Robotické vysavače

pro výuku mobilní robotiky

Pavel Krsek

2. 6. 2021

- Robot Turtlebot a senzory
- Studentské úlohy a kroky k řešení
- Obraz z kamery a postup zpracování
- Hranová detekce, Houghova transformace
- Barevný prostor, segmentace a popis oblastí

https://www.turtlebot.com/

Úvod

České vysoké učení technické v Praze (ČVUT, CTU)

Český institut informatiky, robotiky a kybernetiky (CIIRC)

Výuka

- **Fakulta elektrotechnická**, Katedra kybernetiky
- Laboratoře robotiky (bakalář, 4. semestr)

Další navazující předměty

- Inteligentní robotika (magistr, 2. semestr)
- Mobilní a kolektivní robotika (magistr, 3. semestr)

Robot TurtleBot

Pohon

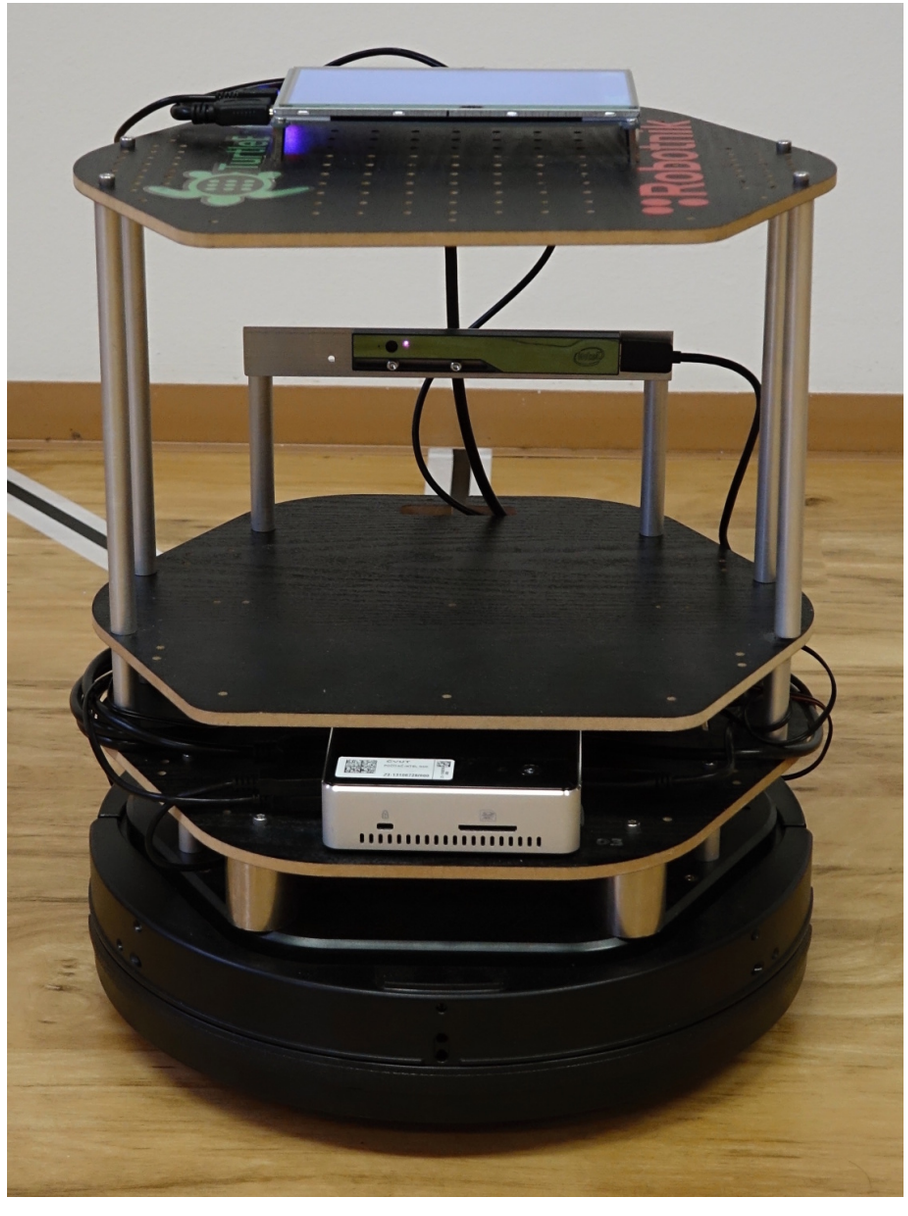
- Dvě samostatně hnaná kola
  (nastavuje se rychlost jízdy a otáčení)

Senzory

- Intel RealSens (Microsoft Kinect)
  - RGB kamera (640x480)
  - 3D hloubková kamera (640x480)
- Nárazník v přední části
- Tlačítka, display ...

