Region Labeling

Dr Frazer Noble

Introduction

In this presentation, I will describe:

- How to use OpenCV to label regions in an image.

Requirements

To follow along with this tutorial, you will need the following tools:

- Python 3.8.6.
- Visual Studio Code 1.53.1.

You will also need to install the following Python packages:

- OpenCV.
- NumPy.

It is assumed that you are using Windows; however, these instructions should be easily adapted to Linux.

Getting Started

Open Visual Studio Code. To open the app: Open the Start menu, type `Visual Studio Code`, and then select the app.

Open the Explorer tab. To display the tab: Left click `View > Explorer` or press `Ctrl` + `Shift` + `E`. This will display the Explorer tab.

Left click on the `Open Folder` button. This will display the Open Folder prompt. Browse to the following directory:

```
C:/Users/%USER%/Documents
```

Note: Replace `%USER%` with your own username. My username is fknoble; hence, the path is `C:/Users/fknoble/Documents`.

In `C:/Users/%USER%/Documents` create a new folder named `opencv_09`. To create a new folder: Right click in the Explorer tab, left click `New Folder`, and rename it.

In `C:/Users/%USER%/Documents/opencv_09` create a new file named `region.py`. To create a new file: Right click on `/opencv_09` in the Explorer tab, left click `New File`, and rename it. The file will open automatically.

`/opencv_09` should contain the following files and folders:

```
/opencv_09
    region.py
```

`region.py`

Type the following code into `region.py`:

```
import cv2 as cv
import numpy as np
```

OpenCV's Python module `cv2` is imported as `cv` and NumPy's Python module `numpy` is imported as `np`.

Type the following code into `region.py`:

```
def main():

    rows, cols = 480, 640
    img = np.zeros((rows, cols, 1), np.uint8)
```

This begins `main()`'s definition. `zeros()` creates a 480 x 640 x 1 array of 0's and assigns it to array `image_1`.

Type the following code into `region.py`:

```
    coordinates = [(int(random.random() * cols), int(random.random() * rows))
                   for _ in range(40)]

    for c in coordinates:

        cv.circle(img, c, 10, (255), -1)

    cv.imshow("img", img)
    cv.waitKey(1)
    cv.imwrite("data/img.PNG", img)
```

An array of 40, random (x,y) coordinates is assigned to array `coordinates`. For each coordinate, a filled circle is drawn on `img` at that location. `img` is then displayed in the `img` window and saved as `img.PNG` in `/data`.

