

재구성된 포인트 클라우드 모델 기반 이미지 편집 시스템 개발

윤현욱*, 홍광진**
*숭실대학교 컴퓨터학부
**숭실대학교 글로벌미디어학부
e-mail : hongmsz@gmail.com

Development of Image Manipulation System based on Reconstructed Point-cloud Model

Hyun-Wook Yoon*, Kwang-Jin Hong**
*Dept. of Global Media, Soongsil University
**Dept. of Computer Engineering, Soongsil University

요 약

현재 사용되고 있는 보편적인 이미지 편집 방식은 이미지 내부 일부 영역을 선택 및 추출하는 방식으로 객체를 배경과 분리한다. 객체가 분리되는 과정에서 객체가 있었던 곳에서는 빈 영역이 발생하게 되는데, 이 문제를 해결하기 위해 인접한 영역을 가져와서 채우거나, 딥러닝을 적용하여 유사한 이미지로 채우는 방식이 가장 보편적이다. 그러나 이러한 방식은 배경에서 유실된 부분을 인공적인 방법으로 채우기 때문에 완벽하게 복원하기가 힘들다. 따라서 본 논문에서는 미리 해당 이미지에 대한 3 차원 정보를 가공 및 저장함으로써 편집으로 인해 유실되는 부분을 3 차원 정보로부터 복구할 수 있는 아이디어를 제안한다.

1. 서론

오늘날 일반적인 디지털 이미지 편집 과정은 수정하고 싶은 객체를 배경과 분리하여 선택한 뒤, 해당 객체에 대해서 위치를 변경하거나 형태를 변형하는 등의 수정을 통해 이루어진다. 이 때 객체와 배경 사이의 분리 작업은 일반적으로 객체와 배경과의 화소 값 비교를 통하여 객체만 선택될 수 있도록 하는 처리과정을 필요로 한다. 그런데 이러한 편집 방식은 객체를 배경과 분리하게 될 경우, 원래 객체가 있었던 공간이 소실되면서 정보의 유실이 발생한다는 문제점이 존재한다.

이러한 정보의 유실 문제를 해결하기 위해 여러가지 방법이 제안되었다. Adobe 의 Photoshop 프로그램에서 사용되고있는 내용 인식 채우기(content-aware fill) [1-3]는 patchmatch 알고리즘 기술을 기반으로 하여 이미지 편집 시 생기는 빈 공간을 이웃한 영역에서 채워 넣는 방법을 사용한다. 빈 공간을 인접한 부분들로 채워 넣기 때문에 작업시간이 매우 빠르지만 편집된 결과물이 합성 이미지처럼 보일 가능성이 높고, 상황에 따라서 굉장히 비정상적인 결과물이 출력될 수 있다.

그림 1 은 포토샵에서 이미지 처리 중 하나로 이미지 내 오브젝트 중 하나에 대한 위치를 조정하고 내용 인식 채우기를 통하여 수정할 경우, 인접한 영

역으로 채우게 되는데, 이동 전과 후 원래 자리의 빈 영역을 비교해보면 매우 어색해보이는 것을 확인할 수 있다.



(그림 1) 내용 인식 채우기 사용

다른 해결방법으로는 James Hays 와 Alexei A. Efros 의 Scene Completion Using Millions of Photographs 방법[4]으로, 디지털 이미지의 일부를 삭제할 경우 생기는 빈 영역에 대해서 유사한 이미지들로 이루어진 데이터를 학습하여 현재 이미지와 가장 일치할 것 같은 이미지의 영역을 가지고 와서 채운다. 굉장히 많은 데이터베이스의 사진들을 가지고 학습하기 때문에 학습량에 따라서 굉장히 자연스러운 결과물을 기대할 수 있다. 그러나 학습하는데 필요한 이미지의 양이 약 230 만장으로, 수정하는데 상당히 많은 양의 데이터와 시간이 필요하다.

