

Computer Vision PA2 : Structure-from-Motion

20175051 Hoyeon Kim

Department of Physics and Photon Science
Gwangju Institute of Science and Technology
Gwangju, Korea

I. INTRODUCTION

SFM (Structure From Motion)은 특정 오브젝트를 다른 카메라 각도로 찍은 여러 장의 사진으로부터 오브젝트의 3D points를 구하는 알고리즘입니다. 먼저 사진들의 특징점을 SIFT를 이용해 구합니다. 이후 Two-view Initialization Step을 통해서 초기 3D points를 구합니다. 본 과제에서는 이것이 주어졌습니다. 이후에 More than Three-view Growing Step을 통해서 모든 이미지를 이용해 3D points를 구합니다. 이후에는 Bundle Adjustments 과정을 통해 구한 3D points를 최적화합니다. 본 과제에서는 3번과 4번의 이미지를 이용하여 Initialization Step이 수행되었고 이후는 아래의 과정에 따라서 진행됩니다.

- 1) SIFT를 이용하여 모든 이미지의 keypoints와 descriptor를 추출합니다.
- 2) 3D points를 생성할 이미지를 선택합니다. 이는 BF-Matcher의 KNNMatch를 이용하여 현재 존재하는 3D points의 descriptor와 가장 많은 match를 가지는 이미지를 선택합니다.
- 3) RANSAC을 이용하여 선택된 이미지의 Camera Pose를 구합니다. 임의로 3개의 match를 선택하고 이를 이용하여 이미지 포인트(2d point)와 오브젝트 포인트(3d point)를 구합니다. 구해진 2d point, 3d point와 intrinsic matrix를 SolveP3P에 넣어 Camera Pose를 구합니다. 구한 Camera Pose를 이용해 모든 매치에 대해서 3d point를 선택된 이미지에 projection 시키고 2d point와의 거리를 계산합니다. 이 거리가 특정 임계값보다 작은 inlier의 개수를 구합니다. 가장 많은 inlier를 가진 Camera Pose를 선택합니다.
- 4) 3D points 생성이 완료된 이미지 중 현재 선택된 이미지와 가장 많은 match를 가지는 이미지를 선택합니다. 현재 선택된 이미지를 best to image, 이미 완료된 이미지 중 현재 스텝에서 선택된 이미지를 best done image라고 하겠습니다.
- 5) best to image 와 best done image 의 Match 를 이용하여 각각의 keypoints 를 구합니다. 이를 inverse intrinsic matrix 를 이용하여 normalized plane으로 옮깁니다.
- 6) best to image와 best done image의 Camera Pose를 이용하여 새로운 3d points를 생성합니다. 이는 Triangulation 방법을 사용합니다. 각 매치된 포인트와 Camera Pose 를 이용하여 Triangulation 방법에서 제시하는 행렬 A를 구하고 특이값 분해를 통해 새로운 3d points 를 생성합니다. Projection Matrix 는 Camera Pose에 intrinsic matrix를 곱한 것입니다. 그런데 normalized plane 에서 reprojection error를 구하기 때문에 Projection Matrix 에 inverse intrinsic matrix를 곱해줍니다. 즉, Projection Matrix는 Camera Pose입니다.

- 7) best done image의 Match 포인트가 이미 3D에 올라가 있다면, 3D 에서 distance를 구하고 이것이 임계값보다 작다면 새로운 3d point로 선택합니다.
- 8) best done image의 Match 포인트가 3D에 올라가 있지 않다면, 이를 best done image의 camera pose를 이용하여 normalized plane에 projection시킵니다. 그리고 이와 매칭되는 best done image의 2d point와 distance를 구하고 이것이 임계값보다 작다면 새로운 3d point로 선택합니다.
- 9) 새롭게 추가된 3d point는 다음 이터레이션에서도 활용됩니다. 모든 이미지에 대해서 3d point를 생성합니다.
- 10) 위의 과정에서 얻어지는 정보를 이용하여 Bundle Adjustment를 수행합니다.

II. METHODOLOGY

Introduction에 소개된 프로세스를 구현하여 결과를 얻었습니다. 부가 설명을 하겠습니다.

