

OpenCV

Andrej Lúčny

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

http://dai.fmph.uniba.sk/w/Andrej_Lucny

www.agentspace.org/opencv

Prednášajúci



RNDr. Andrej Lúčný, PhD.

- zaoberám sa umelou inteligenciou
- som spoluzakladateľ po celom svete operujúcej firmy dodávajúcej hardvér a softvér monitorovacích systémov
 - aktuálne vyvíjam aplikácie počítačového videnia v jej dcérskej spoločnosti a to v OpenCV
- som spoluzakladateľ občianskeho združenia robotika.sk
 - som bývalý porotca ACM Scholastic Programming Contest
- som porotca súťaže mobilných robotov ISTROBOT

www.agentspace.org/andy

CERTIFICATE OF COMPLETION

This is to certify that

Andrej Lucny

has successfully completed the online course

Computer Vision for Faces

October 7, 2018

12018344

Certificate No.
LearnOpenCV.com



Dr. Satya Mallick
Course Director

Hodnotenie

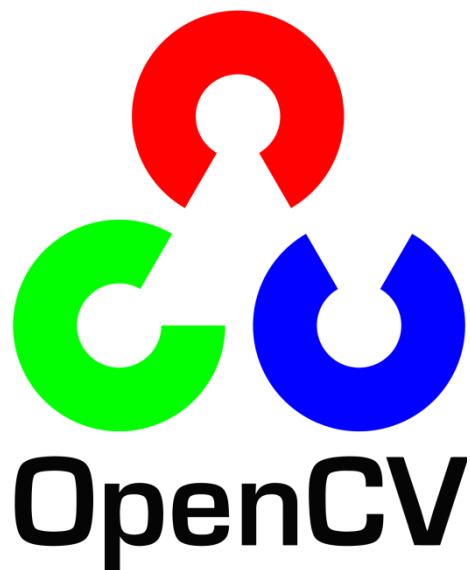
Semester 50%

Podmienka
postupu na
skúšku: odovzdať
správne
vyriešených 5
domácich úloh z
11 možných

Skúška 50%

4 úlohy rôznej
náročnosti

4 správne A
3 správne B
2 správne C
1 správne D
správna idea E
nič Fx



ver 4.4



- Open source knižnica na počítačové videnie
- Intel, Willow Garage, Itseez (Gary Bradski → Satya Malik)
- BSD Licencia
- 2500 rôznych algoritmov
- **C++**, **Python**, Java
- Windows, Linux, Android
- Je to fakt rýchle spoľahlivé
- Podpora CUDA a OpenCL
- Vhodné kombinovať s Fiji

www.opencv.org

Súčasťi OpenCV

aruco	cudaoptflow	hfs	opencv	saliency
bgsegm	cudastereo	highgui	<i>opencv_contrib</i>	shape
bioinspired	cudawarping	img_hash		stereo
calib3d	cudev	imgcodecs		stitching
ccalib	datasets	imgproc		structured_light
core	dnn	line_descriptor		superres
cudaarithm	dnn_objdetect	ml		surface_matching
cudabgsegm	dpm	objdetect		text
cudacodec	face	optflow		tracking
cudafeatures2d	features2d	ovis		video
cudafilters	flann	phase_unwrapping		videoio
cudaimgproc	fuzzy	photo		videostab
cudalegacy	gapi	plot		viz
cudaobjdetect	hdf	reg		xfeatures2d
		rgbd		ximgproc
				xobjdetect
				xphoto

Inštalácia

Python

```
pip install numpy
```

```
pip install opencv-contrib-python
```

C++

Z <https://opencv.org/releases/> stiahnuť binárky a rozbaľiť alebo zdrojáky a cez cmake a VS zbuildovať

v terminálke rozbaľujeme opencv tak, že máme E:\opencv\build
doma napr. c:\opencv\build

- Visual Studio 17 (C++ 15)
distributable pre VS19 - VS15 (tj. C++ 15 a 14)
https://aka.ms/vs/16/release/vc_redist.x64.exe

ak máme grafické karty NVIDIA (v opačnom prípade toto vyhodí hlášku, že je nepodporovaný driver)
inštalujeme CUDA:

- Updatneme driver grafickej karty
- Nainstalujeme CUDA 10.2 z developer.nvidia.com
- + dokopirovať do `c:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\ CuDNN`
- Možeme instalovať aj codec-y pre video
- Zbuildujeme s podporou CUDA (treba k tomu najprv zbuildovať viacero knižníc) alebo použijeme binárky www.agent-space.org/opencv/opencv430.zip

Obraz ... Numpy array

- `np.uint8`
- `np.int8`
- `np.uint16`
- `np.int16`
- `np.int32`
- `np.float32`
- `np.float64`

	C1	C2	C3	C4
<code>CV_8U</code>	0	8	16	24
<code>CV_8S</code>	1	9	17	25
<code>CV_16U</code>	2	10	18	26
<code>CV_16S</code>	3	11	19	27
<code>CV_32S</code>	4	12	20	28
<code>CV_32F</code>	5	13	21	29
<code>CV_64F</code>	6	14	22	30

Povaha objektu reprezentujúceho obraz

- numpy array je viacrozmerné pole
- jeho dáta môžu byť zdieľané viacerými premennými
- = nekopíruje dáta
- Dáta kopíruje `np.copy()`
- Do funkcií sa odovzdávajú hodnotou a preberajú návratovou hodnotou `imgA, imgB=fnc(imgC)`

Obraz ... cv::Mat

- $\text{Vec}\langle\text{uchar},3\rangle = \text{Vec3b}$

- uchar
- char
- ushort
- short
- int
- float
- double

	C1	C2	C3	C4
CV_8U	0	8	16	24
CV_8S	1	9	17	25
CV_16U	2	10	18	26
CV_16S	3	11	19	27
CV_32S	4	12	20	28
CV_32F	5	13	21	29
CV_64F	6	14	22	30

Povaha objektu reprezentujúceho obraz

- `cv::Mat` je hlavička ukazujúca na dáta
- Tieto dáta môžu byť zdieľané viacerými `cv::Mat`
- = kopíruje hlavičky nie dáta
- Dáta kopíruje `copyTo()` alebo `clone()`
- Odporúčanie: `cv::Mat` odovzdávajte vždy referenciou `prototyp(cv::Mat &image)`
- Odporúčanie, zaužívané v OpenCV: ten istý `image` sa má dať súčasne odovzdať ako vstup i výstup, na čo musí vnútorná implementácia pamätať
- `InputArray`, `OutputArray`

Ako načítat', zobrazit', uložit'

```
import numpy as np
import cv2
img = cv2.imread("../lena.jpg",cv2.IMREAD_COLOR)
img.shape
cv2.imshow("color image",img)
cv2.waitKey(0)
cv2.imwrite("lena.png",img)
cv2.destroyAllWindows()
img
```

Pozor! imshow len povie čo zobrazit', zobrazuje to waitKey

Ako načítat', zobrazit', uložit'

```
#include "opencv2/opencv.hpp"

int main (int argc, char *argv[])
{
    cv::Mat img = cv::imread("../lena.jpg", cv::IMREAD_COLOR);
    if (img.empty()) return -1;
    cv::imshow("color image",img);
    cv::waitKey(0);
    cv::imwrite("lena.png",img);
    cv::destroyAllWindows();
    return 0;
}
```

Pozor! imshow len povie čo zobrazit', zobrazuje to waitKey

Region of Interest (ROI)

Python

```
img[0:4, 5:8]
```

C++

```
img(cv::Rect(5, 0, 3, 4))
```

Vytvára novú hlavičku ale zdieľa dáta
Cez tieto dáta sa už nedá jazdiť pointerom inak než
každý riadok zvlášť. Funkcie implementované so
smerníkmi a kompatibilné s ROI musia spracúvať
každý riadok zvlášť (a v knižnici sa to tak robí).

Manipulácie s obrazom

S obrazmi sa dá manipulovať až piatimi spôsobmi:

1. Vždy, keď je to možné, používajte funkcie OpenCV
2. Alebo implementujte svoje funkcie len pomocou týchto funkcií
3. Dá sa pristupovať aj k pixelom cez funkcie OpenCV
4. ... cez funkcie std
5. ... cez smerníky

Farebný obraz ... [0,0,0] – [255,255,255]

