CSE 586 Project2 Report

Shimian Zhang, Hong Zhang, Wenxing Cao, Yaqi Xu

May 1, 2020

1 Introduction

In this project, we focus on using the publicly available COLMAP software for 3D reconstruction[1] to recover a 3D point cloud representing the structure of a scene viewed from overlapping camera views. Then, based on the 3D point cloud results, we need to find the dominant plane of the scene, which is provided for placing a virtual 3D object. In addition, we also place a virtual 3D object on the dominant plane of the scene and project it back to the original images. The reconstruction and project bring lots of difficulties for our implement.

1. To find the dominant plane, a inlier classifier is need to find the inliers of dominant plane in the scene. And the 3D points which is closed to the dominant plane could not form a smooth surface directly. Because of the computing error of COLMAP and different overlapping camera, they are above or below the actual dominant plane.
2. To place the 3D object, the world coordinate system, the camera coordinate system and the object coordinate system should be computed.
3. To project the object back to the 2d image, the transformation matrix between the placed 3d object and cameras should be computed too.

To solve these difficulties, we propose to use the method RANSAC[2] to find the inlier set of dominant plane then fit a plane by these inliers and do the geometric transforms of coordinate system based on the knowledge of camera calibration, DLT calibration, Triangulation.
For recover a 3D point cloud, we use 16 pictures from different overlapping camera. A 3D object 'chicken' is placed on the dominant plane in the scene. To implement the project, there are 8 steps to apply a virtual object in the image, which is shown below:

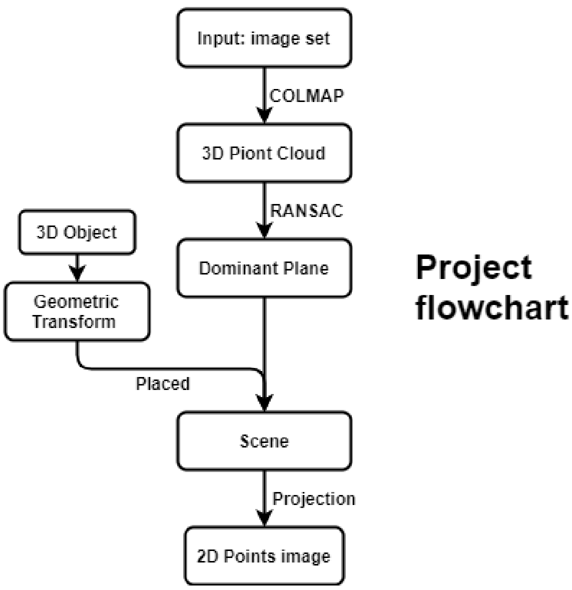


Figure 1: Project flow chart

2 Related Work

2.1 Structure from Motion

Structure-from-motion(SfM) is the process of reconstructing 3D structure from its projections into a series of images taken from different viewpoints. In the COLMAP reconstruction tools, 3D reconstruction from images uses the Incremental Structure-from-Motion strategy. Incremental SfM is a sequential processing pipeline with an iterative reconstruction component. The Incremental SfM is usually divided into two main parts. The first part is the Correspondence Search. It commonly starts with feature extraction and matching, followed by geometric verification. The second part is the Incremental Reconstruction. The resulting scene graph serves as the foundation for the reconstruction stage before incrementally registering new images, triangulating scene points, filtering outliers, and refining the reconstruction using bundle adjustment.