A. Selection of best to image and best done image

바로 옆의 이미지가 가장 많은 매치가 될 것이라고 예상됩니다. 그러나 컵의 손잡이 부분의 feature extracting이 제대로 되지 않아서 14번의 이미지는 정반대에 위치에 있는 6번의 이미지와 가장 많은 매칭 포인트를 가지고 있습니다. 이를 해결하기 위해 best to image 를 선택하는 과정에서 이미 완료된 이미지들의 바로 양 옆의 이미지 중 매치가 가장 많은 것을 선택했습니다. 예를 들어 초기에는 3번, 4번이 이미 완료된 이미지들이고, best to image를 선택하기 위해 2번과 5번을 비교합니다. 그리고 best done image를 선택하는 과정도 양 옆의 이미지 중에서 가장 매치가 많은 이미지를 선택하게 합니다.

B. RANSAC threshold

Threshold를 1을 이용하였습니다.

C. Triangulation Process

A에서 설정한 이미지를 선택하는 순서 때문에, Triangulation을 바로 옆의 이미지와 수행합니다. 그래서 서로 겹치는 부분이 많게 되고, 카메라 포즈를 이어서 생기는 마름모꼴의 영역이 길어지게 됩니다. 따라서 Normalized Plane에 Projection 시켰을때 가까운 곳에 있지만, 이미 존재하는 3d points들과 거리가 먼 곳에 새로운 점이 생길 수 있게 됩니다. 이를 해결하기 위해서 기존의 3d points들의 $[x, y, z]$ 각각 평균과 표준편차를 구해 $[\text{mean} - 4 \cdot \text{std}, \text{mean} + 4 \cdot \text{std}]$ 사이에 존재하는 점을 선택합니다.

D. Re-projection threshold

12, 13, 14번 이미지는 컵의 손잡이 부분이 있어서 많은 매칭이 되지 않아 3d points들이 적게 생성됩니다. Bundle Adjustment 를 수행하기 위해서는 적어도 1개의 점이 생성되어야 하므로 re-projection threshold 를 $5e-3$ 을 이용하였고 나머지 이미지는 $5e-4$ 을 이용하였습니다.

III. BACKGROUND

Bundle Adjustment를 수행하는 코드는 Jacobian 행렬과 Levenberg-Marquardt 최적화 알고리즘을 사용하여 3D Reconstruction의 정확도를 높이고 있습니다. Jacobian 행렬과 Levenberg-Marquardt 최적화 알고리즘에 대해 아래에서 설명하겠습니다.

A. jacobian matrix

Jacobian matrix는 벡터 함수의 1차 편미분을 계산하는 값입니다. 함수 f 가 n 차원에서 m 차원로의 함수일 때 x 는 x_1 부터 x_n 까지 가능하고 f 는 f_1 부터 f_m 까지 가능합니다. Jacobian matrix의 (i,j) component 는 f_i 를 x_j 로 편미분한 값을 가집니다. 이를 이용하여 입력의 변화에 대해서 벡터 함수의 출력이 어떻게 변할 예정인지 근사적으로 예측가능합니다. 따라서 비선형 문제에서 최적화 해를 찾기 위해 사용됩니다.

B. Levenberg-Marquardt Algorithm

Levenberg-Marquardt Algorithm은 최소 제곱 문제를 최적화하기 위해서 사용됩니다. 최소 제곱 문제를 최적화하는 기법에는 Newton's Method 와 Gradient Descent이 있습니다. Newton's Method는 2차 미분 행렬을 이용하여 해를 찾는 데 이는 비선형 문제에서 잘 동작하지 않을 수 있습니다. 왜냐하면 로컬 미니멈을 탈출하는 방법이 필요하기 때문입니다. Gradient Descent는 1차 미분 행렬을 이용하여 해를 찾는 데 수렴 속도가 느립니다. 정리하자면, Levenberg-Marquardt Algorithm은 Newton's Method의 수렴 속도와 Gradient Descent의 안정성을 선택한 알고리즘입니다. λ 라는 변수를 사용하여 수렴을 조절합니다.

