CV2 Examples

January 30, 2017

0.1 videorecdisp

- Import relevant modules

```
In [ ]: import cv2
```

- Program to capture a video from the default camera(0) and display it live on the screen

```
In [ ]: cap = cv2.VideoCapture(0)

        while(True):
            #capture frame-by-frame
            retval, frame = cap.read()

            #Display the resulting frame
            cv2.imshow()

            if cv2.waitkey(1) & 0xFF == ord('q'):
                break
```

- When everything done, release the capture

```
In [ ]: cap.release()
        cv2.destroyAllWindows()
```

0.2 imagerecdisp

Program to save a photo from webcam(default camera). * Import cv2.

```
In [ ]: import cv2
```

- Program to capture an image from camera and display it live on the screen

```
In [ ]: cap = cv2.VideoCapture(0)
```

- Capture Frame

```
In [ ]: ret, frame = cap.read()
```

- Display the resulting frame

```
In [ ]: cv2.imshow('frame', frame)
        cv2.imwrite('pycolorphoto.jpg', frame)

        while(True):
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
```

- When everything done, release the capture

```
In [ ]: cap.release()
        cv2.destroyAllWindows()
```

0.3 pyimageshowpixel

Program to capture an image from a camera and display the pixel value on the screen. * Import cv2

```
In [ ]: import cv2

        cap = cv2.VideoCapture(0)
```

- Capture one frame

```
In [ ]: [ret, frame] = cap.read()

        print("image format: ", frame.shape)
        print("pixel 0,0: ",frame[0,0,:])
```

0.4 videorecdispRGB

Program to capture a video from a camera and display Original and R,G,B, compinents live on the screen. * Import cv2

```
In [ ]: import cv2

        cap = cv2.VideoCapture(0)
```

- Set up window frames for different color components

```
In [ ]: cv2.namedWindow('Original')
        cv2.namedWindow('B Komponente')
        cv2.namedWindow('G Komponente')
        cv2.namedWindow('R Komponente')
```

- Start recoring and disply it frame by frame.

```
In [ ]: while(True):
            # Capture frame-by-frame
            [ret, frame] = cap.read()

            # Display the resulting frame
            cv2.imshow('Original',frame)
            cv2.imshow('B Komponente',frame[:,:,0])
            cv2.imshow('G Komponente',frame[:,:,1])
            cv2.imshow('R Komponente',frame[:,:,2])

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
```

- When everything done, release the capture

```
In [ ]: cap.release()
        cv2.destroyAllWindows()
```

0.5 videorecprocy

Program to capture video through camera, frame-by-frame and the disply back its luminance and green component. * Import cv2 and numpy.

```
In [ ]: import numpy as np
        import cv2

        cap = cv2.VideoCapture(0)
```

- Set up different windows to display different characteristics.

```
In [ ]: cv2.namedWindow('Original')
        cv2.namedWindow('Luminanz Y')
        cv2.namedWindow('Zum Vergleich: Gruen Komponente')
```

- Capture fram-by-frame and display back windows with corresponding characteristics.

```
In [ ]: while(True):
            # Capture frame-by-frame
            [ret, frame] = cap.read()

            # Our operations on the frames come here
            #Berechnung der Luminanz-Komponente Y:
            # Y= 0.114*B+0.587*G+0.299*R :
            # /256 because the result is float values which imshow expects in range 0...1:
            Y=(0.114*frame[:,:,0]+0.587*frame[:,:,1]+0.299*frame[:,:,2])/256;
```

```
        #Vergleich mit Gruen Komponente:
        G=frame[:,:,1]

        # Display the resulting frame
        cv2.imshow('Original',frame)
        cv2.imshow('Luminanz Y',Y)
        cv2.imshow('Zum Vergleich: Gruen Komponente',G)
        #Ende durch Taste "q":
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

- When everything done, release the capture.

```
In [ ]: cap.release()
        cv2.destroyAllWindows()
```