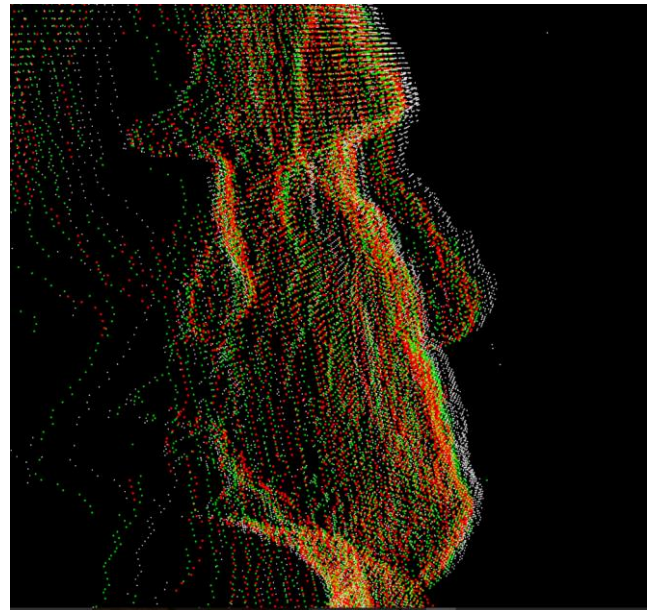


그림 2. 흰색(기준 포인트 클라우드), 빨간색(목표 포인트 클라우드), 초록색(카메라 포즈 계산 후 목표 포인트 클라우드로 이동시킨 기준 클라우드)의 세 포인트 클라우드로 나타난 ICP 를 통한 카메라 포즈 계산 결과.

2.3 잡음제거

RGB-Depth 카메라로 획득한 깊이 정보에는 매우 많은 잡음이 존재한다. 이를 처리하지 않고 그대로 3 차원 복원을 수행하면 그림 3 의 (a)와 같이 원래 객체를 제대로 알아볼 수 없을 만큼 지저분한 결과가 발생한다. 따라서 이러한 잡음을 제거하여 3 차원 복원을 깔끔하게 하기 위해 깊이 정보의 잡음을 제거해야 한다.

잡음 제거는 두 단계로 수행된다. 첫 번째 단계에서는 두 포인트 클라우드 간의 가장 가까운 점의 거리를 기준으로 수행된다. 계산된 카메라 포즈를 포인트 클라우드에 적용하여 두 포인트 클라우드를 겹쳐놓은 상황에서, 한 포인트 클라우드의 각 점에서 가장 가까운 다른 포인트 클라우드의 점과의 거리가 문턱값(threshold) 이상 떨어져 있다면 이 점을 잡음이라고 판단한다. 계산된 카메라 포즈가 정확하다면 두 포인트 클라우드는 일치할 것이기 때문에, 만약 두 포인트 클라우드의 점들이 문턱값 이상 떨어져있다면 잡음이라고 판단하기에 충분하다. 두 번째 단계에서는, 기준 포인트 클라우드의 한 점의 깊이값이 그 점을 목표 영상 평면에 투영시킨 좌표의 깊이값보다 작다면, 이는 곧 목표 영상에서는 보이지 않는 점이라고 판단하여 제거한다. 이러한 두 단계의 잡음 제거를 통해 그림 3 과 같이 상당수의 잡음을 제거할 수 있다.

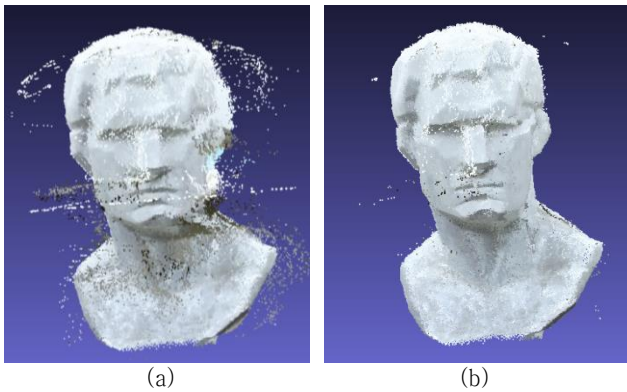


그림 3. (a)잡음 제거 전. (b) 잡음 제거 후.

2.4. 객체 변형

객체를 3 차원으로 복원 한 후 [7]의 방법을 이용하여 객체를 변형한다. 객체 변형을 위하여 포인트 클라우드를 메쉬화한 후 라플라시안 기반 메쉬 처리[8]로 객체의 변형을 수행한다. 포인트 클라우드에서 메쉬 단위로 변환하는 과정에서 추가적으로 잡음과 구멍(hole)을 제거할 수 있다. 그림 3 의 (b)와 그림 4 의 (b)를 비교하면 이를 확인할 수 있다.

객체 변형은 만들어진 객체의 3 차원 모델에 사용자가 임의로 변형을 가한 뒤, RGB 카메라 영상에서 변형하고자 하는 객체를 추적하면서 변형한 객체를 그 위에 나타내주는 형태로 진행된다.