Řídicí počítač

- Intel NUC i5
- Paměť 8GB RAM, 128GB SSD



Programové prostředky (software)

Systém

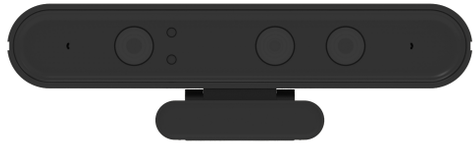
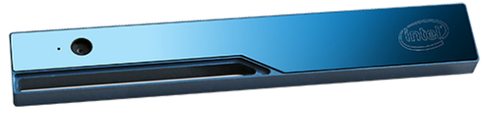
- Operační systém Linux (Ubuntu 18)
- ROS, verze Melodic
  Robot Operating System
- Interface pro Python (objekt)
- Gazebo - simulátor

Programování

- Python 2.7 (omezeno ROS)
- Knihovna OpenCV
  Algoritmy pro zpracování obrazu
- Numpy pro výpočty

Kamera a hlubokový senzor (depth camera)

Parametry hloubkových snímačů

	 Orbbec Astra	 RealSense R200	 RealSense D415
FOV [stupňů]:	60 x 49.5	59 x 45.5	65 x 40
Rozsah [m]:	0.6 ... 8.0	0.5 ... 3.5 (4.0)	0.45 ... 3.0
Chyba XY [mm]:	7.2 (3m)	—	—
Chyba Z [mm]:	12.7 (3m)	10 (2m)	2% (2m)
Rozlišení [px]:	640x480	640x480	1280x720

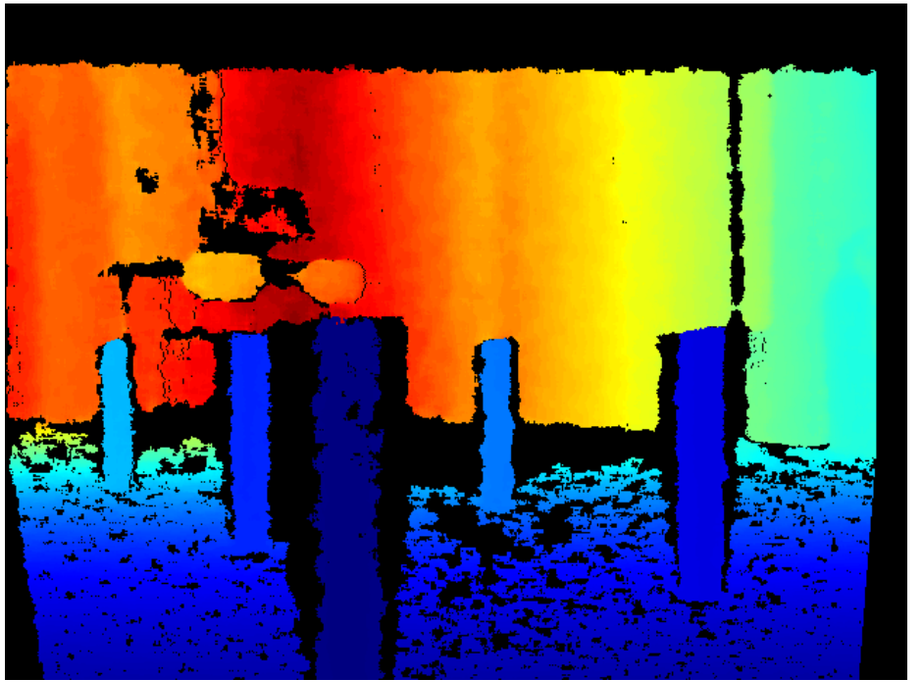
Stránky výrobců

- https://www.intel.com/content/www/us/en/architecture-and-technology
- https://orbbec3d.com

Orázky: stránky výrobců

Snímky poskytované senzorem

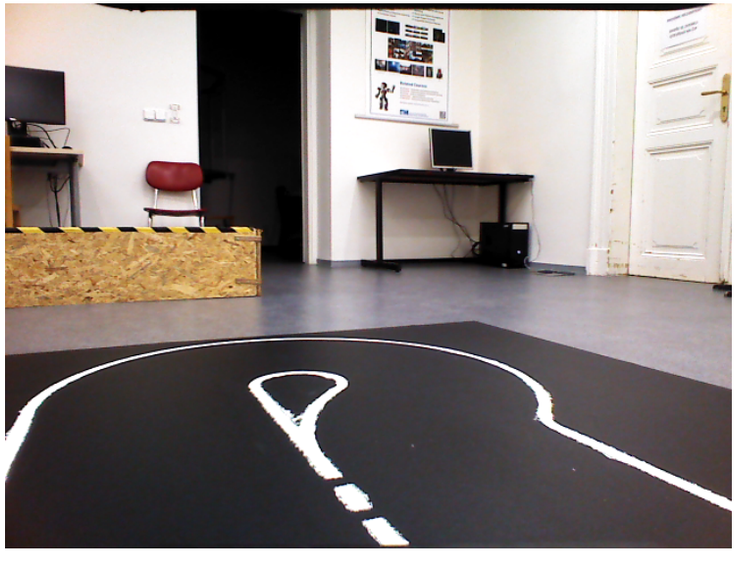
- Rozlišení 640x480 bodů
- RealSense D415 má menší minimální hloubku (31 cm), rychlejší
- Snímky jsou geometricky přibližně zarovnány