2.2 RANSAC algorithm

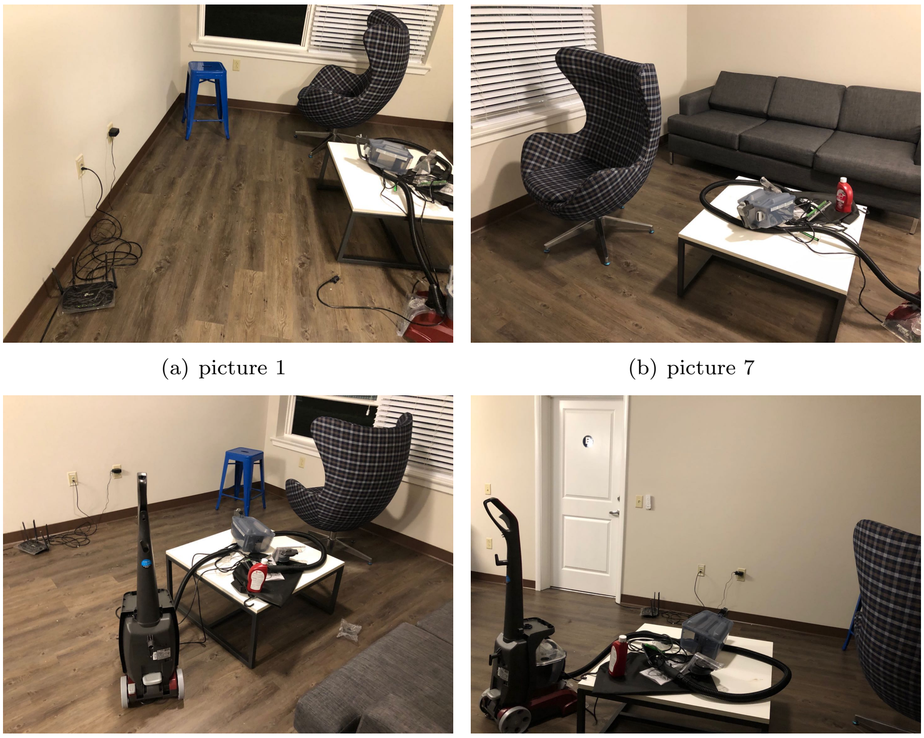
In our project, we are not just based on the traditional RANSAC algorithm to set the threshold of the inliers and assign the probability. We use a trick to achieve the RANSAC application. Comparing the RANSAC algorithm, we assign the limitation of the inliers, which is the distance between points and the model plane. Then, we replace the probability by the maximum number of points instead the proportion inliers over all points set. In order to get the most inliers model plane, we just set the iteration number is large enough to get the accurate one instead of calculate the iteration numbers. Actually, we just change the equation of the RANSAC algorithm to get what we want.

3 Methodology

In this section, we are going to introduce the COLMAP software, RANSAC method. And show how we use geometric transformation method transform the coordinate system of object, 3D point cloud.

3.1 COLMAP

We implement COLMAP software for the recover of 3D point cloud from pictures. Image-based 3D reconstruction from images traditionally using Structure-from-Motion strategy. Structure-from-Motion is the process of reconstructing 3D structure from its projections into a series of images. The input is a set of overlapping images of the same object, taken from different viewpoints. The output is a 3-D reconstruction of the object, and the reconstructed intrinsic and extrinsic camera parameters of all images. This output then serves as the input to Multi-View Stereo to recover a dense representation of the scene.



(a) picture 1 (b) picture 7

(c) picture 10 (d) picture 16

Figure 2: Pictures taken by camera(16)

COLMAP provides an automatic reconstruction tool that simply takes a folder of input images and produces a reconstruction in a work-space. So, we could apply images taken by overlapping camera into COLMAP software and get the recovered 3D point cloud of the structure in picture.

For the image which we will apply into COLMAP, the scene in them need to fulfill some rules to build the 3D point cloud. The “scene” needs a dominant planar surface to put the virtual object. This surface should have lots of texture provided for COLMAP to find and match features.
Then, after feature extraction, matching, incremental reconstruction and bundle adjustment, we will get a 3D cloud of points.