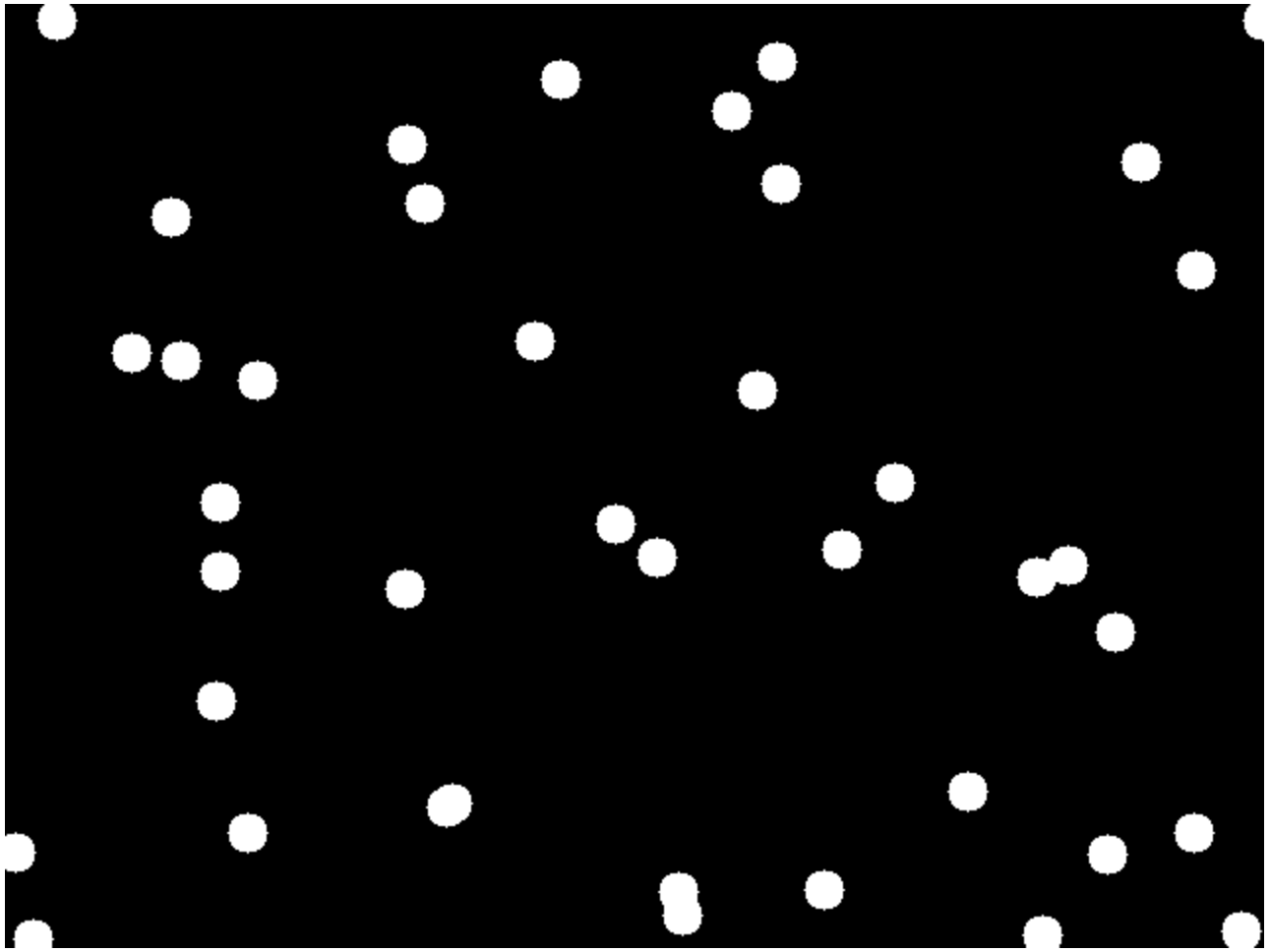

Figure: The `img` array.

Type the following code into `region.py`:

```
    contours, hierarchy = cv.findContours(
        img, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)
```

`findContours()` detects contours in `img`. If contours are found, each contour, and its hierarchy, is stored in `contours` and `hierarchy`.

Type the following code into `region.py`:

```
    regions = np.zeros((rows, cols, 3), np.uint8)

    for i, c in enumerate(contours):

        colour = (random.random() * 255, random.random()
                  * 255, random.random() * 255)

        cv.drawContours(regions, contours, i, colour, -1)
```

```
    cv.imshow("regions", regions)
    cv.waitKey(0)
    cv.imwrite("data/regions.PNG", regions)

    cv.destroyAllWindows()

    return 0
```

`zeros()` creates a 480 x 640 x 1 array of 0's and assigns it to `regions`. For each contour in `contours`, the contour is drawn on `regions` and filled with a random colour. `regions` is then displayed in the `regions` window and saved as `regions.PNG` in `/data`.