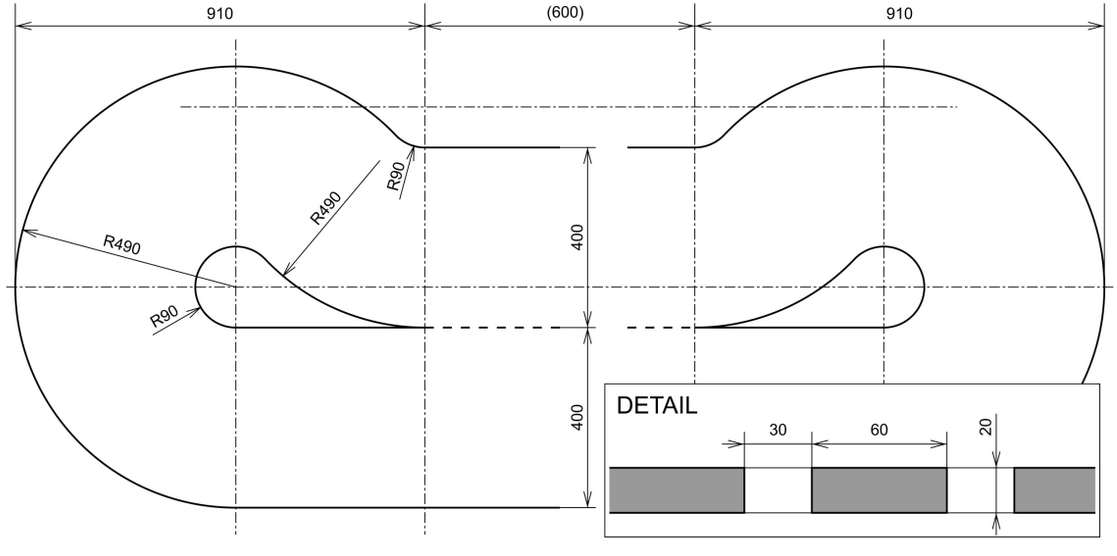


Řešené úlohy

Parkování (příčné, ke zdi)