2. 개요

본 논문에서는 위와 같은 2 차원상에서 특정 객체를 배경과 분리하게 될 경우 생기는 영역 유실의 문제점을 해결하기 위해 2 차원상의 디지털 이미지를 3 차원으로 재구성 하는 방법을 제시하고자 한다. 3 차원에서의 재구성을 통하여 편집하고자 하는 객체에 대해서 손쉽게 배경과 분리가 가능해진다. 2 차원 이미지를 3 차원으로 재구성 하기 위해서 포인트 클라우드 기술을 적용한다. 편집하고자 하는 대상의 이미지에 대한 다 시점 이미지를 가져오면, 이미지들의 메타 데이터와 각 화소 간의 상관관계를 바탕으로 오브젝트를 이루는 점의 집합으로 삼각분할법 및 메쉬를 입히는 방식으로 3 차원 좌표계 내에서 3 차원 객체를 손쉽게 만들어 낼 수 있다.[5] 이러한 3 차원 상의 특정 객체를 수정 및 사용자가 원하는 방향으로 투명하여 다시 래스터라이징하면 얻을 수 있는 디지털 이미지는 3 차원에서는 공간적인 이동이 가능해지므로 오브젝트의 구도를 바꾸는 것이 자유롭다.

그러나 3 차원 상에서 이미지에 대한 모델을 만드는 데에는 두가지 어려운 점이 있다. 첫번째로 이미지에 대한 포인트 클라우드를 생성하기 위해서는 동일한 이미지에 대해서 여러 각도에서 찍은 다시점 이미지 데이터가 필요하다는 점이다. 3 차원 모델을 구축하기 위해서는 다시점 이미지들의 화소간의 관계 값이 필요하다. 두번째로는 만약 다시점 이미지가 두 종 이상의 카메라로 찍혔을 경우, 다른 종 간의 내부 파라미터에 대한 보정 작업이 필요하다는 점이다. 서로 다른 카메라에 대해 내부 파라미터가 서로 다르기 때문에, 이에 따라 달라지는 이미지 중심점이나 초점 값에 대한 보정이 반드시 필요하다.

먼저 첫번째 문제를 해결하기 위해서 우리는 데이터 수집 단계에서 안드로이드 어플리케이션을 활용하는 방법을 생각했다. 사용자에 대한 접근성이 용이하고 손쉽게 이미지 데이터를 습득할 수 있기 때문이다. 실제로 구글 스토어에서 서비스 되고 있는 다양한 3D 모델링 구현 어플리케이션들을 보면 직관적인 가이드라인으로 사용자가 손쉽게 모델 구축을 위한 사진을 찍을 수 있도록 구현해 놓았다. 두번째 문제에 대해서는 오픈소스로 제공되는 OpenMVG 를 사용한다.[6] OpenMVG 에서 제공되는 incremental pipeline 을 통하여 하나의 카메라에 대한 내부 파라미터

값만 가지고 있으면, 그 외 다른 종류의 카메라로 찍은 사진들의 내부 파라미터 값도 어느정도 보정이 된다.

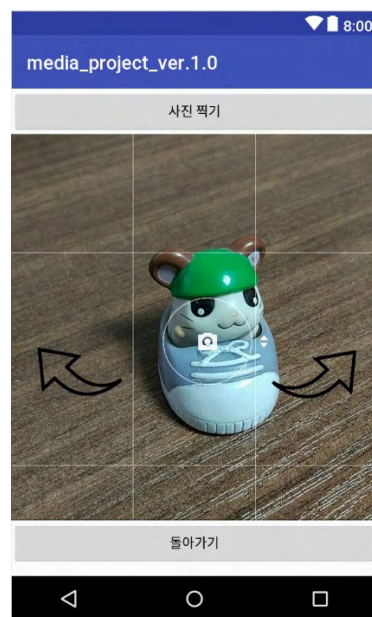
3. 데이터 수집을 위한 어플리케이션 구축

수집 단계에서 사용될 안드로이드 어플리케이션에는 사용자 시점에서 적절한 UI 를 사용하여 사용자가 알기 쉽게 다초점 사진을 찍을 수 있게 하는 카메라 기능과 개발자 시점에서 찍은 이미지들의 메타 데이터 정보를 따로 저장 및 열람이 가능한 열람 기능이 필요하다.