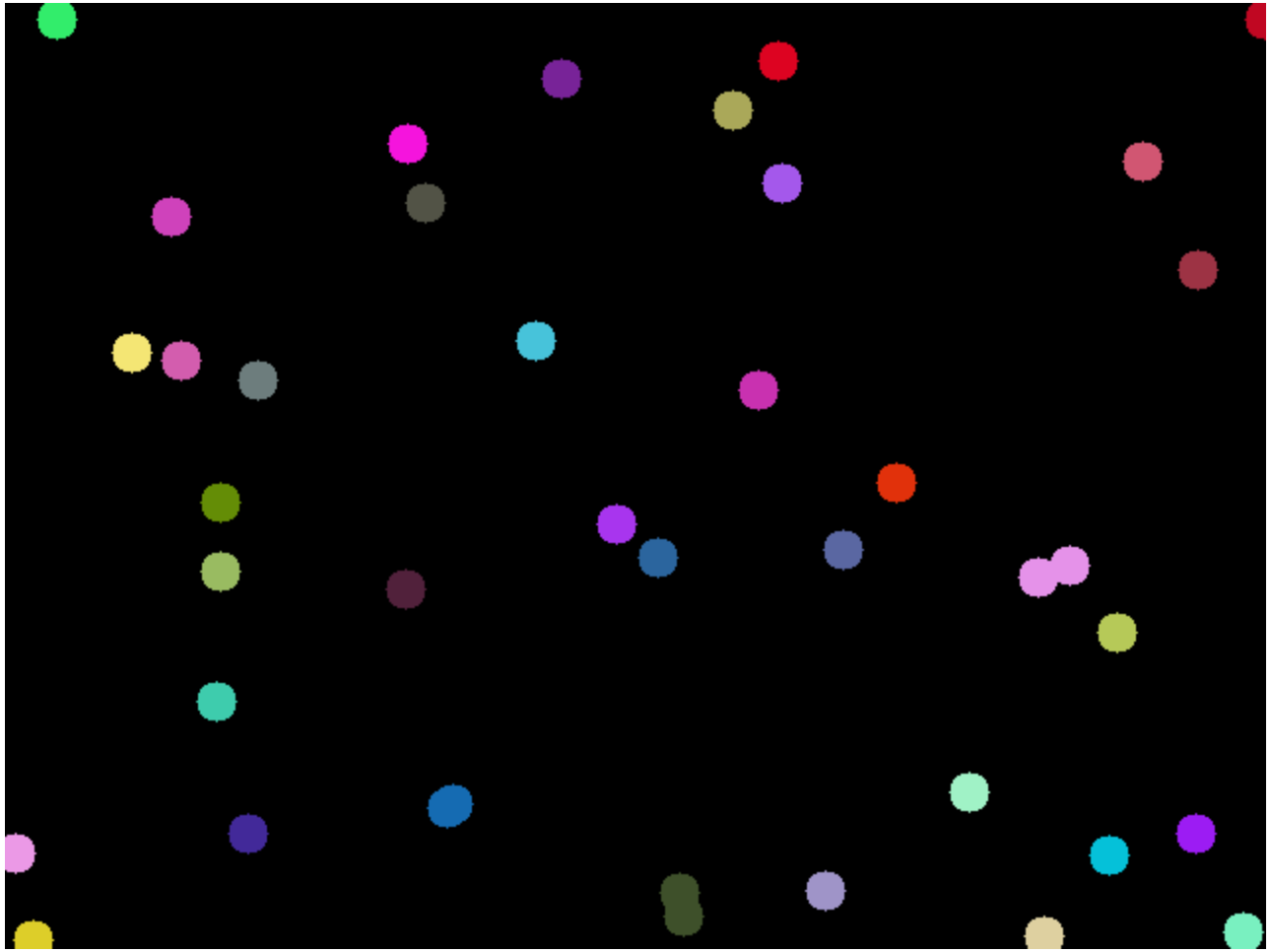

Figure: The `regions` array.

Type the following code into `region.py`:

```
if __name__ == '__main__':

    main()
```

`main()` will be called when the `region.py` is run.

Run `region.py`

Open a new terminal in Visual Studio Code. To open a new terminal: Left click `View > Terminal` or press `Ctrl`+`` ` ``.

Type the following commands into the terminal and then press `Enter` after each one:

```
cd ./opencv_09
python region.py
```

This will change the current directory to the `/opencv_09` sub-directory and then run `region.py`.

Press any key to close the windows and stop `region.py`.

Conclusion

In this presentation, I have described:

- How to use OpenCV to label regions in an image.

References

1. https://docs.opencv.org/.