hochschule mannheim

Fakultät für Informatik

Computer Vision (CVIS)

Einführung in openCV

Dr. Oliver Wasenmüller

Allgemeines

UPDATE
Gültig ab 24.10.2018

Übungsformat

Pflichtübung

➔ Voraussetzung zur Teilnahme an der Prüfung

Was bedeutet das?

- ~~Anwesenheitspflicht~~ ➔ Anwesenheit empfohlen aber nicht zwingend
- Aktive Teilnahme an der Übung empfohlen, nicht zwingend
- Sinnvolle Bearbeitung der Übungsaufgaben, zwei Möglichkeiten für Nachweis
  - Im Laufe der Übungsstunde in der Blatt ausgegeben wird
  - Am Anfang der darauffolgenden Übungsstunde

Allgemeines

UPDATE
Gültig ab 24.10.2018

Wie wird Übung bewertet?

Ampelsystem

- Grün: Gute Bearbeitung der Übungsaufgaben
- Gelb: Lösung der Übungsaufgaben weisen Schwächen auf
- Rot: Keine Nachweis erbracht

Wie werde ich zur Prüfung zugelassen?

- Maximal 2x rot
- Maximal 2x gelb

Ausnahmen?!?

Nur bei gutem Grund: Krankheit, einmalige parallele Veranstaltung der HS, usw.
(immer Nachweis erforderlich)

Pakete in Python importieren

Python bietet die Möglichkeit verschiedene Pakete/Bibliotheken zu importieren:

```
>>> import cv2
```

OpenCV

```
>>> import numpy as np
```

numPy

numPy

Einführung numPy

NumPy ist eine Programmbibliothek für die Programmiersprache Python, die eine einfache Handhabung von Vektoren, Matrizen oder generell großen mehrdimensionalen Arrays ermöglicht.

Neben den Datenstrukturen bietet NumPy auch effizient implementierte Funktionen für numerische Berechnungen an.

NumPy wird von openCV intensiv genutzt.

2006 als Nachfolger von Numeric erschienen.

Einführung numPy

NumPy muss immer erst importiert werden!

```
>>> import numpy as np
```

Erstellung eines Arrays:

```
a = np.array([2,3,4])
b = np.array([[1,2,3], [4,5,6]])
```

Einführung numPy

Eigenschaften von Arrays:

```
>>> b.shape
(2, 3)

>>> b.ndim
2

>>> b.dtype.name
'int32,

>>> type(b)
<type 'numpy.ndarray'>
```

Einführung numPy

Kreierung von Arrays:

```
>>> a = np.array([2,3,4])
>>> a
array([2, 3, 4])
>>> a.dtype
dtype('int32')

>>> b = np.array([1.2, 3.5, 5.1])
>>> b.dtype
dtype('float64')
```

Einführung numPy

Kreierung von Arrays:

```
>>> np.zeros( (3,4) )
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])

>>> np.ones( (3,4) )
 array([[ 1., 1., 1., 1.],
        [ 1., 1., 1., 1.],
        [ 1., 1., 1., 1.]])
```

Einführung numPy

Kreierung von Arrays:

```
>>> np.eye(3)
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])

>>> a = np.ones((2,3), dtype=int)
>>> a *= 3
>>> a
array([[3, 3, 3],
       [3, 3, 3]])
```

Einführung numPy

Arrays umformen

```
>>> a = np.arange(6)
>>> print(a)
[0 1 2 3 4 5]

>>> b = np.arange(12).reshape(4,3)
>>> print(b)
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

Einführung numPy

Basisoperatoren

```
>>> a = np.array( [20,30,40,50] )
>>> b = np.arange( 4 )

>>> a-b
array([20, 29, 38, 47])

>>> b*2
array([0, 2, 4, 6])

>>> a<35
array([ True, True, False, False])
```

Einführung numPy

Basisoperatoren

```
>>> A = np.array( [[1,1],[0,1]] )
>>> B = np.array( [[2,0],[3,4]] )

>>> A * B
array([[2, 0],      # elementwise product
       [0, 4]])

>>> A.dot(B)
array([[5, 4],      # another matrix product
       [3, 4]])
```

Einführung numPy

Zusammensetzen von Arrays

```
>>> R = np.eye(3)
>>> t = np.zeros((1,3))
>>> R
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])
>>> t
array([[ 0.],
       [ 0.],
       [ 0.]])
```

Einführung numPy

Zusammensetzen von Arrays

```
>>> np.hstack((R,t))
array([[ 1.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.],
       [ 0.,  0.,  1.,  0.]])

>>> np.vstack((R,t))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```

Dimensionen müssen passen!

openCV

Einführung openCV

openCV:

- bekannteste Bibliothek im Bereich Computer Vision
- erster Release im Jahr 2000
- verfügbar in C++ und Python

- open-source
- Frei verfügbar für Forschung und Kommerzielle Nutzung

Import der Bibliothek durch

```
>>> import cv2
```

Einführung openCV

Einlesen von Bildern

```
>>> img = cv2.imread(„test.jpg“, 1)
```

Zweites Argument spezifiziert den Typ des eingelesenes Bildes

`cv2.IMREAD_COLOR` : Läd ein Farbbild ➔ 1

`cv2.IMREAD_GRAYSCALE` : Läd ein Graustufenbild ➔ 0

Einführung openCV

Anzeigen von Bildern

```
cv2.imshow(,Bildanzeige',img)

cv2.waitKey(0)
```

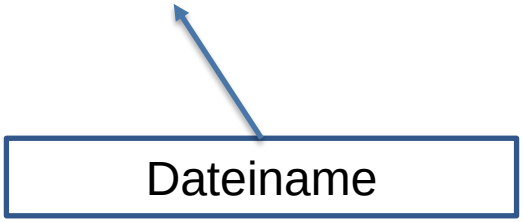
Name des Fensters

Warten auf eine beliebige Tastatureingabe

Einführung openCV

Schreiben von Bildern

```
cv2.imwrite(„test_new.png“,img)
```



Einführung openCV

Wie werden Bilder/Pixel in openCV repräsentiert?

Farbbild ist drei-dimensionales numPy array

```
>>> img.shape
(640,480,3)
```

```
>>>img.dtype
dtype(‚uint8‘)
```

Vorsicht:

openCV speichert Farben als **BGR**, nicht RGB

Einführung openCV

Zugriff auf Pixel

```
>>> px = img[100,100]
>>> print px
[157 166 200]

# Zugriff auf blauen Pixel
>>> blue = img[100,100,0]
>>> print blue
157
```

Einführung openCV

Auswahl von Bildbereichen

Mit Hilfe von : können Bereich ausgewählt werden:

```
>>> region = img[100:200, 300:350]
>>> region.shape
(100, 50, 3)
```

Mit Hilfe von : können auch alle Bereiche ausgewählt werden:

```
>>> blaue_pixel = img[:,:,0]
```

Einführung openCV

Aufteilen und Zusammensetzen der Farbkanäle

```
>>> b,g,r = cv2.split(img)
>>> img = cv2.merge((b,g,r))
```

Vielen Dank!

Nächste Übung: Projektion