(그림 2) 어플리케이션 구조

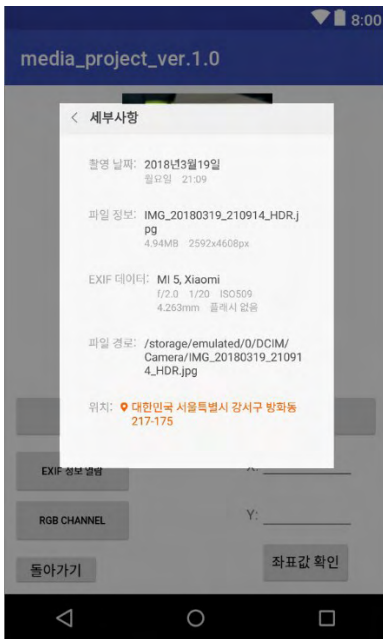
사용자는 해당 어플리케이션을 통하여 특정 이미지를 찍을 때, 사진을 찍은 장소로부터 오른쪽 혹은 왼쪽으로 돌아가며 추가적으로 사진을 찍을 수 있다. 해당 가이드라인은 그림 3 에 보여진 것과 같이 시각적인 UI 를 이용하여 표현될 수 있다.



(그림 3) 영상 촬영 기능

가이드라인 방향대로 오른쪽 또는 왼쪽으로 돌려가며 사진을 찍기 때문에 3D 모델링에 필요한 다 시점 이미지 정보를 얻을 수 있고, 이미지의 개수가 수십장에 불과하기 때문에 데이터 수집량에서도 합리 적이다.

사용자가 어플리케이션을 이용하여 사진을 찍 으면, 해당 이미지와 메타 데이터 정보가 일차적으로 찍은 스마트폰에 저장되며, 이차적으로 어플리케이션 과 연동된 서버의 데이터베이스에 저장된다. 저장될 때에는 각 이미지에 대해서 텍스트 파일 형태로 저장 되며, 사용자는 그림 4 와 같이 자유롭게 스마트폰에 저장된 메타 데이터 파일을 열람할 수 있다.



(그림 4) EXIF 열람 기능

4. 데이터 가공을 위한 오픈소스 활용

사용자에 의해 받아진 일련의 메타 데이터를 가지고 3D 모델 구축을 위해 필요한 데이터는 크게 3 가지로 나뉜다. 첫번째로는 카메라 내부의 정보를 가 지고 있는 내부 파라미터이고, 두번째로는 카메라와 찍은 이미지 사이의 거리 관계를 나타내는 외부 파라 메터, 그리고 마지막으로 이미지들간의 상관관계에 따라 구현될 수 있는 3D 좌표계 상의 좌표 값이다. OpenMVG 는 크게 4 가지 단계를 거쳐서 이러한 정보 를 가지는 3D 모델을 만든다. 첫번째로 ImageListing 과정이 필요하다. 이 과정에서 받은 메타 데이터들을 기반으로 각 이미지들의 높이와 너비 정보, 그리고 내부 파라미터 정보를 json 형식의 파일에 저장한 다.[8] 내부 파라미터의 경우 라이브러리에 존재하는 카메라 센서 정보 파일을 이용하여 동일한 모델에 대 한 내부 파라미터 정보를 저장한다. 이때 메타 데이 터가 존재하지 않는 이미지의 경우 이미 메타 데이터 가 존재하는 이미지들의 집합에서 구한 내부 파라메 터 값을 가지고 자체적인 초점 계산 공식인 식(1)을

