Description

Depth estimation is to estimate the distances of objects in a scene to the observer. There are multiple ways to accomplish the task. Some popular approaches include the time-of-flight (TOF) devices, structured light cameras and multiple-view geometry, among which the approach of multiple-view geometry is of the lowest cost. In multiple-view geometry, we care about how to estimate the depth of scenes with ordinary cameras. In this problem you will need to gradually design a pipeline to estimate the depth with cameras. The attached files will be needed in the programming problems.

(★ means slightly difficult. ★★ means hard. ★★★ refers to optional bonus problems.)

Camera Basics

1. What are the *intrinsics* and *extrinsics* of a camera? (Learn about camera model in `https://en.wikipedia.org/wiki/Camera_resectioning`) What is the camera matrix? Write down your answers.

2. (Camera Imaging) Given an ordinary camera with focal length f_x and f_y, optical center (c_x, c_y), the transform between the 3D world coordinates and the 3D camera coordinates is $(R|t)$, can you transform a 3D point $\mathbf{X} = (X_1, X_2, X_3)$ onto the image plane? (Consider only pinhole camera model. 3D camera coordinate position: front is +Z, left is +X, up is +Y. Image plane: right is +x, down is +y). Write down your answers in matrix form.

3. Given a 2D image point (u, v), what shape does it correspond to in the 3D camera coordinate? Can you derive its equation?

Now continue to read the camera calibration [1]section in the OpenCV document. (Cease reading when you reach the API descriptions of the function *calibrateCamera*.) The following problems will be based on the symbols and definitions acquired from this webpage. Look out for the coordinate directions.

4. (Distortion) Describe the distortions of the cameras. Given a 2D image point (u, v), can you find its coordinate before distortion? (For simplicity, consider only the distortion model with 4 distortion coefficients, e.g., k_1, k_2, p_1, p_2)

5. (Calibration) Describe what the camera calibration does.

6. (★ Programming) Provided a series of images taken by the camera, can you calibrate the camera with OpenCV functions? (use the images in *left.zip*. Read more about the APIs in camera calibration [1] section)

[1] opencv calibration document

7. (★ Programming]) Undistort the images with the calibration results you computed. What functions of OpenCV do you use?

8. (★★★ Programming) Learn about Zhang's method [1] for camera calibration. Can you implement Zhang's method? Report the calibration result with your implemented approach. Compare with the result from Problem 6.

Till now you have been skilled with the tricks of a single camera. Can you use a single camera to estimate the depth of the pixels (the distance of their corresponding 3D points to the image plane)? If not, how will you manage to do that? Explain your reasons. (Hint: Recall Problem 3.)

Binocular Basics

Binocular stereo refers to a stereo system consisting of two cameras. You can locate the 3D coordinates of a point providing its projections in two cameras from different views. This is called the binocular depth estimation. From now on, you will learn how two cameras can be used to estimate the depth of a point.

For simplicity, consider the two camera with intrinsics matrix M_l and M_r and temporarily ignore the distortions. The cameras will be distinguished as left and right. Let the 3D world coordinate be aligned with the 3D camera coordinate of the left camera. Denote the transform from the left camera to the right camera as $(R|t)$. $M_l = (f_x, 0, c_x; 0, f_y, c_y; 0, 0, 1)$ and $M_r = (f'_x, 0, c'_x; 0, f'_y, c'_y; 0, 0, 1)$.

9. (Projection) Given a point with 3D world coordinate $\mathbf{X} = (X_1, X_2, X_3)$. Write down its projection in left and right camera planes.

10. (★★ Epipolar Line) Given a pixel in the left image as $x_l = (u, v)$, its corresponding 3D point will also project onto the right image plane. Without knowing its precise 3D coordinate, the probable projection point on the right image will have to lie on a line (This is called the epipolar constraint). This line is called the epipolar line of (u, v). Can you derive the line's equation? (in the right image's pixel coordinate) Draw a figure to help you derive.

11. (Fundamental Matrix) Generally, given a fixed two camera system. There is a matrix F that for any 3D point, its projections in two images x_l and x_r will satifsy $x_l^T F x_r = 0$. F is called the fundamental matrix. Can you derive the matrix?

