Seminár Robotika.SK

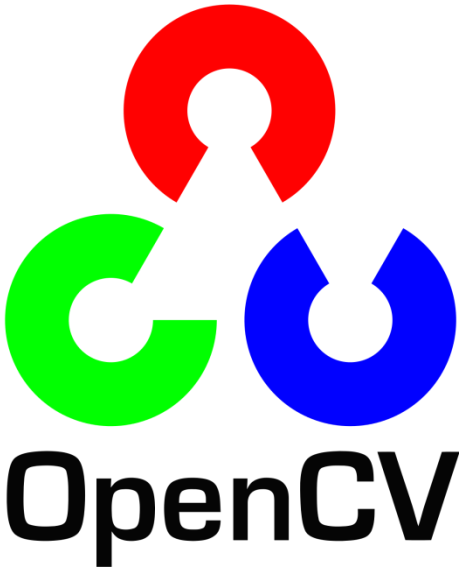
OpenCV

Andrej Lúčny

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

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



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

`www.robotika.sk/seminar/2017/cvicenie5.zip`

Obraz ... cv::Mat

Obrazov je 6x4 typov

- uchar
- char
- u. short int
- short int
- int
- float
- double

- Vec<uchar,3> = Vec3b

	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

Obraz ... cv::Mat

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

Dajú sa použiť templaty na univerzálne metódy s case-ami vo vnútri

Odovzdávanie referenciou

- cv::Mat je hlavička ukazujúca na dáta
- Tieto dáta môžu byť zdieľané viacerými cv::Mat
- = kopíruje hlavičky nie dáta
- Dáta kopíruje `copyTo()` ale snažte sa jeho použitiu vyhnúť
- 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

c1

Binárny obraz ... 0 a 255

- je CV_U8C1 v ktorom sú len hodnoty:
  0 (čierna) a 255 (biela)
- Možno ho získať porovnaním (image > 128)
- alebo prahovaním:
  `threshold(grey,binary,value,255,TRESH_BINARY)`
- alebo adaptívnym prahovaním:
  `adaptiveThreshold(grey,binary,255,`
  `ADAPTIVE_THRESH_MEAN_C,15,0)`
- `Hranovými operátormi`
  `Canny(grey, binary, value, value*3, 3);`

c2

ROI - region of interest

- ROI je realizovaný pomocou operátora `()`
  `image(Rect(0,0,100,100))`
- 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).

c3

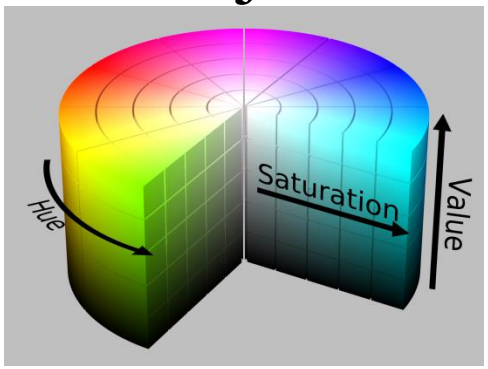
Video

- Video je stream Mat-ov
- Kamery sú číslované 0, 1, ...
- Video musí byť v podporovanom formáte (použite Freemake video converter na konverziu do ...)
- Wifi kamera sa číta cez ffmpeg (opencv musí byť skompilované s jeho podporou)
- waitKey() musí byť zavolaný s nenulovým argumentom
- Driver kamery musí byť nabehnutý, inak sa to môže zrútiť (ale za chvíľu sa to naštartuje normálne)

c4

Farebné modely

- Farba je bežne implementovaná BGR modelom, `Scalar(255,0,0)` je modrá, `Scalar(0,0,255)` červená, `Scalar(255,0,255)` fialová, `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



c5

Momenty

Obrazu priraďujú číslo bez ohľadu na jeho otočenie

- Maximum

```
double mx; minMaxLoc(img,NULL,&mx);
uchar mx=*min_element(img.begin<uchar>(),img.end<uchar>());
```

- Suma

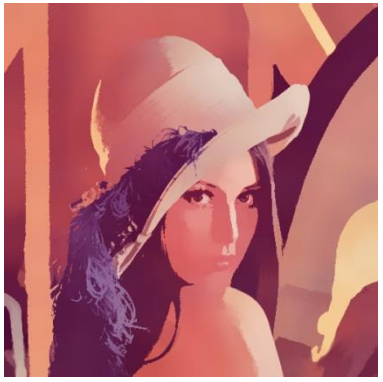
```
double total = sum(sum(img))[0];
//double total = moments(img).m00;
int total=accumulate(img.begin<uchar>(),img.end<uchar>(),0);
```