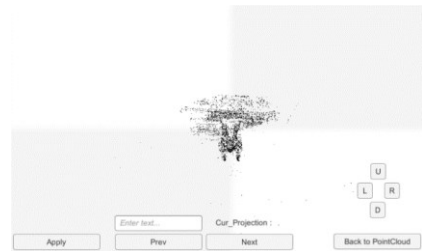
이용하여 보정된 초점 값을 구한다. 해당 공식에서는 메타 데이터가 없는 이미지의 너비 또는 높이의 최대 값에 초점 길이를 곱하고, 해당 값을 미리 정의된 센 서 값으로 나누어서 구한다.

$$focal_{pix} = \frac{max(w_{pix}, h_{pix}) * focal_{mm}}{ccdw_{mm}} \quad (식 1)$$

두번째 과정인 ComputeFeature 에서는 이미 지들의 특징들에 대한 정보들을 추가한다. 각 이미지 의 로컬 영역 또는 피쳐에 대한 정보를 추가한다. 세 번째 ComputeMatches 과정으로 들어가게 되면 이미지 들 사이의 관계를 파악해 시각적으로 겹치는 부분에 대해서 해당 경로를 추적하여 저장한다. 그리고 마지 막으로 IncrementalSfM 과 GlobalSfM 단계에서는 보 정된 내부 파라미터 데이터 값을 가지고 자세 추정 및 삼각 측량법을 사용하여 객체의 3D 정보를 가지는 점들을 가지고 계산된 좌표 값들을 하나씩 추가하면 서 포인트 클라우드를 구축한다.[9]

5. 구축 결과

OpenMVG 의 결과값인 내부 파라미터와 외부 파라미터 정보를 가지고 3 차원 공간 위에 좌표 값들 에 해당하는 구 형태의 오브젝트를 위치하는 방식으 로 구현하기 위해 그림 5 와 같이 3D 모델링 툴을 사 용하였다.



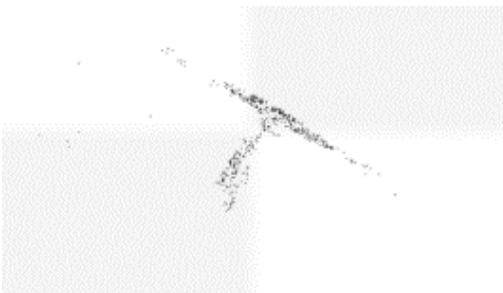
(그림 5) 이미지 뷰 모습

사용자가 편집 프로그램을 사용하면 먼저 찍어 놓은 사진들을 바탕으로 구현된 포인트 클라우드 정보 를 불러오게 된다. 불러온 포인트 클라우드를 조작할 수 있도록 회전 버튼이 제공되며, 사용자는 원하는 대 로 본인이 찍었던 위치정보에 오브젝트가 투영된 결과 물을 텍스트 입력창과 “Prev”, “Next” 버튼을 통 하여 열람할 수 있게 된다. 텍스트 입력창에는 원하는 투영 오브젝트의 번호를 입력하면 되고, “Prev” 버 튼에서는 현재 투영 오브젝트의 바로 전 번호로, “Next” 버튼에서는 현재 투영 오브젝트의 다음 번호 로 이동할 수 있다. “Back to PointCloud” 버튼을 누르면 다시 원래 포인트 클라우드를 바라보는 카메라 뷰로 돌아오게 된다. 그림 6 에서는 5 번째 뷰로 투영 된 이미지의 모습을 보여주는 것이다.