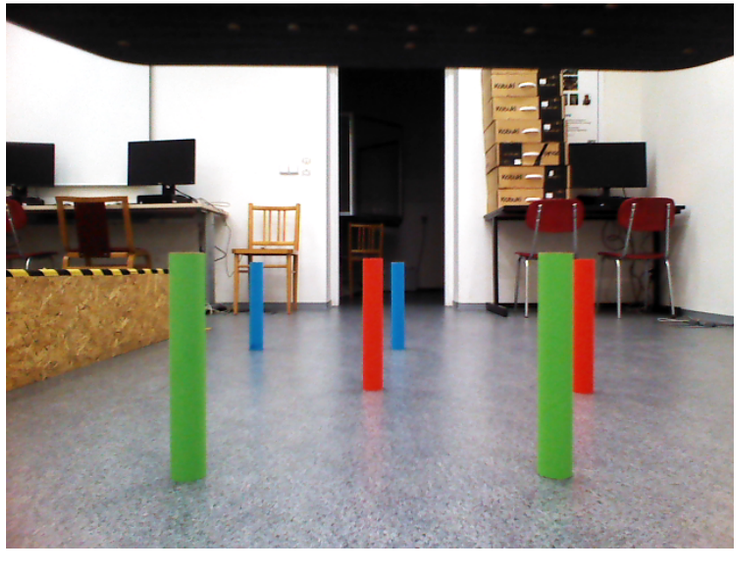
Jízda v pruhu



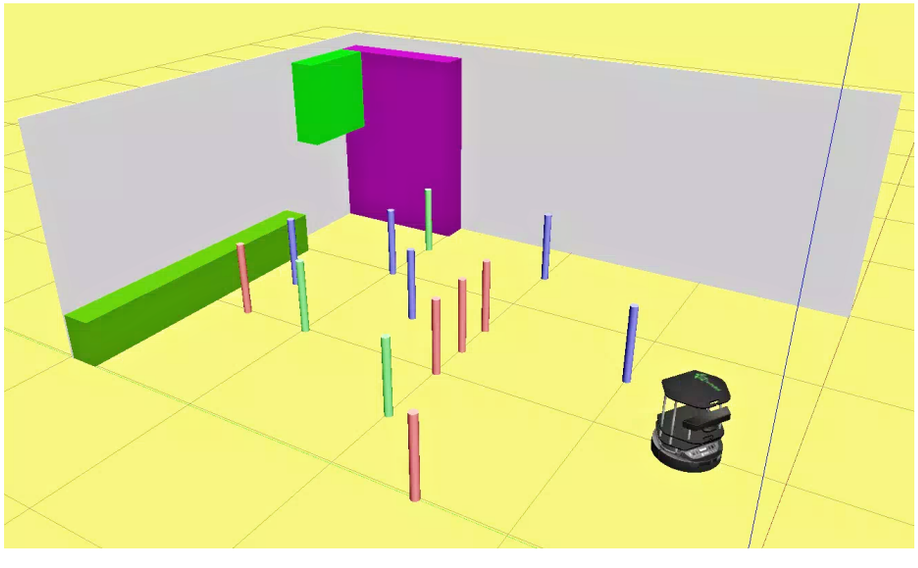


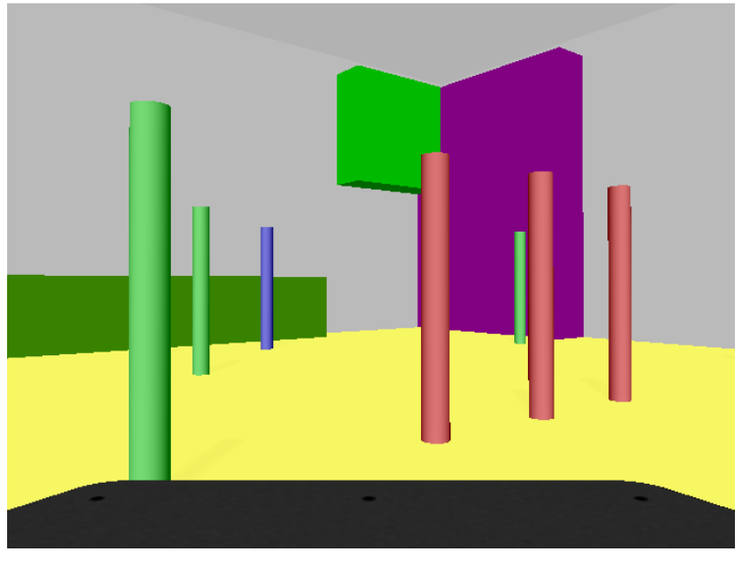
Řešené úlohy

Slalom mezi sloupky



Hledání (srážení) sloupku





Obraz z kamery

Pořízení obrazu a jeho reprezentace

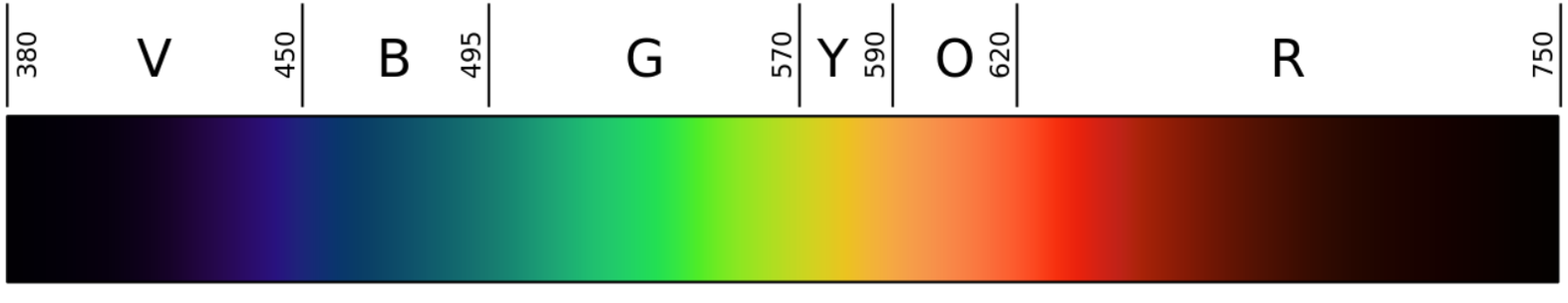
Světlo

Světlo = elektromagnetické záření

Viditelná část spektra: 390–760 nm (390–790 THz)

Parametry záření:

- Frekvence záření (vlnová délka) je detekována kamerou nepřímo.
- Amplituda (intenzita) je snímána kamerou přímo.
- Polarizace (příčné vlny) se snímá pomocí polarizačních filtrů.
- Fáze má význam pouze pro koherentních zobrazení (holografie, interferometrie).



vlnová dálka uvedena v nanometrech [nm]

