

1 Object Features Detection

Today's exercise is focused on implementation of simple object features. This process will take some time and we will spend more time on this topic.

One of the possible ways in detection and classification of objects in images is using object features. Today's lecture will consist of the following steps:

1. thresholding the image
2. indexing objects
3. computing moments and features

In the end, our task is to label each object in the image (Fig. 1.1).

1.1 Thresholding the image

The first step is to separate objects in the image f from the background. In our example image, this task is not so hard, so we can apply the following thresholding criterion to each pixel in image f to construct a new image g

$$g(x, y) = \begin{cases} 1, & f(x, y) > t \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where t is a chosen threshold. There are many methods to select t , see the text for this course. However, in our case, we can set the threshold manually.

After this step, we have a binary image that indicates where are the objects and where is the background.

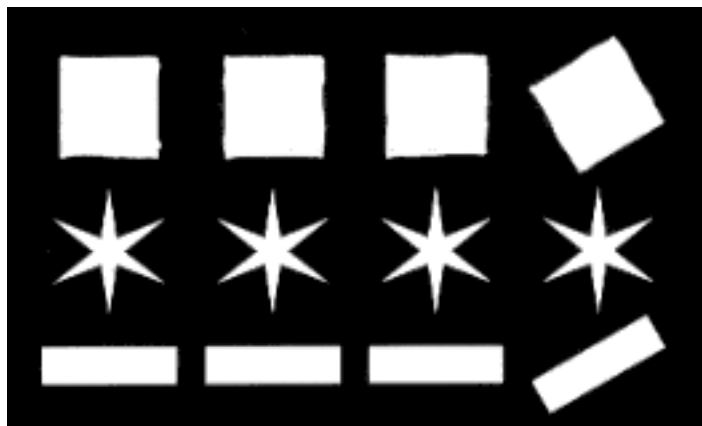


Figure 1: Input image.

1.2 Indexing objects

In order to compute moments and features of the upcoming classification, we need to separate detected objects from the previous step. This process is called indexing and its purpose is to distinguish between the objects in the binary image. The result of such process is shown in Fig. 1.2.

We can implement this process using a simple flood fill algorithm. In this algorithm, we go through all the pixels and if it belongs to the objects, we label this pixel with chosen constant and recursively flood fill four neighbours with this constant too. As one might expect, we label each object with different constant.

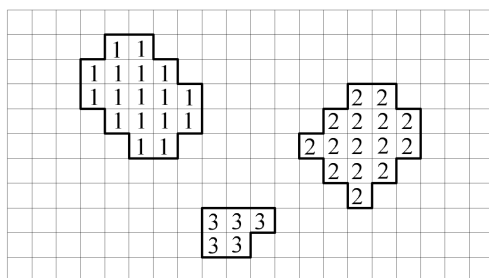


Figure 2: Indexed image.



Figure 3: Colorized indexed image for out training image. Each color represents different object index.

1.3 Computing moments and features

Moments are common features that are used in object classification since they are easy to compute.

The moment that is related to the coordinate system is computed as follows

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x, y), \quad (2)$$

(in our case, we do not have to multiply the coordinates by the corresponding image function, since it is almost the same, see Fig. 1.1).

Let us explain some important moments:

The **area** of an object can be computed as $m_{0,0}$.

The **center of the mass** of an object:

$$x_t = \frac{m_{1,0}}{m_{0,0}}, \quad y_t = \frac{m_{0,1}}{m_{0,0}} \quad (3)$$

The **perimeter** of an object may be computed using a simple algorithm that scans each pixel of the object and if the four neighbour of the pixel contains the same values as the central pixel than this pixel does not contribute to the perimeter, otherwise it does.