TP7: Stereo Vision and disparity map calculation

Loic Simon Chaohui Wang Panagiotis Koutsourakis

30 Mai 2007

1 Theory

A binocular stereo vision system is composed of two cameras both of which observe the same scene. We get thus two images from two different angles of view. It can be verified that for any 3D point, its projection in one image must lie in the plane formed by its projection in the other image and the optical centers of the two cameras. This is known as *epipolar constraint*.

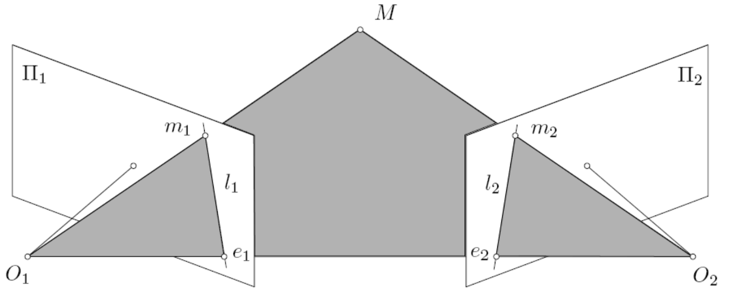

Figure 1: Epipolar geometry

Assuming that a 3D point M is observed by two cameras with optical centers O_1 and O_2, we get the projections m_1 and m_2 of this point in the two image planes (c.f.: Fig. 1). We define *epipolar plane* as the plane containing M, O_1 and O_2. Note that m_1, m_2 also belong to this plane.

Consider the case where m_2, O_1 and O_2 are given and we want to find the corresponding point of m_2 in the fisrt image, i.e.: m_1. The epipole plane is determined by m_2, O_1 and O_2(without knowing the position of M). Since m_1 must belong to this epipole plane and the image plane of the first camera, it must lie on the line l_1 which is the intersection of these two planes. We call the line l_1 the *epipolar line* associated with m_2. By symmetry, the m_2 must lie on the epipolar line l_2 associated with m_1. This epipolar constraint is used to look for the corresponding point in one image given a point in the other image. The search can be restricted to the epipolar line instead of the whole image.

In this lab, we consider the simplest case where the stereo images have been rectified, i.e.: the epipolar lines fall along the horizontal scan lines of the images. This can make the processing much easier. Note that in this case, if we have a point $m_1 = (u_1, v_1)$ in one image, the corresponding point $m_2 = (u_2, v_2)$ in the other image is in the same height as m_1, i.e.: $v_1 = v_2$. In this context, the *disparity* is defined as:

$$d = u_2 - u_1$$

We can obtain the depth information of a 3D point from the disparity since its depth is inversely proportional to the corresponding disparity.

The task of this lab is to calculate the *disparity map* of two given rectified images. For each 2D point in the first image, we need to find the corresponding 2D point in the second image and then calculate the disparity d. For the matching of two points, we will use two methods: sum square difference (SSD) and normalized cross-correlation (NCC) (c.f.: TP4).

2 Computation

We need to complete the function *computeDisparityMapWithSSD* and *computeDisparityMapWithNCC* which calculate the disparity map *disparityMap* respectively using SSD and NCC. They have the arguments as follows:

1. The two stereo images: *leftimage* and *rightimage*

2. The size of the window used for the matching: *sizeWindow*

3. The maximum of the disparity: *maxDisparity*. It will avoid scanning the whole search line.

4. The disparity map: *disparityMap*. It has the same size as those stereo images. We need to implement the calculation of this map.