- Ťažisko

```
Moments m = moments(img);
x= m.m10/m00; y = m01/m00;
```

Filtre

- Priemer
- Vážený priemer
- Medián
- Maximum (Dilatácia)
- Minimum (Erózia)
- Hrany (Canny)
- MeanShift (cartoon)



```
blur(src,dst,Size(kx,ky))
gausianBlur(src,dst,Size(kx,ky),s)
medianBlur(src,dst,ksize);
dilate(src,dst,element);
erode(src,dst,element);
Canny(src,dst,thres,2*thres,3);
pyrMeanShiftFiltering()
Mat element = Mat::ones(ky,kx,tp);
Mat element =
getStructuringElement
(MORPH_CROSS, Size(kx,ky));
```

Detekcia objektov

Objekt sa dá detekovať rôznymi stratégiami:

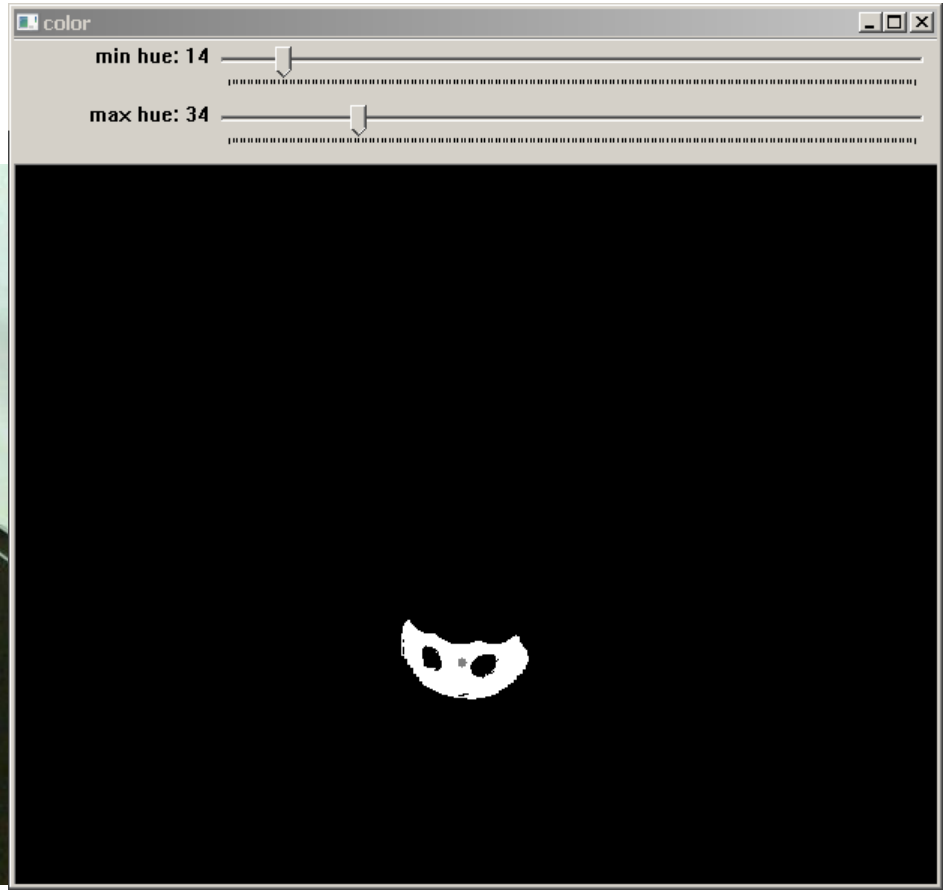
1. Farbou
2. Histogramom farieb (MeanShift, CamShift)
3. Pravidelným tvarom (Houghova transformácia)
4. Nepravidelným tvarom (HOG, Haar)
5. Rozložením význačných bodov (SIFT, SURF)
6. Fázovou koreláciou