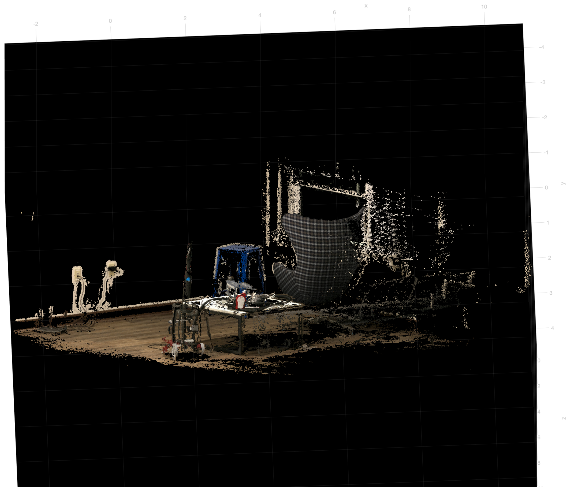


Figure 3: The scene reconstructed by COLMAP (3D point cloud)

In the picture, we provide a dominant surface “floor”, and the surface has a lot of texture for COLMAP to find and match.

3.2 RANSAC

The Random Sample Consensus (RANSAC) algorithm proposed by Fischler and Bolles [2] is a general parameter estimation approach designed to remove outliers when a large proportion of outliers in the input data. RANSAC is a re-sampling technique that generates candidate solutions by using the smallest set possible and proceeds to enlarge this set with consistent data points. The basic algorithm is summarized as follows:

1. Randomly select three points required to determine the model plane
2. examine the distance of points from the set of all points to the model plane is less or equal than ϵ , then they are inliers
3. examine the numbers of the inliers of model plane and save the number and position data of inliers
4. If the number of inliers of this model is more than before, save the inliers position and the data of model plane. Then, repeat N times until the number is the maximum.

From the traditional RANSAC method, we need to calculate the number of samples, it could get from the equation below.

$$1 - (1 - (1 - e)^s \ )^N = p \tag{1}$$

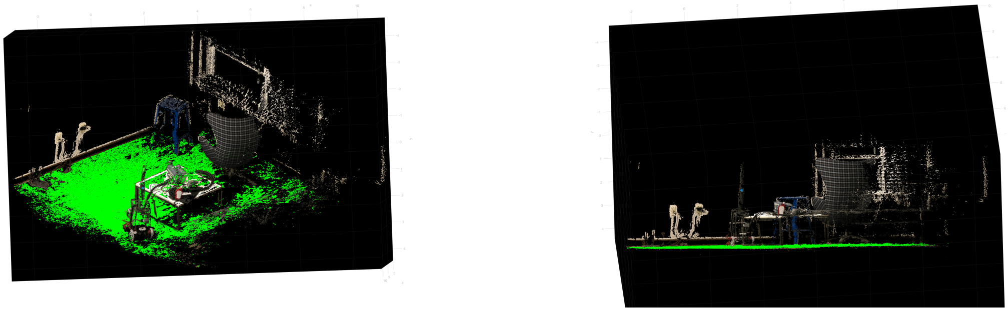
- e = probability that a point is an outlier
- s = number of points in a sample
- N = number of samples (what we want to compute)
- P = desired probability that we get a good sample (usually is 0.99)

Or we could have another expression to compute:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - e)^s)} \tag{2}$$

Based on the RANSAC method, we assign the probability p is equal to 0.99, the probability e is equal to 0.4442, so we could get the approximate iteration number is equal to 25.3.
So in our project, in order to get the model plane, we assign the distance ϵ between inliers and model plane is 0.03, and the iteration number N is 600. And we want to find the maximum probability of p, which means the

maximum number of inliers. After iteration, we find the maximum number inliers and change the color to show them.



(a) Dominant plane inliers from camera view (b) Dominant plane inliers from Horizontal angle

Figure 4: Dominant plane inliers

3.3 Plane Building

After RANSAC method, we just get the most inliers points position of the dominant plane, but these points could not form a smooth plane. So we need to define a coordinate plane using these inliers.A simple least squares solution could achieve the goal.
The equation for a plane is:$ax + by + cz + d = 0$.We assume c=1. So set up matrices like this with all the inliers:

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \dots & \dots & \dots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ \dots \\ z_n \end{bmatrix} \tag{3}$$

In other words:

$$Ax = B \tag{4}$$

Now solve for x. We could get a simple expression of above euqation to solve the matrix x.

$$\begin{bmatrix} a \\ b \\ d \end{bmatrix} = (A^T A)^{-1} A^T B \tag{5}$$

So, using above equation, we could get the plane parameters $[x(1), x(2), 1, x(3)]$ to determine the dominant plane.

