Jesús Castillo Benito
09/12/2024

C - Find centroid of the rectangle that is the farthest to the right

1) Input generation

To generate the input, I created a simple script that creates a random image each time, that fulfills the requirements.
It generates an image with white background and 7 rectangles that have a 50% chance to be bigger in size. Then they are also distributed randomly all over the image. Then some noise and smoothing is done to conclude.

The script is attached in the project folder with the name “generate_input.py”.

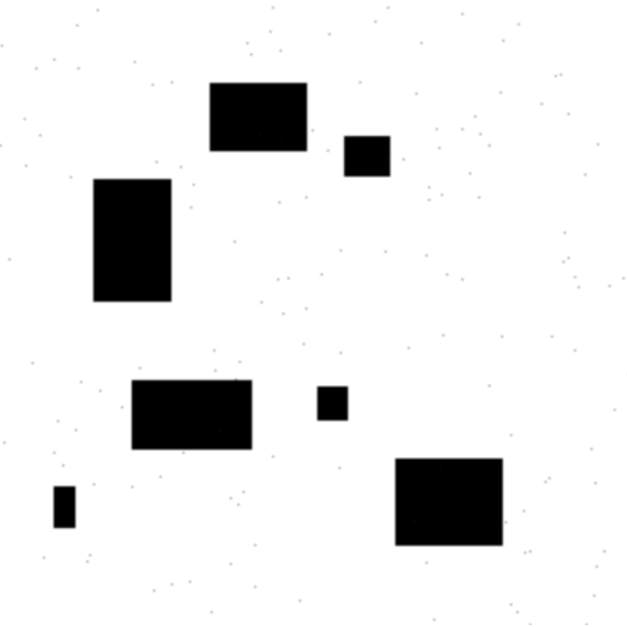
2) OpenCV implementation

For the implementation, I created another script called “solution.py”, that can also be found in the project folder.
This script follows these steps to accomplish the task:

1. **Preprocessing:** the image is loaded in grayscale and thresholding is applied to it to separate the rectangles from the background.
2. **Noise reduction:** to remove small noise after thresholding, a 3x3 kernel is used.
3. **Contour detection:** using cv2.findContours, we identify the boundaries of the rectangles of the image.
4. **Rectangle filtering and centroid calculation:** remove from the list of contours those that don’t match the characteristics of a rectangle, Then, for each remaining confirmed rectangle, calculate its centroid.
5. **Find the farthest centroid to the right:** compare “x” coordinates of all centroids and select the one that has the highest.
6. **Give and output:** draw the selected centroid in the rectangle of the original image (with a red circle) and output it.

This step by step list explains the approach that I followed to implement my solution to this project. Here you can see the results in action:

Input



Output