Nicholas Sun Jun Yang | A0217609B

Code Process

Preprocess

detect_keypoints

1) Read the image using imread and then extract the kepoints and descriptors using cv2.SIFT_create() and sift.detectAndCompute

create_feature_matches

1) Create a list of 2 nearest neighbor matches from descriptor 2 per each descriptor 1

2) Filter the feature matches using the Lowe ratio test. Good matches are when the first match has distance less than the lowe_ratio * distance of the second best match.

create_ransac_matches

1) findEssentialMat is used to calculate the essential matrix and inlier mask

create_scene_graph

1) For each of the pair of images, first get the number of inliers between each images and if this number is larger than the given minimum number of inliers ,we connect the pair of images as an edge to the graph.

SFM

get_init_image_ids()

1) Iterate through the graph to find the pair with the largest number of inliers

get_init_extrinsics

1) Use recoverPose to recover the rotation matrix and translation vector and then combine these two to get back the camera pose

get_reprojection_residuals

1) Iterate through the 3D points and apply the computed camera matrix onto each 3D point to project it into a 2D point

2) Find the Euclidean distance between the reprojected 2D point and original 2D point

solve_pnp

1) Use solvePnP to get the rotation and translation vectors from 3D-2D point correspondences. Then use Rodrigues to convert the rotation vector to a matrix

2) Use get_reprojection_residuals with the computed extrinsics and together with 3D-2D point correspondences and instrinsincs to get the residuals

add_points3d

1) triangulate between the unregistered pair of images to get new 3D points.

get_next_pair

1) For each registered images, get an unregistered neighbor and then calculate the number of inliers between this pair and update the current maximum number of inliers as well as the image pair with these inliers.

2) At the end of the iteration, we get the image pair with the highest number of inliers which will be the next pair.

Bundle Adjustment

compute_ba_residuals

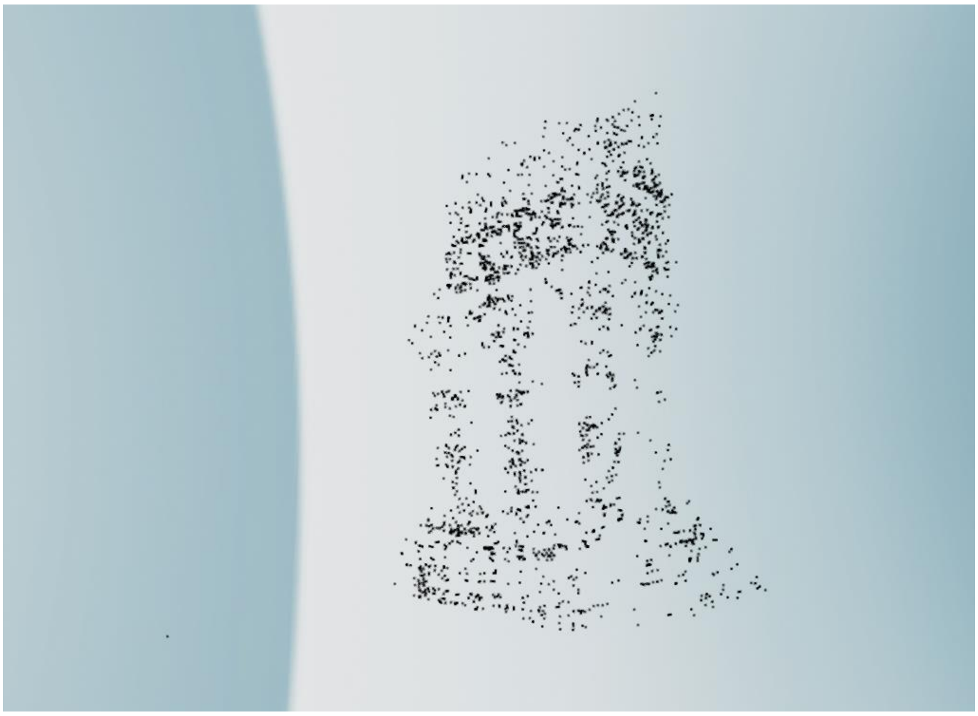
1) From the instrinsics and extrinsics, we first compute the projection matrixes for every corresponding instrinsic and extrinsic

2) After homogenizing the 3D points, the corresponding 3D points is retrieved using the parameter provided and the reprojected 2D points is then computed using these 3D points and the projection matrixes.