12. (★★ Stereo Calibration) We've already learnt how to calibrate a single camera. For a binocular camera system, calibration does more than calibrate each of them. The transform $(R|t)$ between the left and the right cameras also needs calibration. To know more about stereo calibration, you may refer to `http://blog.csdn.net/xuelabizp/article/details/50417914`. Now please use OpenCV to calibrate the binocular cameras with images in *left.zip* and *right.zip* respectively. Report the

results. (Make sure you understand the geometrical meanings of the parameters)

Here is a tutorial [2] from the University of Illinois. Before continuing, **it's highly recommended that you read the tutorial from page 1 to page 34.** We've learnt from Problem 10 that the epipolar constraint exists for a binocular camera system. To decide a left-image pixel's real 3D world coordinate, we must decide which right-image pixel on the epipolar line matches the left-image one. The search is a one-dimensional search. We will see in following sections that the search can be made easier if we rectify the images as if the cameras are placed with parallel optical axes.

13. Given left-image pixel p_l, write down its epipolar line. Pick an arbitrary point on the epipolar line as the projection of the 3D point. Can you derive the 3D coordinate of the point?

14. (★★ Rectification) The pose of the two cameras are usually arbitrary. It is usually necessary to make the two cameras parallel so that the epipolar line will become horizontal. This is achieved by adding a Rotation transformation to one of the cameras. (See `http://blog.csdn.net/gdut2015go/article/details/48391949`. You may also refer to the tutorial [3] for more details). Now use OpenCV to rectify the left and the right images with the calibration results obtained from Problem 13. Display the images and check with your eyes to see if they are really rectified. (A pair of the images is enough. Remember to undistort the images)

15. Once the images are rectified, the epipolar lines will become parallel to the image axes. For such a binocular camera system, the transformation between them will be simplified to $(I|t)$. The rotation matrix will become unit matrix since their axes are parallel. The translation matrix t will become $(b, 0, 0)$ with the coordinates defined in the OpenCV document [1]. Can you derive the *baseline* b from the results in Problem 14?

16. (Depth-Disparity) For a calibrated binocular camera system, the epipolar lines can be made parallel with rectification. For a pixel on the left image with coordinate (x_l, y), its matching point (the projection of the 3D point on the right image) must have coordinate of the form $(x_l - d, y)$. $d = x_l - x_r$ is called the pixel's disparity. Can you derive the 3D point's coordinate given baseline b and the camera matrices? If the camera has indentical vertical and horizontal focal lengths, can the depth z (Z coordinate value) of a pixel be written as $z = bf/d$? Write down your derivation.

Stereo Matching

From the last problem, we can find a way to estimate the depth of pixels with a binocular camera system. The calibration can give us the baseline b and all

[2] Epipolar tutorial
[3] Rectification tutorial

the camera intrinsics and extrinsics. We can undistort the images, then rectify them to have a simple relation between the disparity and depth as derived in Problem 16. What remains is to compute the disparity of each pixel. This is a one-dimensional search along the horizontal epipolar line for each pixel. The task is called *stereo matching overview*.

Stereo matching is a classical task in computer vision and has a long history. You may refer to the simple guide [4] for a quick overview. A lot of approaches have been developed during the past decades. A classical algorithm named SGM (semi-global matching) [2] is the most widely used approach and also stands as the baseline for many radical new methods. OpenCV has an implementation of a variation of this algorithm.

17. (★ SGBM) Can you use OpenCV to compute the disparity maps for the images you used for stereo calibration? Visualize several disparity results and check with your eyes to see if the results are reasonable.

The KITTI vision benchmark suite [5] holds a benchmark for stereo matching. You can see that most recently developed approaches are deep learning based. The deep learning based approaches are usually supervised.

18. (★ Unsupervised) Do you think that supervised deep learning approaches can be easily used for stereo matching? Why or why not? Recently unsupervised deep learning based approaches have been proposed. Can you find out these approaches from the evaluation page?

19. Find out the most efficient deep learning based approach and compare it with SGM. In real time scenarios like autonomous driving, which approach will you prefer? Explain your reason.

20. (★★★) Can you implement SGM with GPU (CUDA) acceleration? Test your implementation on KITTI data. Report your results (speed and error).

21. (★★★) We have been thinking about binocular depth estimation. Now that we have deep learning approaches, can you come out an approach for single image depth estimation? Does your approach need any premises?

22. (★★★) Try more stereo matching approaches, what common weaknesses have you discovered in all the approaches?

23. (★★★) Can you further accelerate one of the deep learning based approaches so that it can work in real-time ($\geq$30fps for KITTI sized images)?

[4] stereo matching
[5] KITTI stereo evaluation

References

[1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[2] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 807–814.