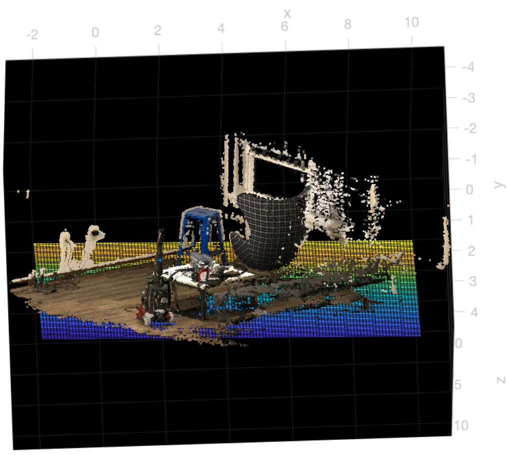


Figure 5: The Dominant plane

3.4 Homogeneous Coordinate

Aiming to place the object on the dominant plane in 3D point cloud, we build a designed coordinate on the dominant plane. then compute the transformation matrix to transform the object coordinate system into the designed coordinate system(build on dominant plane). There are several steps needed to compute the transformation matrix:

1. Build the local coordinate system of the object [Up-o, Forward-o, Left-o], and its origin of the object is near the origin of the world coordinate system. the UP, Forward and Left vector denotes the pose of the 3d object.

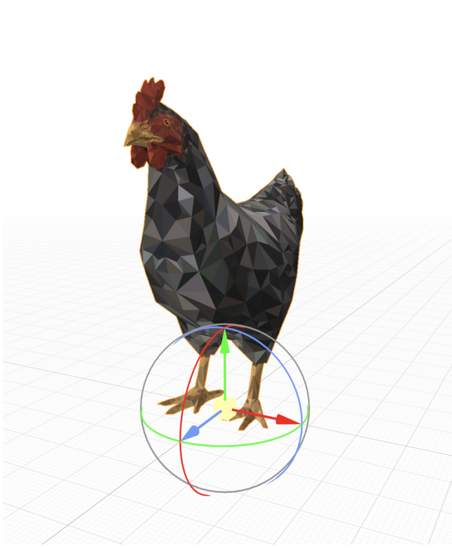


Figure 6: Build local object coordinate system

2. Then, we construct a designed coordinate system [Up, Forward, Left] at the position $\begin{bmatrix} 2 & 1 & 7.3387 \end{bmatrix}$ where we want to put the object. In the figure 7, the designed coordinate (Up, Forward, Left) is represented by (red, blue, green unit vector) on the ground.

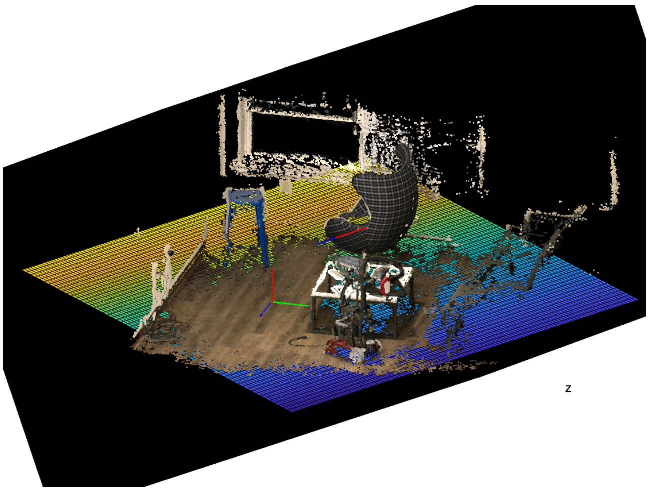


Figure 7: The designed coordinate system in 3D point cloud

3. Firstly, Rotate the object coordinate system to be same as the designed dominant plane coordinate system
4. Secondly, translate the rotated object to the position designed at dominant plane