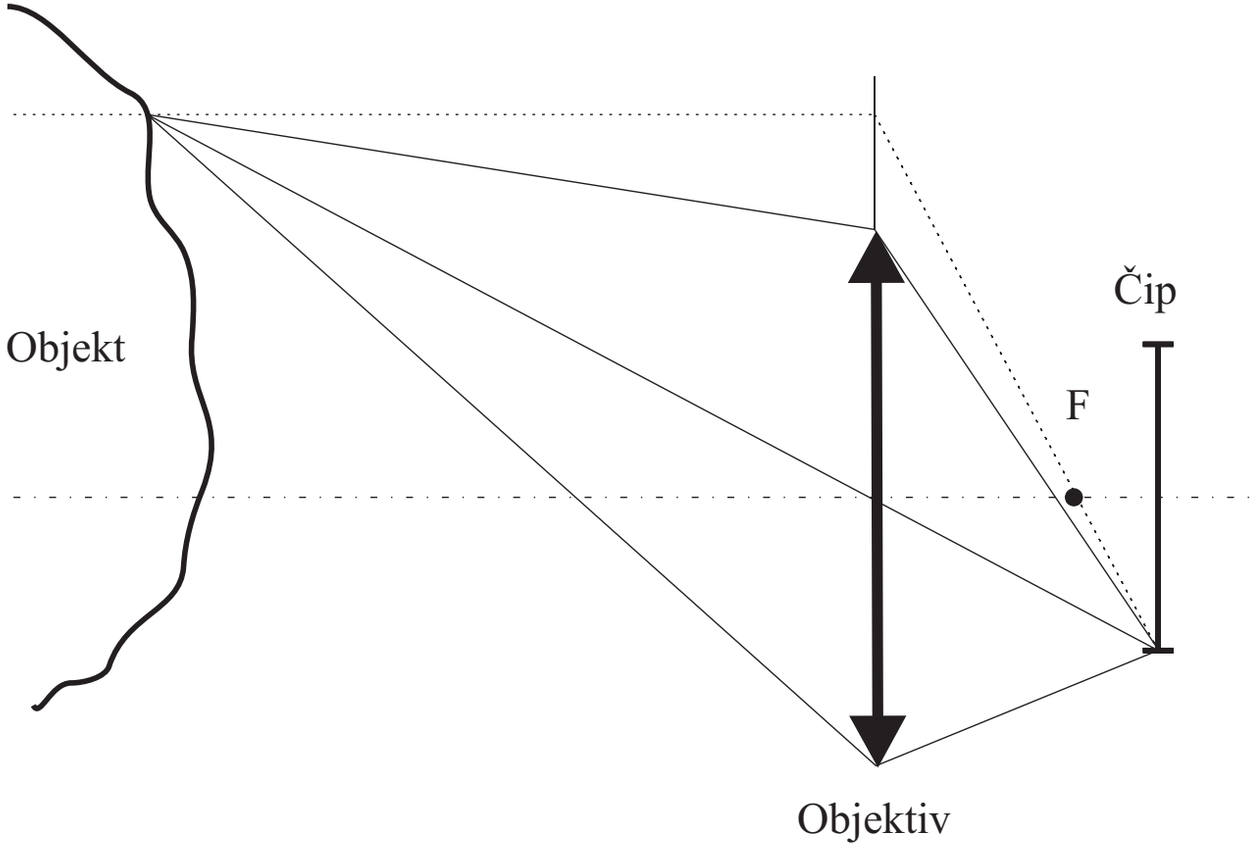
Formování obrazu

Klasický objektiv

- Vzdálenost objektu $\gg$ ohnisková vzdálenost.
- Objektiv modelujeme jako tenkou čočku (středové promítání)

Vznik obrazu

- Světlo se odráží od povrchu objektu.
- Odražené světlo promítá objektiv na čip (snímač).
- Snímač převádí světlo (fotony) na elektrický signál.

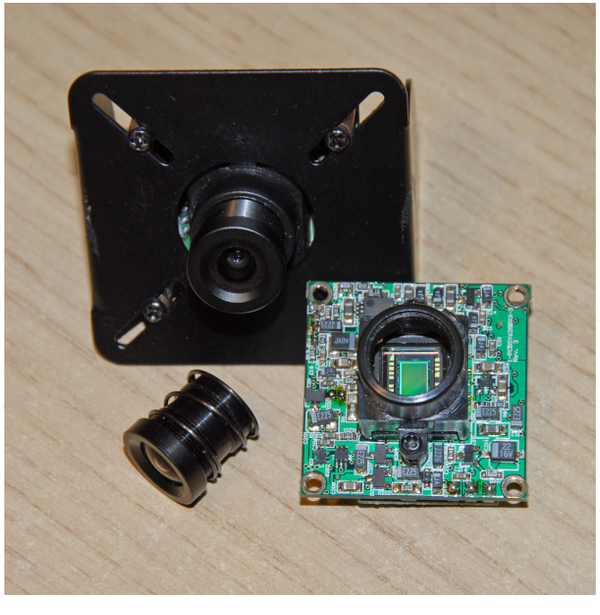


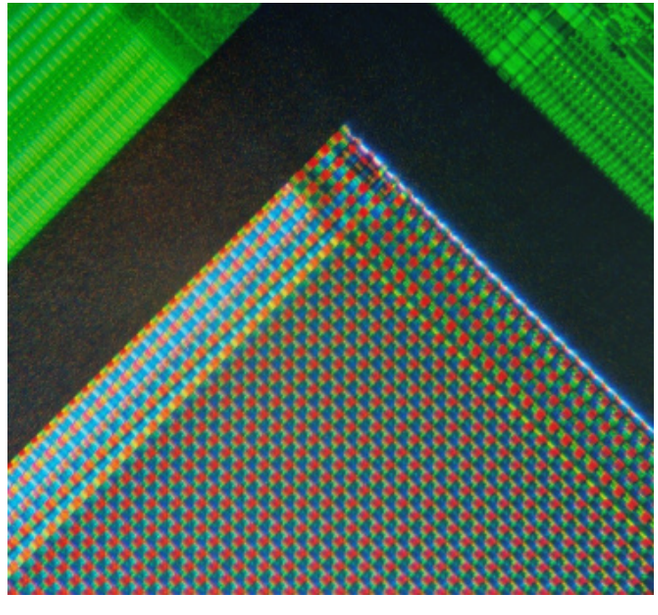
Orázek: autor prof. Václav Hlaváč, CIIRC ČVUT

