

Experiment 5: Stereo

Adrian F. Clark

This experiment explores the use of computational stereo to estimate distance, with its focus being on how accurately distance can be measured. Using a 3D model of a head, computer graphics software has been used to generate the pictures that would be seen by a pair of cameras at a known distance — this is similar to the approach commonly used in the movie industry when generating CGI characters. With information you know from the graphics software and measurements you make from the imagery, you can calculate by hand the distance of the model from the cameras.

OpenCV also provides functionality to calibrate cameras, so you can perform camera calibration and hence calculate the distance using the calibration data. But does that agree with the hand-calculated results and the ground truth...and if not, why not?

Contents

1	Introduction	2
2	Estimating the focal length	2
3	Camera calibration to determine the focal length	2
4	Distance calculation from calibration data	3

1 Introduction

You have been approached by a company, Wivenhoe Intelligent Systems Engineering (WISE) because of the extensive knowledge of computer vision acquired through the *Computer Vision* module. WISE is working on a stereo system for capturing faces in 3D. To aid their development work, they are using a modified version of the Candide face model, which they have “imaged” from a model of their stereo rig using the **POV-ray ray-tracer**. A ray-tracer is a program that takes a description of the components of a 3D scene, including lighting, camera and appearance, and calculates how it should appear. POV-ray is particularly attractive because this description is a kind of program.

WISE’s virtual stereo rig has identical cameras arranged so that their optical axes are precisely parallel — one of the advantages of working in simulation is that this can be guaranteed. This means that it should be possible to use the equation described in the lecture notes on stereo:

$$Z = \frac{fB}{D}$$

where Z is the distance to the object, f the focal length, B the baseline and D the disparity (parallax) of a feature between the left and right images. They have supplied you with a **zip-file containing programs and imagery**.

2 Estimating the focal length

The virtual cameras are at (x, y, z) locations $(\pm 75, 300, 800)$, which means that $B = 150$ mm. Vertex 5 of Candide, the tip of her nose, is at location $(0, 289, 280)$; hence, the distance from the cameras to it, Z , is approximately $800 - 280 = 520$ mm (why only approximately?). Measure the locations of that point in the images from the **left** and **right** cameras: use `xv` to display each image and press the middle mouse button on the tip of her nose to read the position. Estimate your accuracy in reading the position and, using all this information, work out using a calculator the focal length and its uncertainty (“error”) as described in Chapter 9 of the lecture notes.

3 Camera calibration to determine the focal length

The calculation above gives a rough estimate of the camera’s focal length. However, it should be possible to do better using computer vision, a process known as *camera calibration* — and there are routines in OpenCV to perform this. They were designed for a person to move a calibration target of known size, which looks like a chess-board, around in the field of view; images are captured in a variety of positions and orientations, and the locations of the corners of the pattern are used to compute several camera parameters, including the focal length.

To this end, a series of images has been rendered using POV-ray with the same cameras as for Candide but with a model of a calibration target; these are in [the zip-file](#) along with the program you need to use. Make sure you understand what this calibration program does and then run it as follows:

```
python3 calibrate.py
```

It should output the computed focal length and an estimate of its error (uncertainty).

4 Distance calculation from calibration data

Finally, using the focal length output by the calibration program, calculate the distance to the tip of Candide's nose and its uncertainty. Is this more accurate than the rough-and-ready calculation in the first part of the experiment?