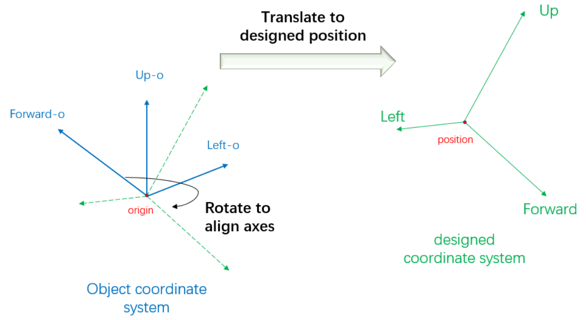


Figure 8: Homogeneous coordinate transformation

Based on the matrix form of Homogeneous Coordinate:

$$P_C = RP_W + T \tag{6}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \tag{7}$$

Combining the rotation matrix and the translation matrix of the object, we could get the whole transformation matrix of the 3D model. Then, we apply into the object and rotate the object to fit the designed coordinate system of the dominant plane. we could get the scene with placed object as the following figure 9:

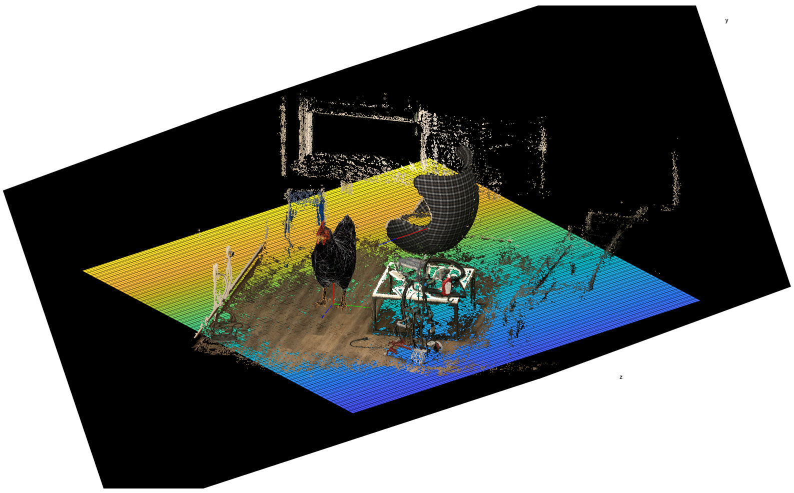


Figure 9: Object placed on dominant plane

3.5 Projection

To project the 3D points of 3D model, we first need to find the camera position of each picture, which we could get from the COLMAP export files. Then, for the projection of points, we need to find the transform matrix of each camera coordinate system to the world coordinate system, which is like the transform we did on 3D model. And reverse the transform. Now, we project the points of model on the image. But, for the complex model, the forward faces would cover the faces behind based our impression. So the next step is to make the model more real. The detail method would be shown below.

Model Projection When we apply image into the COLMAP tools, we could get the 3D point cloud and the camera location data.

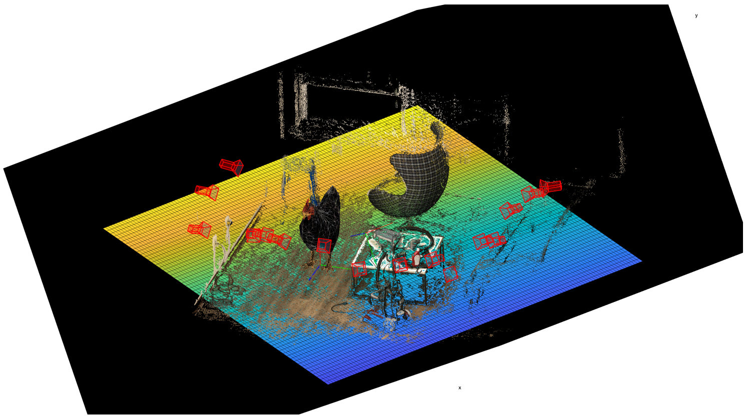


Figure 10: The cameras location

Then, we just use the camera position and rotation matrix R and shift matrix T. From the COLMAP tools, we use the rotation matrix R and shift matrix T to put the points of image into the world coordinate system. Now, we just use the inverse rotation matrix R' and T'.