C. Usage of Bundle Adjustment

Camera Pose를 구하는데 발생하는 오차와 이미 오차가 발생한 Camera Pose를 통해 Reconstruct한 3d points의 오차를 복원하기 위해 Levenberg-Marquardt Algorithm을 사용합니다. 먼저 Camera Pose, 3D points를 이용하여 초기값을 설정합니다. 그리고 모든 3D points를 모든 Camera Pose 를 이용하여 2D points로 Projection 시킵니다. Projection 시킨 2d points와 원래의 image 2d points의 오차의 제곱합을 Cost로 정의합니다. 이 Cost를 Levenberg-Marquardt Algorithm을 이용해 최소화 시켜 Bundle Adjustment를 수행합니다. λ 값을 적절하게 조정하여 속도와 안정성을 조절하고 Cost 값이 일정 이하가 될때 까지 반복합니다. 각 반복마다 Camera Pose와 3D points 들이 최적화됩니다. Growing Step에서 각 이터레이션의 마지막마다 이를 수행하면 결과가 더 좋아질 것입니다.

IV. RESULT

Growing Step을 통해서 구한 컵의 3d points는 그림 1, 2와 같습니다. Growing Step이후 Bundle Adjustment까지 수행한 컵의 3d points는 그림 3과 같습니다.

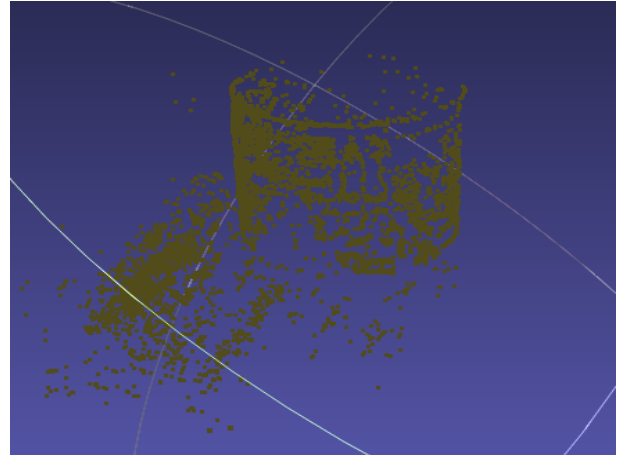


Fig. 1: reconstructed 3d points

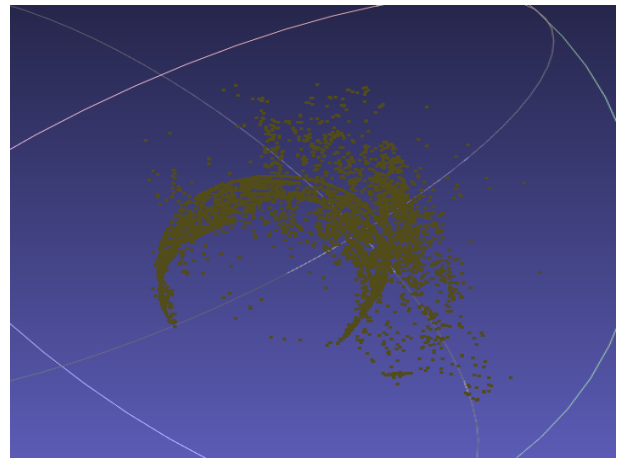


Fig. 2: reconstructed 3d points, 다른 방향 캡처, 컵의 손잡이 부분이 없습니다.

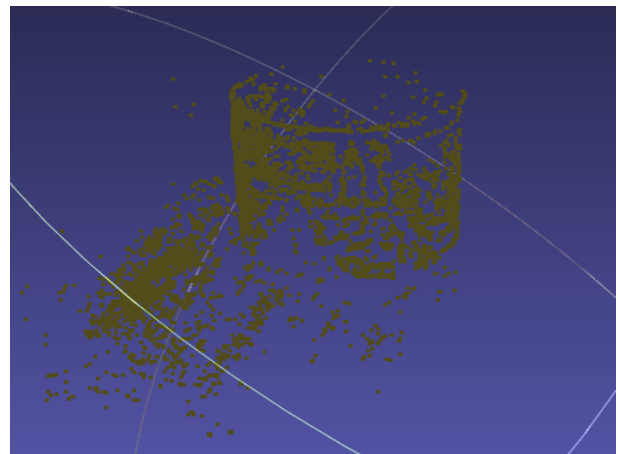


Fig. 3: reconstructed 3d points with bundle adjustment