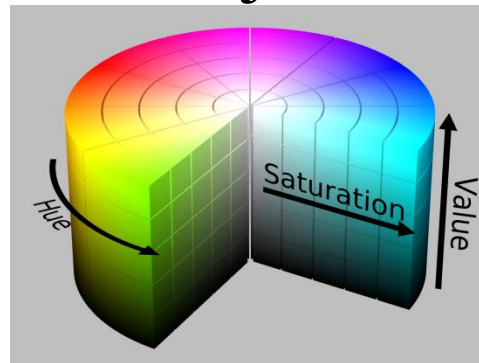


Čiernobiely obraz ... 0-255



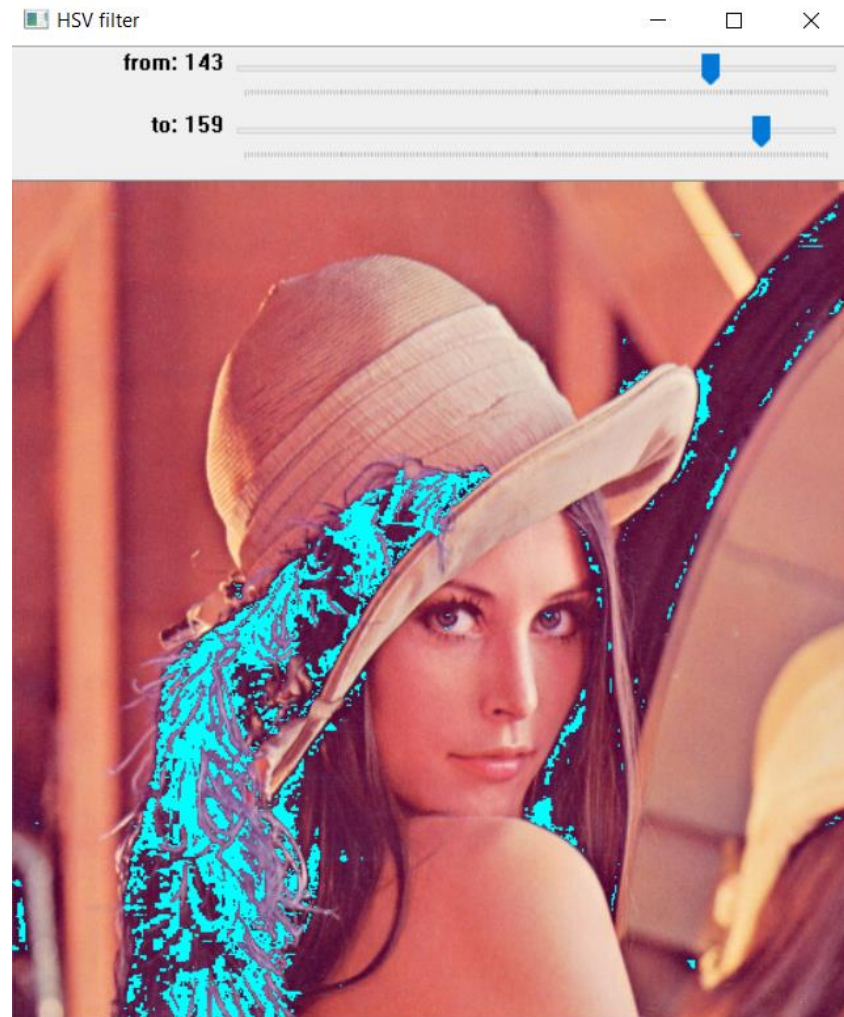
Farebné modely

- Farba je bežne implementovaná BGR modelom, $\text{Scalar}(255, 0, 0)$ je modrá, $\text{Scalar}(0, 0, 255)$ červená, $\text{Scalar}(255, 0, 255)$ fialová, $\text{Scalar}(127, 127, 127)$ sivá
- OpenCV poskytuje prevod do viacerých iných farebných modelov
- Najužitočnejší je HSV, v ktorom sa dá farba vyjadriť intervalom *hue* zložky
- HSV taktiež umožňuje zmierniť vplyv osvetlenia scény



Trackbar

- Highgui má podporu myši a posuvných ukazovateľov



Inicializácia obrazu

Python

```
img = np.array([[255,0],[0,255]],np.uint8)
```

C++

```
cv::Mat img =  
    (cv::Mat_<uchar>(2,2) << 255, 0, 0, 255);
```

Zdroje pre štúdium

- www.opencv.org
- www.learnopencv.com
- www.pyimagesearch.com
- www.agspace.org/opencv