Bundle Adjustment

The aim of this project is to find camera calibrations and 3d point position describing a scene given observations of the 3d points in each camera.

We have N cameras and M points. We have the following information:

- Rough initial estimates of each cameras calibration.
- Rough initial estimates of each points 3d position.
- 2D observations of points in each camera. That is: for each camera a list of `(point_id, point_projected_into_camera)` tuples.

The objective is to find camera calibration (extrinsics and intrinsics for all N cameras) and 3d point positions such that, for every observation, the 3d point projects to the correct 2d point.

With correct calibrations and 3d points the projected points should match exactly with the observations (the dateset has been generated synthetically).

We have provided you with three files:

1. `projection-check.json`: This file contains the calibration and observations of one camera and can be used to make sure that your camera projection model is correct.
2. `noisy-calibrations-and-world-points.json`: This file contains the actual calibrations and 3D points that need adjusting. The 2D observations are accurate but we have added noise to the 3D world points and the calibrations (rotation, translation and focal length).

3. `noisy-calibrations-and-accurate-world-points.json`: This file has accurate 2D to 3D correspondences with an initial noisy calibration. To be used only with the alternative route described below.

File Structure

```
{
  "cameras": {
    "{cameraId}": {
      "calib": {
        "extrinsics": [], // 3x4 row major OpenCV style extrinsics matrix.
        "intrinsics": [], // 3x3 row major OpenCV style intrinsics matrix.
        "size": []        // [width, height] of image.
      },
      "observations": {
        "{pointId}": [], // [x, y] in image co-ordinates.
        ...
      }
    },
    ...
  },
  "worldPoints": {
    "{pointId}": [], // [x, y, z] in world co-ordinates.
    ...
  }
}
```

Camera Model

The camera model is an opencv style pinhole projected camera. There are no distortion coefficients. The aspect ratio is 1 that is: `(focalLengthX = focalLengthY)`. As mentioned above the provided intrinsic values are correct in `noisy-calibrations-and-world-points.json` except for focal length.

Submission

1. Your optimised calibrations and world points in the same file format as `noisy-calibrations-and-world-points.json` in a file called `answer.json`.
2. Your source code.
3. A brief note about what you have done!

Guidance

We will assess your submission factoring in how much third party library code you have used. For two submissions implementing the same features less library code would be preferred. For many candidates using an existing non-linear least squares library would be appropriate.

Extensions/Alternatives

We assign this project to a range of candidates. There are adjustments to make the task easier or harder. Use your judgement to pick a route that works with the time/skills you have.

Extensions

1. Don't use the 3D world points! You can get initial guesses using triangulation.
2. Test your solution with noise/outliers in the 2d points.
3. Use less framework/library code to demonstrate certain skills.
4. Some sort of visualisation.

Alternative route

An alternative task which might be easier to start on is to: refine approximate calibrations of the cameras cameras to the fixed ground truth 3d points (a straightforward calibration refinement with 3d-2d point correspondences) using `noisy-calibration-with-accurate-points.json`.

You could then move onto bundle adjustment or take on some of these features to show off your skills:

1. Try to solve this problem without using the initial calibrations at all - just the correspondences.
2. Make the solution robust to noise and outliers in the 2d points.

Notes

- `[0, 0]` is the top left of the image and `[width, height]` is the bottom right of the image.
- Feel free to use existing frameworks/tooling but you will get bonus points for doing things from scratch.
- Don't spend more time on the assignment than you are comfortable with!