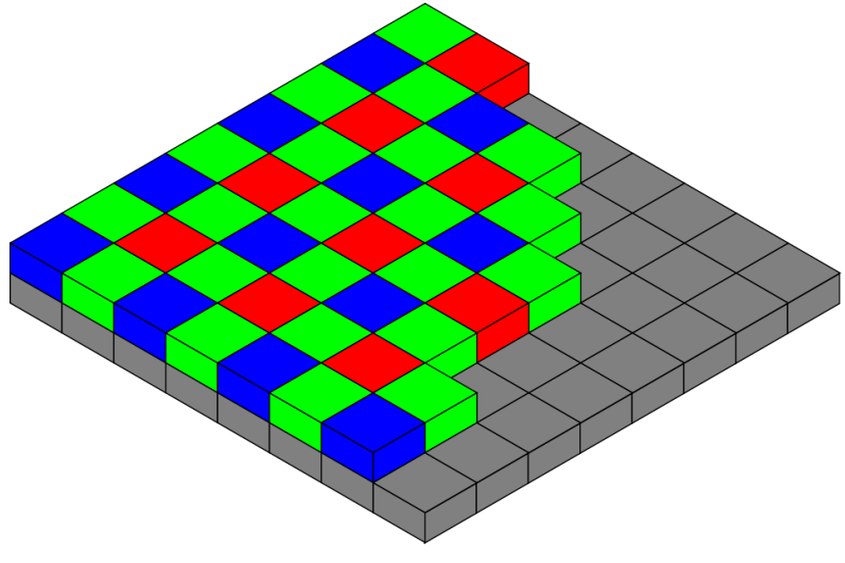
Snímač - obrazový senzor

Obvyklá realizace snímače

- Snímač se skládá z jednotlivých citlivých buněk - obrazových bodů.
- Dopadající světlo (fotony) se v polovodiči mění na nábojové páry (elektron-díra).
- Nábojové páry se ve statickém elektrickém poli mění na proudové impulzy.
- Impulzy jsou po dobu expozice integrovány nabíjením/vybíjením kondenzátoru.
- Obrazové body tvoří pravoúhlou síť se stejnými rozestupy v obou směrech.



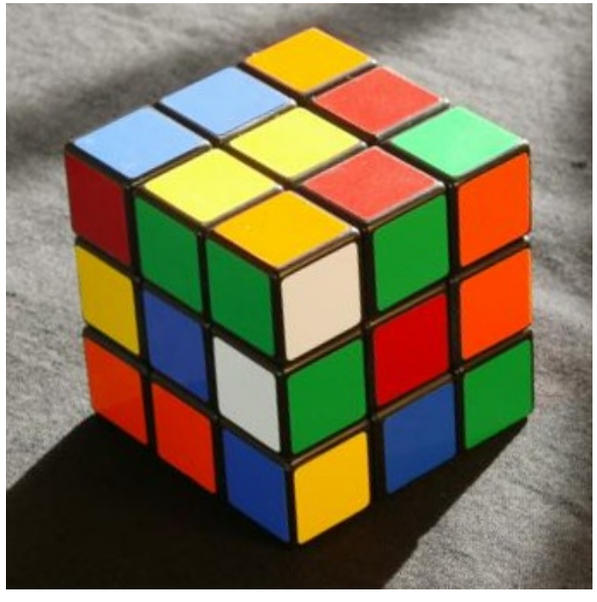




Vliv osvětlení na obraz

Rubikova kostka

- Osvětlená sluncem.
- Stín zprava osvětlen navíc žárovkou.
- Různé nastavení vyvážení bílé.



Barevné spektrum

- Osvětlené sluncem.
- Stín vlevo osvětlen navíc žárovkou.

RGB snímek

`value = image[řádek, sloupec, barva]`

value $\in \langle 0; 255 \rangle$

barva $\in [0, 1, 2] \equiv$ B, G, R

```
im_rgb =  cv2.cvtColor(im_bgr, cv2.COLOR_BGR2RGB)
```

Jak zpracovat obraz

Detekce přímek

- Detekce hran
- Houghova transformace

Segmentace a výběr oblastí

- Barevný prostor a prahování
- Spojité oblasti (labeling)
- Popis oblastí

Detekce hran

Významná změna jasu ve snímku

Převod z barevného na černobílý obraz

Způsoby převodu

1. $I = 0.333 I_R + 0.333 I_G + 0.333 I_B$
2. $I = 0.299 I_R + 0.587 I_G + 0.114 I_B$

Realizace

```
gray =  cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
```

1

2

Detekce hran ve snímku (edge detection)

Jednoduše: Hrana je místo ve snímku, kde se výrazně mění jeho jas (intenzita).