$$R'P_C - R'T = P_W \tag{8}$$

Then, the P_W is the data of 3D model in World coordinate system, and the P_C is the camera coordinate system. Now we could get the 3D model data projected in camera coordinate system. Next, we need to project the camera coordinate system to the 2D image. So, based on the perspective projection method, we could get the projection matrix:

$$\begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \\ 1 \end{bmatrix} \tag{9}$$

Because the 2D image could be presented by matrix [x y], so we assign the z = 1 in our matrix. In addition, for the world wide coordinate system data [u v w], we distort the coordinate system to [x',y',1]. But the distortion is a nonlinear distortion. For the camera model, we used to get the camera data is the Simple Radial camera model in COLMAP. So the distortion equation could be represented as :

$$x' = u * (1 + k(u^2 + v^2)) \tag{10}$$

$$y' = v * (1 + k(u^2 + v^2)) \tag{11}$$

In the equation, k is the internal parameter of camera, which we could get from the camera data after COLMAP tools. [u,v,w,1] is the points data which is get from the transformation of model points[U,V,W,1] in world coordinate system .

$$\begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \tag{12}$$

Then, the placed model could be projection on our image which is taken by camera.
But after projection finished, we find a problem that all the points of model are been projected onto the image. Those points contain the points behind the model which are on faces blocked. In the real world, the optical light could be blocked by the object, so we could not see the points behind the model. In order to solve this problem, we use a method(based on Z-buffer concept) to sort the points:

1. Compute the distance of each face to the camera location, which is based on the normal vector of the faces

2. Sort the faces from furthest to nearest (Z-buffer concept)

3. Use those faces filling the polygons from furthest to nearest, so the further faces polygons would be covered by the nearer one

4 Results

In this section, we are going to show the final results that placed 3D object project on the camera view image, and analyse the problem about the projection image.

(a) picture 1

(b) picture 7

(c) picture 10

(d) picture 16

Figure 11: Pictures after projection(16)

From the result picture above , there are several key points we could get.

- The object is just placed on the design dominant plane(floor) perfectly as the image shown in picture 1. Its foot center point is just on the dominant plane.
- For those points in object behind the surface, the projection picture just show the faces we could see from camera view and block the points behind. (projection picture 7)
- The problem of this projection is the projected model could cover on the image instead of based on the structure of the 3D scene. So, in some cases, the model would been seen as stand on the table or float on the image, which is shown not real. This is because the image cover the items of the original pictures.(projection picture 10 and 16)

If we want to avoid the problem and make the object shown more real, we will need to projection the whole scene based on the 3D points, and analyze the structure relationship between the object in the whole scene.

5 Conclusion

In this project, we implement reconstruction 3D scene by COLMAP software, 3D object placement and Projection from 3D model to images. By object placement and projection, we design the transform method and projection method basing on Pinhole Camera Model homogeneous transformation.
After experiment, we've implement the placement of the object to design dominant plane and the optical light blocking of the model. However, in some cases, the object might block the points of image to make the object shown not real. To solve this problem, the projection need to be implement by whole scene points and analyze the structure relationship between 3D model and reconstructed scene items.
Due to time reason and the knowledge limitation, the projection image is based on the reconstruction 3D points, which could not show well. And the structure of the 3D model and reconstruction scene item is complex, we need to design a method to find the surfaces to shown in our image. But by implementing the object placement and picture projection, we gain more experience and method on 3D reconstruction. We also solve some problems to make the object shown more real basing on the recognition of physical knowledge.

6 Division of Works

Team member:

- Shimian Zhang : Code for Projection part
- Hong Zhang : Code for Ransac algorithm and tranformation matrix computation part
- Wenxing Cao : Report writing
- Yaqi Xu : PPT writing

References

[1] Schönberger, Johannes Lutz and Frahm, Jan-Michael. Structure-from-Motion Revisited. Conference on Computer Vision and Pattern Recognition (CVPR),2016

[2] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, 1981.