c6

c6/1

Detekcia objektov - farbou

1. Obraz prevedieme do HSV
2. Tým pádom farba je vyjadriteľná intervalom

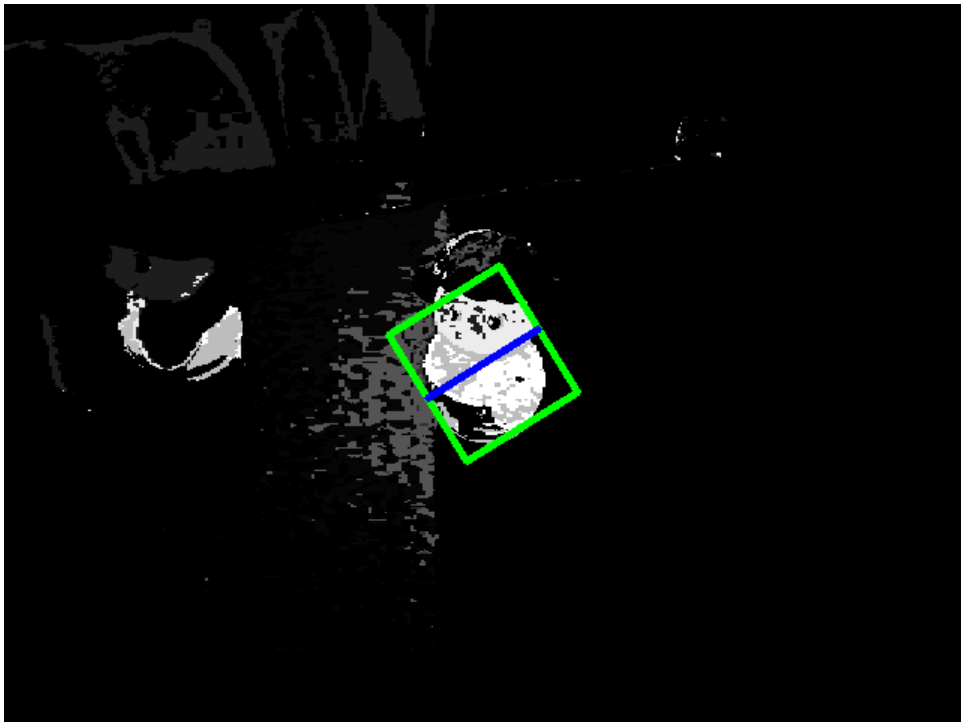


Det. obj. Histogramom farieb

1. Z vybranej oblasti sa urobí histogram farieb
2. Urobí sa šedý backProjection image v ktorom sú farby reprezentované ich výskytom vo zvolenom histograme
3. Pomocou MeanShift posúvame oblasť tak, že zodpovedá najintenzívnejším bodom. (MeanShift je vo všeobecnosti posun z bodu do priemeru blízkych bodov s podobnou farbou
4. Pomocou CamShift hľadáme v oblasti správnu rotáciu

Det. obj. Histogramom farieb

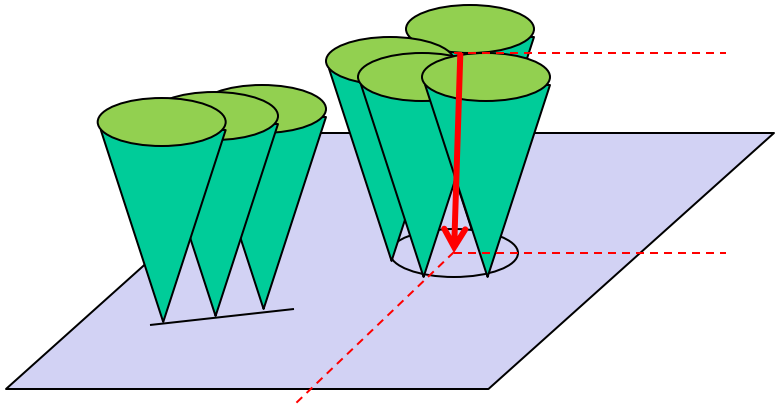
c6/2



Det. obj. pravidelným tvarom

1. Houghova transformácia: obraz transformujeme do priestoru parametrov objektu (v prípade kružnice: center.x, center.y a radius)
2. Obraz binarizujeme a každý svetlý bod hlasuje za všetky parametre pomocou ktorých by mohol byť vykreslený (v prípade kružnice predstavuje hlasovanie jedného bodu v priestore parametrov kužel)
3. Tieto hlasy sa sčítajú a vyberú sa parametre za ktoré zahlasovalo najviac bodov.

