Videos

Dr Frazer Noble

Introduction

In this presentation, I will describe:

- How to use OpenCV to get frames from a video source.

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_05`. To create a new folder: Right click in the Explorer tab, left click `New Folder`, and rename it.

In `C:/Users/%USER%/Documents/opencv_05` create a new file named `video.py`. To create a new file: Right click on `/opencv_05` in the Explorer tab, left click `New File`, and rename it. The file will open automatically.

`/opencv_05` should contain the following files and folders:

```
/opencv_05
    video.py
```

`video.py`

Type the following code into `video.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 `video.py`:

```
def main():

    camera = cv.VideoCapture(0)

    if not camera.isOpened():
        print('ERROR::CV::Could not open video device.')
        return 1
```

This begins `main()`'s definition. `VideoCapture(0)` creates a VideoCapture object using the default device and assigns it to `camera`. If the object is not open, a message is displayed and `main()` returns 1.

Type the following code into `video.py`:

```
    while True:

        ret, frame = camera.read()

        if ret is True:
            cv.imshow("frame", frame)

        i = cv.waitKey(30)

        if i == 27:
            break

    cv.destroyAllWindows()

    return 0
```

`camera`'s `read()` requests a frame from `camera`. If a frame is retrieved, it is saved in `frame` and `ret` is assigned `True`. If `ret` is `True`, `frame` is then displayed in the `frame` window. `waitKey()` waits for input. If `Esc` is pressed, the loop ends.

Figure: The `frame` array.

Type the following code into `video.py`:

```
if __name__ == '__main__':

    main()
```

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

Run `video.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_05
python video.py
```

This will change the current directory to the `/opencv_05` sub-directory and then run `video.py`.

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

Conclusion

In this presentation, I have described:

- How to use OpenCV to get frames from a video source.

References

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