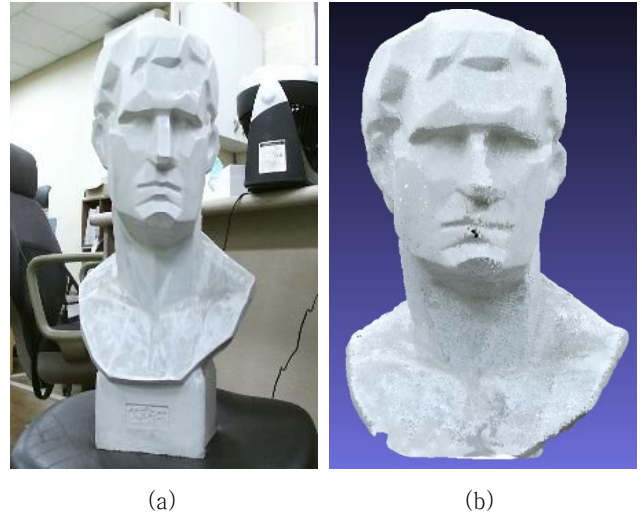


그림 4. (a) 대상 객체의 영상. (b) 객체를 3 차원 복원한 뒤 메쉬 단위로 처리한 결과.

3. 실험 결과

본 논문에서는 영상을 획득하기 위해 RGB-Depth 카메라로 Kinect v2 를 사용하였으며, 카메라 내부 파라미터는 사전에 카메라 캘리브레이션을 통해 획득하였다.

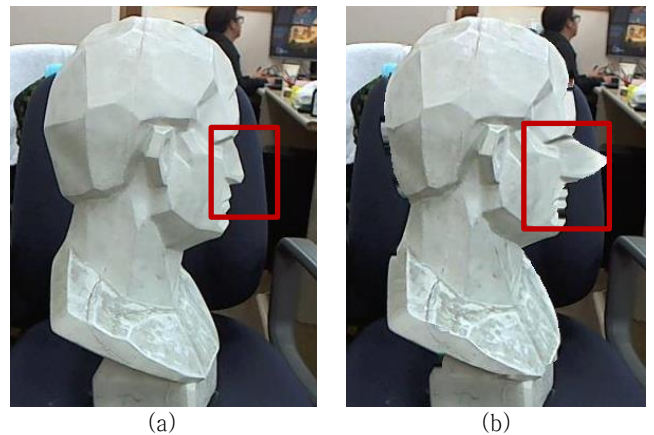


그림 5. (a) 카메라 영상의 객체의 모습. (b) 변형한 객체를 보여주는 모습.

2 장에서 설명한 두 이미지 사이의 카메라 포즈 계산과 하나로 합친 포인트 클라우드의 잡음을 제거하는 방법을 통해 복원해낸 3 차원 모델은 그림 3 의 (b)에서 확인할 수 있다. 그림 4 의 (a)는 객체를 촬영한 영상 중 하나이고, (b)는 객체를 3 차원 복원한 뒤 메쉬로 처리한 모습이다. 전체적으로 복원이 잘 수행된 것을 확인할 수 있으나, 잡음 제거 과정에서 일부 제거되지 않은 잡음으로 인해 입술 중앙 아래 부분에 구멍이 생긴 것을 볼 수 있다. 3 차원 모델이 완성된 후 객체 변형을 수행하였으며, 그 결과 그림 5 와 같이, 객체 변형이 잘 수행되는 것을 확인할 수 있다.

4. 결론

증강현실이나 가상현실 기술을 구현하기 위해선 3 차원

모델이 필요하며, 특히 실사 객체 변형 기술에는 대상 객체에 대한 정확한 모델을 얻기 위해 비전 기반 3 차원 복원이 수반된다. 본 논문에서는 실사 객체에 대한 3 차원 복원 및 이를 이용한 실사 객체 변형 프레임워크를 제안하였다. 실세계의 객체를 여러 방향에서 촬영한 각 영상에 대한 카메라 포즈를 계산하고, 계산된 카메라 포즈를 이용하여 각 영상의 포인트 클라우드를 병합한 뒤 잡음을 제거하는 과정을 반복하여 하나의 포인트 클라우드로 합성한다. 통합된 포인트 클라우드는 메쉬화되어 이후 사용자의 상호작용을 통해 메쉬 단위로 변형된 모습을 영상에 겹쳐 렌더링함으로써 실사 객체가 변형된 것처럼 보여준다. 제안된 방법을 통해 3 차원 복원 및 실사 객체 변형이 잘 수행되는 것을 확인할 수 있었지만, 복원된 3 차원 포인트 클라우드의 잡음이 일부 남아있는 문제가 있었다. 이러한 문제점을 해결하기 위한 연구를 진행할 예정이다.

감사의 글

이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.B101-16-0295, 초고품질 콘텐츠 지원 UHD 실감방송/디지털시네마/사이니지 융합서비스 기술 개발)

참고문헌

- [1] A. Fedosov, I. Elhart, E. Niforatos, A. North, and M. Langheinrich., "SkiAR: Wearable Augmented Reality System for Sharing Personalized Content on Ski Resort Maps," Proceedings of the 7th Augmented Human International Conference 2016.
- [2] T. Chatzidimitris, D. Gavalas, and D. Michael., "SoundPacman: Audio Augmented Reality in Location-based Games," Electrotechnical Conference (MELECON), 2016.
- [3] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, A. Fitzgibbon., "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, 2011.
- [4] P. Ondruška, P. Kohli, and S. Izadi, "MobileFusion: Real-time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones," IEEE Transactions on Visualization and Computer Graphics, Vol. 21, no. 11, pp. 1251-1258, 2015.
- [5] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," Proceedings of the 23rd Annual ACM Conference on Computer Graphics and Interactive Techniques, 1996.

[6] Dirk Holz, Alexandru E. Ichim, Federico Tombari, Radu B. Rusu, Sven Behnke., "Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D," IEEE Robotics and Automation Magazine Vol. 22, no. 4, pp. 110-124, 2015.

[7] J. Park, B.-K. Seo, and J.-I. Park, "Interactive Deformation of Real Objects" Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR), 2014.

[8] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," Proceedings of Eurographics Symposium on Geometry Processing, pp. 175-184, 2004.