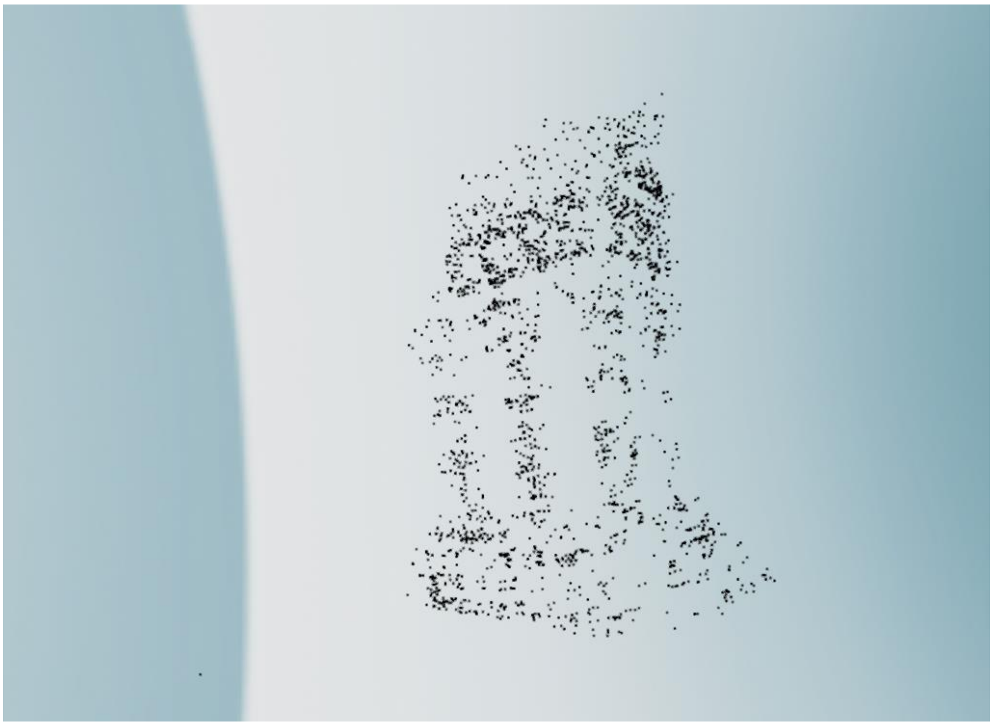
3) The Euclidean distances between the original and reprojected 2D points is then obtained through the formula sqrt(square(x_org – x_repro) + square(y_org – y_repro))

4) The distances are then summed to get the residuals of the bundle adjustment

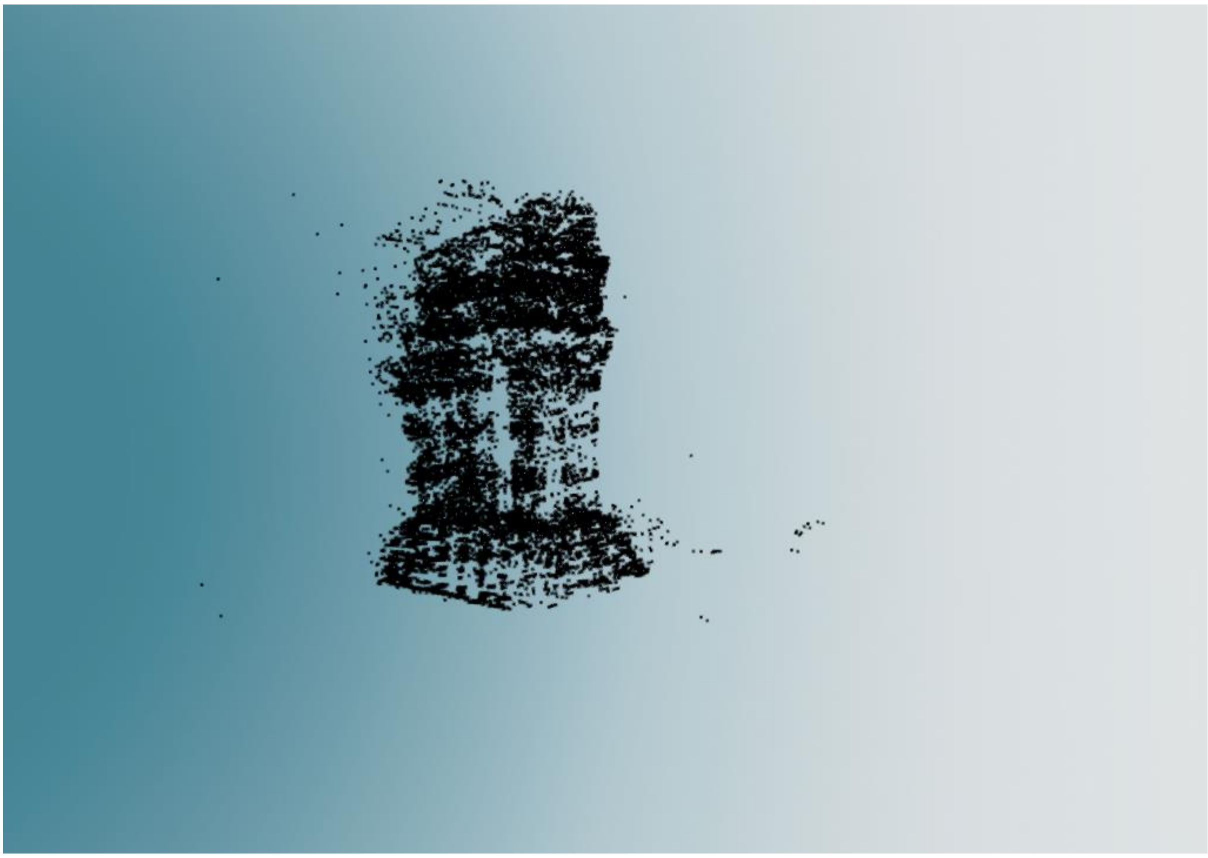
Results



Mini Temple visualized



Mini Temple with Bundle Adjustment visualized



Temple visualised