Sobelův hranový detektor

- Konvoluce obrazu s maskou pro svislé a vodorovné hrany:

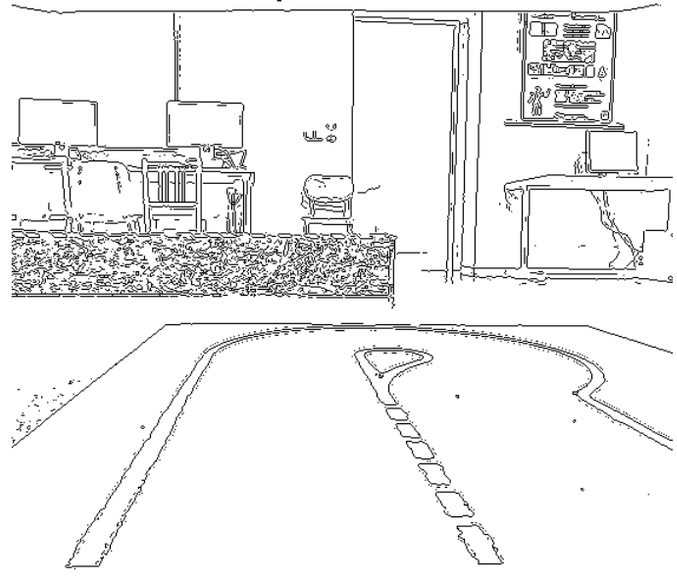
$$K_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, K_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

- Intenzita hrany v daném bodě je $G = \sqrt{G_x^2 + G_y^2}$.
- Nastavuje se práh intenzity G.

Realizace

```
Gx = cv2.Sobel(img, cv2.CV_64F, 1 ,0, ksize=3)
Gy = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3)
G = np.sqrt(Gx**2 + Gy**2)
```

Vstupní snímek



Sobelův detektor

Detekce hran na základě 2. derivace

Jinak: Extrém detekujeme jako pomocí derivace.
Proto: Používáme 2. derivaci jasové funkce

Metoda Laplacián/Gaussián

- Detekce používá Laplaceův operátor kombinovaný s Gausovským filtrem.
- Konvoluční jádro:

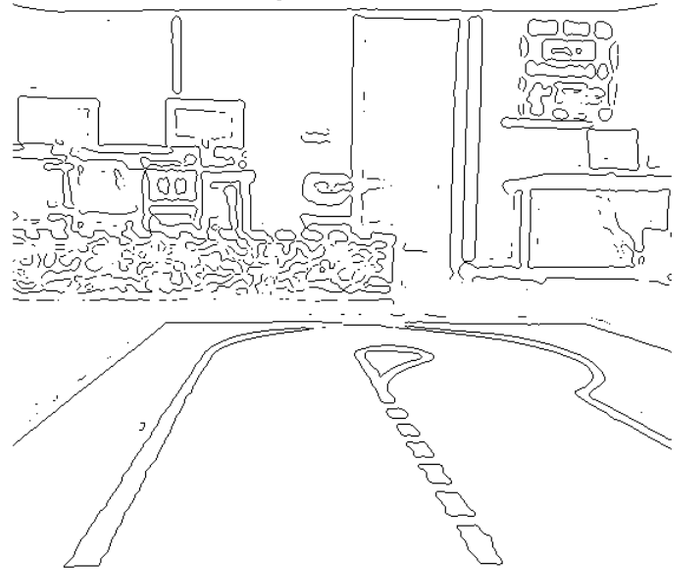
$$K_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

- Detekují se průchody nulou 2. derivace obrazové funkce
- Nastavuje se práh na detekci hrany a σ filtru.

Realizace

```
laplacian =
  cv2.Laplacian(img, cv2.CV_64F, ksize = 1)
```

Vstupní snímek



Laplacián/Gaussián

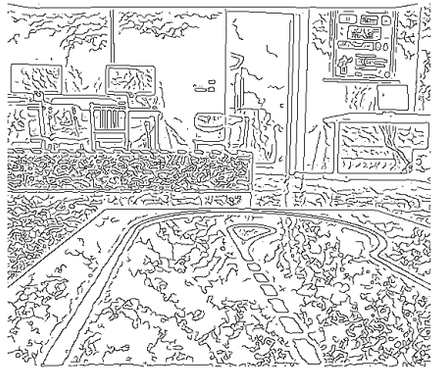
Cannyho hranový detektor (1986)

Popis algoritmu

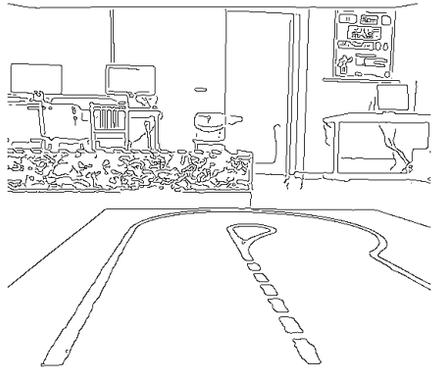
1. Omezení vlivu šumu Gaussovým filtrem.
2. Hrany jsou detekovány Sobelovým operátorem.
3. Intenzita a směr $G = \sqrt{G_x^2 + G_y^2}$, $\Theta = atan2(G_y, G_x)$.
4. Provedeno "ztenčení" hran ve směru kolmém na hranu (4 směry). Ponechá pouze body s maximální intenzitou.
5. Dvojité prahování a sledování hrany:
   - Jsou vybrány body, které mají intenzitu větší než horní práh T_1 (silné hrany).
   - Postupně jsou jako hranové body označovány body sousedící s hranovými, pokud je intenzita hrany v daném bodě větší než dolní práh T_2.