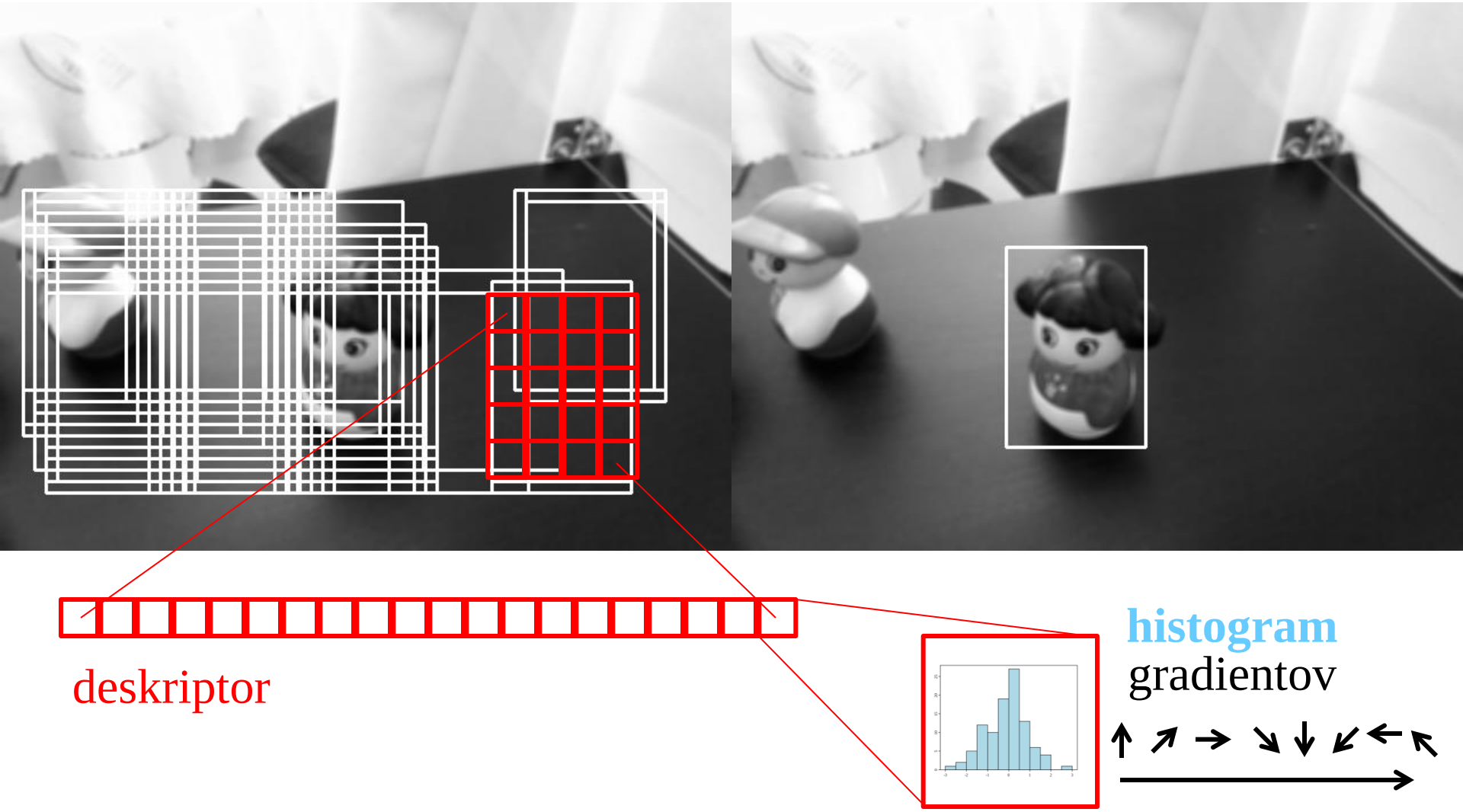
Det. obj. pravidelným tvarom



Det. obj. nepravidelného tvaru

1. HOG detektor.
2. Vybraný objekt sa pokryje sieťou regiónov
3. V každom regióne sa spočíta histogram gradientov.
4. Súbor týchto histogramov zo všetkých regiónov tvorí deskriptor
5. Na danom obraze potom hľadáme oblasti s podobným deskriptorom
6. Spravidla potrebujeme sadu pozitívnych a negatívnych príkladov a učíme z nich klasifikátor, napríklad Support Vector Machine