(그림 6) 이미지 투영 모습

만약 사용자가 오브젝트를 조작하고 싶다면 4 방향으로 회전이 가능한 회전 기능을 이용하면 된다. 사용자는 동, 서, 남, 북 방향으로 들어가는 형태로 회전이 가능하며 사용자가 원하는 만큼 회전할 때까지 버튼을 누르고 있으면 된다. 포인트 클라우드 오브젝트는 1 회당 5 도씩 회전되며 이 때 각 카메라 뷰에 투영된 이미지들 역시 회전된 만큼 동일한 각도로 움직인다. 회전을 시키는 도중 투영 결과물을 바로바로 확인할 수 있으며, 투영된 오브젝트는 사용자가 입력한 회전 값만큼 회전된 포인트 클라우드가 투영된 형태로 업데이트 된다. 그림 7 은 객체를 동쪽 방향으로 45 도 회전시킨 결과이다.



(그림 7) 객체 45도 회전 결과

6. 결론

본 논문에서 소개하는 포인트 클라우드를 이용한 2 차원 이미지의 3 차원 모델링화는 포인트 클라우드라는 요소를 이용하여 2 차원에서 할 수 없는 제한적인 기능들을 3 차원을 이용하여 해결할 수 있을 것이다.

예를 들어 첫번째는 롤리캠과 같은 실시간 이미지 합성 서비스가 있다. Face detection 을 통하여 인물사진에 래스터 그래픽적인 요소들을 합성 시켜줄 수 있는데, 래스터 그래픽으로만 표현하기 때문에 사람의 얼굴 윤곽에 정확히 맞추어 합성되지 않는다. 만약 사용자 사진에 대한 포인트 클라우드를 구축할 수 있다면 두상에 대한 3D 정보를 가지고 있기 때문에 가상의 요소들을 얼굴에 합성할 때 보다 정밀한 합성이 가능할 것이다.

다른 예로는 포토샵 편집 서비스가 있다. 기존의 편집 방식은 내용 인식 채우기 기능을 사용하여 삭제된 부분을 채우기 때문에 상황에 따라서 편집된 이미지 퀄리티가 상이할 수 있다. 그러나 해당 이미

지에 대한 포인트 클라우드를 구축할 수 있다면 이미지 내부의 어떠한 객체 요소를 편집해도 소실된 영역에 대해서 3 차원 정보를 가지고 있기 때문에 복원이 가능하게 될 것이다.

결론적으로 본 논문에서 소개하는 포인트 클라우드를 이용한 2 차원 이미지의 3 차원 모델링화는 기존 이미지 편집 서비스에 대한 한계점을 보완하고, 더 나아가 3 차원 정보를 활용하여 다양한 서비스에 활용 가능한 장점을 가지고 있다. 그러나 아직 복잡성에 대한 검증이 제대로 이루어지지 않았기 때문에, 시간적, 비용적 측면에서 기존의 방식보다 효율적인 부분이 있다는 것이 입증이 된다면 충분히 상업적으로 적용이 가능할 것이다. 따라서 앞으로 이러한 부분에 대한 연구를 더 진행할 예정이다.

참고문헌

- [1] Rodriguez, Edward (2008). Computer Graphic Artist. Netlibrary. p. 163. ISBN 81-89940-42-2.
- [2] Y. Wexler, E. Shechtman and M. Irani. "Space-Time Video Completion"
- [3] Mike Rossner, Kenneth M. Yamada, What's in a picture? The temptation of image manipulation
- [4] James Hays and Alexi A. Efros. "Scene completion using millions of photographs" (Carnegie Mellon University, 2007)
- [4-2] James Hays and Alexi A. Efros. "Scene completion using millions of photographs" (Carnegie Mellon University, 2007)
- [5] Rusinkiewicz, S. and Levoy, M. 2000. QSplat: a multi-resolution point rendering system for large meshes. In Siggraph 2000. ACM , New York, NY, 343-352.
- [6] Fredric S. Young. Method for general image manipulation and composition
- [7] Pierre Moulon. OpenMVG: Open Multiple View Geometry
- [8] https://openmvg.readthedocs.io/en/latest/software/SfM/SfMInit_ImageListing, OpenMVG : ImageListing
- [9] <https://openmvg.readthedocs.io/en/latest/software/SfM/IncrementalSfM>, OpenMVG : IncrementalSfM