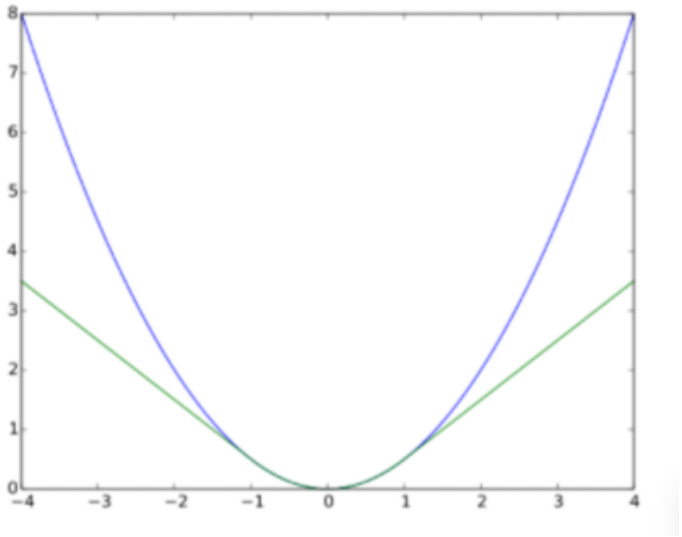

Main
show_extra_options
show_frame1
44
show_cam1
0
show_frame2
44
show_cam2
1
lock_frames
show_detected
show_matches
show_inliers
show_tracks
show_reprojections
show_outlier_obs
show_ids
next_step_hint
continue_next
next_step
detect_keypoints
clear_matches
match_stereo
match_all
match_bow
build_tracks
init_scene
camera_candidates
add_camera
add_landmarks
optimize
remove_outliers
save_map
load_map
Detected 159 corners
Detected 89 matches
Detected 142 corners
Detected 89 matches

Part3 Discuss in your answer PDF what a robust loss function such as Huber does and why we should use it here,

Not all noises follow the Gaussian Distribution, if there's some extreme outlier, the least square will punish them seriously, however Huber loss can alleviate this punishment. As the following graph roughly shows, the green line is the Huber loss and the blue line is the least square.



Part4 describe in your answer PDF the implemented criteria to detect outliers. For each criterion, what do you think might be the cause of such an outlier and why do we need to remove it?

1. OutlierReprojectionErrorHuge:
   if the reprojection error is larger than the threshold_huge_pixel;
   might be due to some noise
   remove it in case the unprecise of the map

2. OutlierReprojectionErrorNormal:
   if the reprojection error is larger than the threshold_normal_pixel;
   might be due to the unprecise camera intrinsic parameters
   remove it cause it might be wrong match points

3. OutlierCameraDistance
   If the 3d distance to camera is larger than then threshold;
   might be due to some mismatches
   too near to the camera, some distortion might happen.

4. OutlierZCoordinate
   if the z coordinate of some points is too smaller than the threshold
   might be some mismatches

Part5

164 cameras are added, it takes less than 5min, I think adding cameras takes longest time. We might add cameras simultaneously instead of one by one. match_bow has less matches compared to the full brute force.