Det. obj. pravidelným tvarom

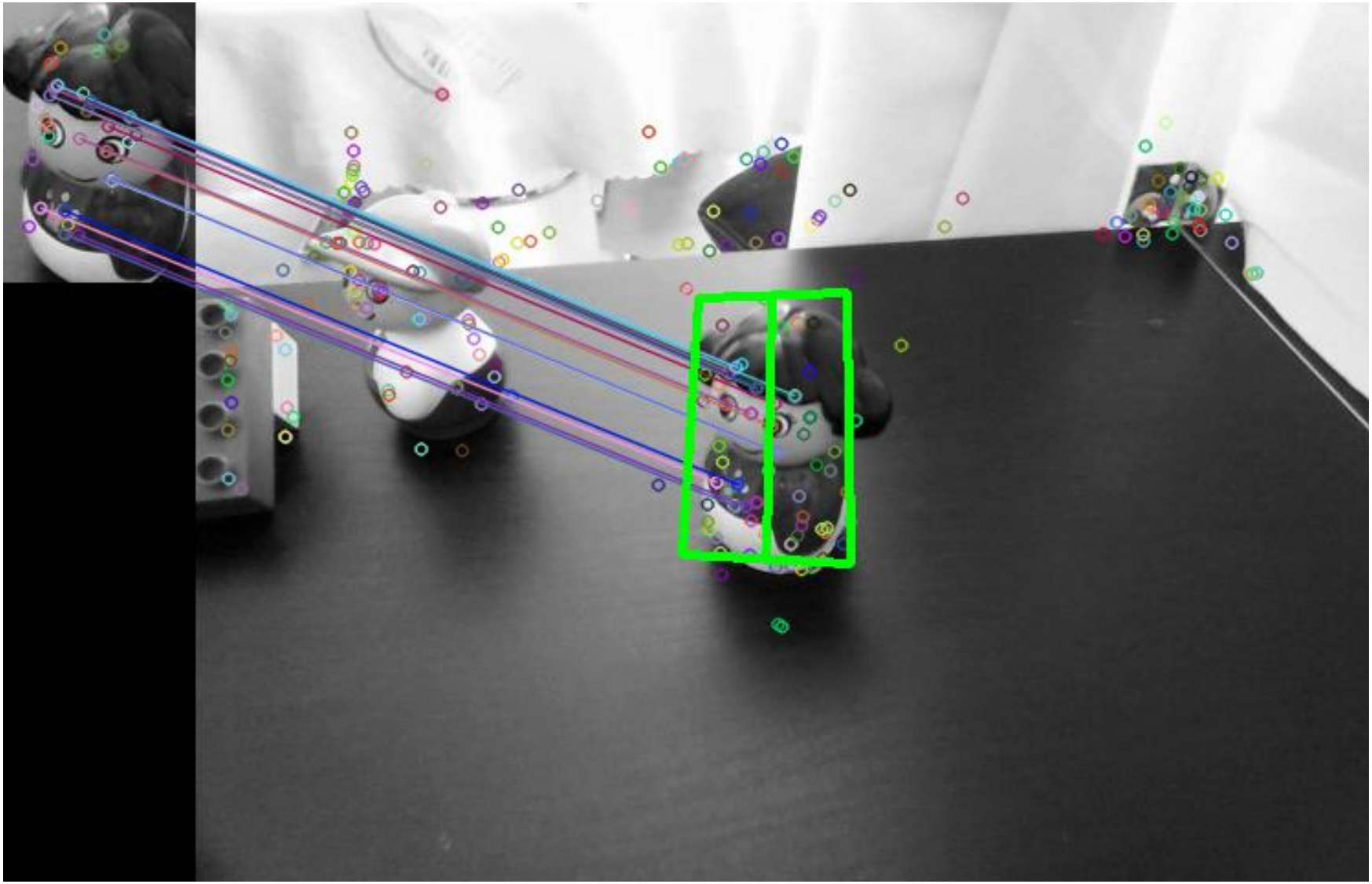


Det. obj. Feature Detector-om

1. Pomocou detektora význačných bodov dostaneme sadu bodov (a ich deskriptorov) zo vzoru a obrazu
2. (detektory SIFT, SURF, ORB tieto body hľadajú ako extrémy v trojrozmernom priestore, kde základňu tvorí obraz a poschodia nad ním čoraz viac rozmazaný obraz. Pritom neskúšajú len pôvodnú veľkosť ale obraz viacnásobne podvzorkujú)
3. Snažíme sa napárovať body vzoru na obraz (RANSAC), pokiaľ sa to dostatočne pekne podarí, zdetekovali sme objekt

Det. obj. Feature Detector-om

c6/5



Det. obj. fázovou koreláciou

1. Binarizujeme vzor i obraz
2. Vybraný vzor sa snažíme priložiť do každého bodu obrazu, spočítať chybu a vybrať miesto s chybou dostatočne malou či minimálnou
3. Efektívne sa tieto chyby dajú spočítať tak, že miesto minima zo sumy druhých mocnín rozdielov sa hľadá maximum zo sumy súčinov.
4. Sumy súčinov môžeme totiž počítať pomocou cirkulárnej konvolúcie, tj. vziať obraz a vzor, urobiť z nich Fourierovu transformáciu, vynásobiť ich spektrá a urobiť inverznú Fourierovu transformáciu

Det. obj. fázovou koreláciou

Ďakujeme za pozornosť

Seminár Robotika.SK

OpenCV

Andrej Lúčny

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

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