Announcements

Accepted additional feature ideas

In total, you need to implement three additional interesting features to obtain a 0.5 grade increase. Needless to say, we reserve the right to NOT award bonus points if we feel that these are implemented sloppily.

- Quantitatively compare own VO pipeline to existing V(I)O pipelines. See https://www.dropbox.com/s/6lc47rrkjfsdqab/reproducible_SLAM.pptx?dl=0 for good practices in evaluating trajectory estimations.
- Integrating Bundle Adjustment:
    - MUST NOT be simply taking the exercise code and applying it to your map
    - Should be continuous: While or between adding new frames, but not at the very end.
    - Should be applied to (the most recent) PART of the map (not the full map)
        - Consider different ways to define this “part” of the map and write which one you end up using and why.
        - Make sure the full map stays consistent even if you just optimize part of the map
    - Please check with us if you would like to integrate BA in any other way.
- Identifying known markers (for example, fiducial markers, April Tags or else...) in your own recorded dataset, and using these markers to correct for the scale drift periodically.

Questions

The poses predicted by my pipeline jitter a lot. Is there anything I can do about it?

Different issues in your pipeline might cause such problems.

- As a general advice, consider logging/plotting the output/status of the different algorithms involved to identify possible bugs.
- Check that you understand the impact of the parameters you feed to the different parts of your pipeline (for example, the RANSAC inlier thresholds are sensitive parameters, make sure you select them wisely and that the values that you choose are meaningful).
- In general, the 3D point cloud you use for tracking needs not be very dense. If you are maintaining multiple thousands of 3D points, consider lowering this number and favor fewer, higher quality 3D points. In a first step, make sure your pipeline gives reasonable results by just using the 3D points from bootstrapping without triangulating new points yet.

- Make sure that the features you track are evenly distributed in the image plane. For example, in KITTI, tracking features only in the center of the image (where everything is almost at infinity) will cause tracking problems.
- Consider refining each pose right after P3P/DLT using nonlinear optimization to minimize the reprojection error (see next question).

What do you mean by "nonlinear refinement" in the example videos linked in the problem statement?

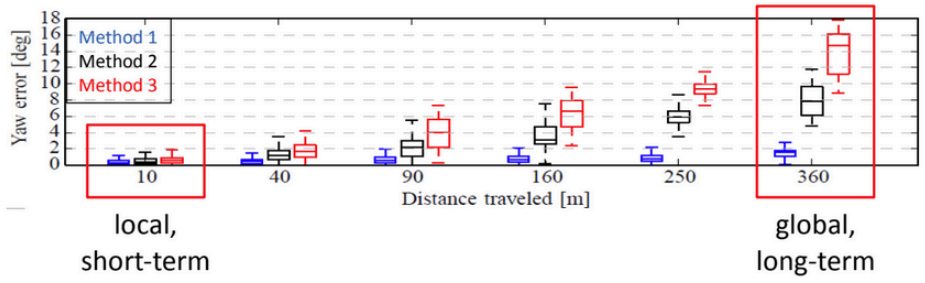
After computing a new camera pose from your set of 2D <-> 3D correspondences using P3P or PnP, it is possible to refine the pose a bit further by adding a step of nonlinear optimization to minimize the reprojection error of the 3D points on the predicted camera pose. The cost function and optimization method will be similar to Bundle Adjustment (see the exercise on Bundle Adjustment), but instead of refining jointly multiple camera poses and 3D landmarks, you will only refine the camera pose after obtaining a first guess from P3P or PnP. You can do that for every new camera pose. This tends to reduce the jitter in the predicted camera poses as well.

Does it count as an extra feature to compare the results of my pipeline *qualitatively* against one or more other state of the art pipelines (like SVO, ORB-SLAM or ROVIO)?

We do NOT accept qualitative evaluations (e.g. simply plotting the estimated trajectories and comparing them visually) as one of the bonus feature. However, a quantitative evaluation (see question below on how to do it) will be accepted as a bonus feature.

How can I proceed if I want to quantitatively measure the accuracy of my pipeline with respect to ground truth, even if my pipeline suffers from significant scale drift?

Obviously, if your pipeline suffers from significant scale drift, it will not be possible to align it globally to the ground truth. A possible approach to solve this problem is described in this document: https://www.dropbox.com/s/6lc47rrkjfsdqab/reproducible_SLAM.pptx?dl=0 (section "Relative Error"). The idea is to "slide" windows or varying lengths along the trajectories you wish to compare, align the corresponding sub-trajectories and compute the errors. This will allow you to produce an error plot such as this one:



Where the scale drift would appear clearly as large errors for large distances travelled.

Should I triangulate a fixed number of landmarks, or is better to change it dynamically?

It is up to you to decide that based on empirical evidence.

Consider limiting the global number of 3D landmarks you use (see question “The poses predicted by my pipeline jitter a lot”).

Can I use the function bundleAdjustment() from Matlab and still get the bonus points?

Since bundle adjustment has been implemented in exercise 8, it is OK to use the *bundleAdjustment* function from Matlab.
However, as explained above (in the list of accepted features), it is not enough to run bundle adjustment once at the end of the trajectory. We rather require you to implement some form of sliding window bundle adjustment, where the poses and 3D landmarks are periodically refined using a set of past (key)frames.

Clarification about the triangulation threshold angle:

The threshold angle alpha is there so you can decide to triangulate a landmark as soon as the angle between the bearing vectors of the first and most recent observations **exceed** it. This allows you to simply decide when to triangulate new landmarks without e.g. needing to keep track of how many landmarks have been lost in tracking (which is a valid alternative approach to decide when to triangulate new features).

A few points are triangulated behind the camera, can we just remove them because they are behind the camera?

Yes, no need for a more fancy, principled approach.

Can we implement loop detection / loop closing as an additional feature?

You are welcome to do so. Note that implementing loop detection and reflect it in the bundle adjustment is quite difficult and will require a lot of work.
Attempt to do it only when you have a pipeline that works well and you have some spare time.

The knowledge of which exercises is needed to be best prepared for the project?

1, 3, 4 (detection, not matching), 6, 7, 9 (KLT). 8 (BA) could be inspirational for bonus features.

Do we need to declare the use of the reference exercise solutions?

No, you can just copy them and don’t need to mention it. However, we would recommend you to prefer the equivalent Matlab functions (for example, *estimateFundamentalMatrix, detectHarrisFeatures,* etc.) over the reference solutions when they are available.

Instead of doing 8pt RANSAC for bootstrap, could I also use an algorithm which exploits the restricted camera motion (plane, negligible roll and pitch)?

Yes.

What do you mean “scale drift”?

As you know, monocular visual odometry has scale ambiguity: Purely from the landmark observations, you can’t really tell how big the scene is. The way we work with this is that we arbitrarily assign a scale at the beginning of the VO. However, this scale estimate might drift as you progress, i.e. far apart scenes that should have the same size, will have different size estimates.