Realizace

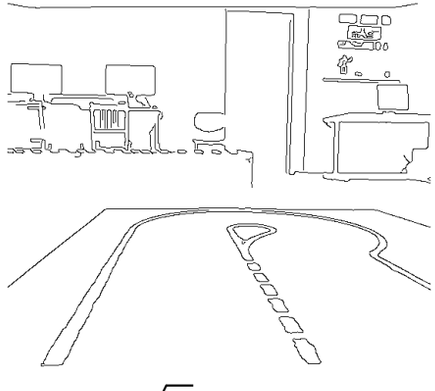
```
edges = cv2.Canny(img, T1, T2, apertureSize = 3)
```



$\sigma = \sqrt{2}$, $T_1 = 0.01$



$\sigma = \sqrt{2}$, $T_1 = 0.1$



$\sigma = \sqrt{2}$, $T_1 = 0.3$

Houghovy transformace

Detekce přímek v obraze (detekované hrany)

Houghova transformace (1971)

Principy Houghovy transformace

- Hledáme instance křivky definované rovnicí $f(\mathbf{u}, \mathbf{a}) = 0$, kde $\mathbf{u}$ jsou souřadnice ve snímku a $\mathbf{a}$ je parametrizace hledané křivky.
- Vstupem jsou obvykle body na hranách (edge detection).
- Každý hraniční bod může ležet na nekonečně mnoho instancích hledaných křivek.
- Transformujeme body ze souřadnic snímku do souřadnic parametrů $\mathbf{a}$ křivky.

Algoritmus Houghovy transformace

1. Provedeme kvantizaci vektoru parametrů $\mathbf{a}$.

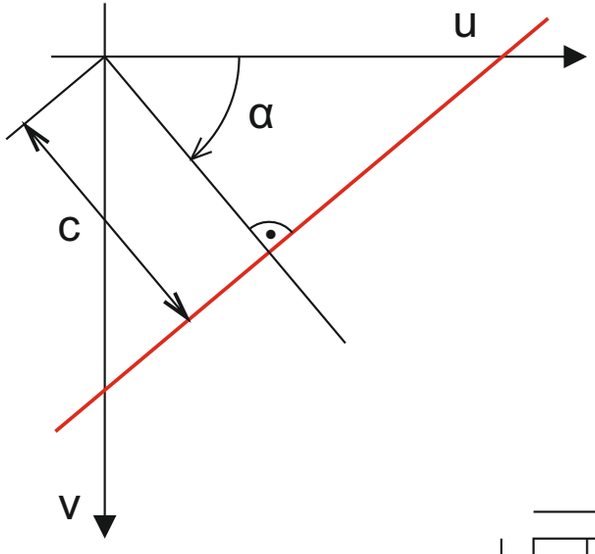
2. Vytvoříme n-rozměrné pole akumulátoru $\mathbf{A}(\mathbf{a})$, které strukturou odpovídá parametrům $\mathbf{a}$.

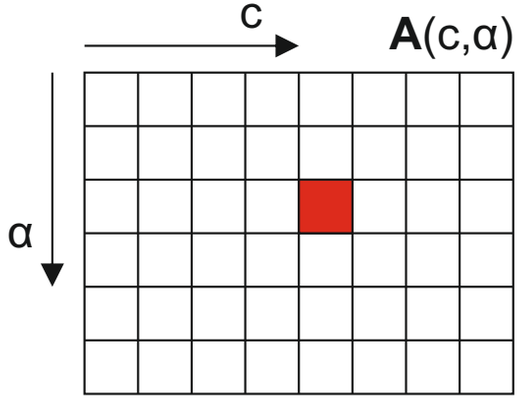
3. Pro každý nalezený hraniční bod $\mathbf{u} = [u, v]$ v obraze budeme inkrementovat všechny prvky akumulátoru $\mathbf{A}(\mathbf{a})$ pro něž platí $f(\mathbf{u}, \mathbf{a}) = 0$.

4. Po započtení všech hraničních bodů platí, že lokální maxima v akumulátoru $\mathbf{A}(\mathbf{a})$ odpovídají jednotlivým instancím křivky $f(\mathbf{u}, \mathbf{a}) = 0$

Parametrizace přímky

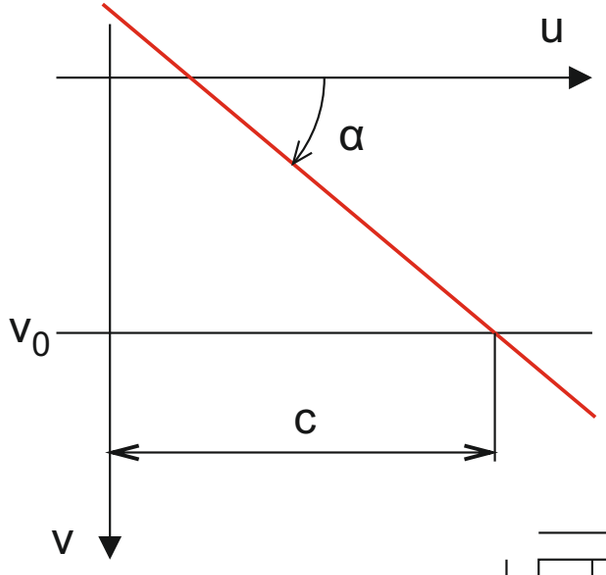
Obvyklá parametrizace

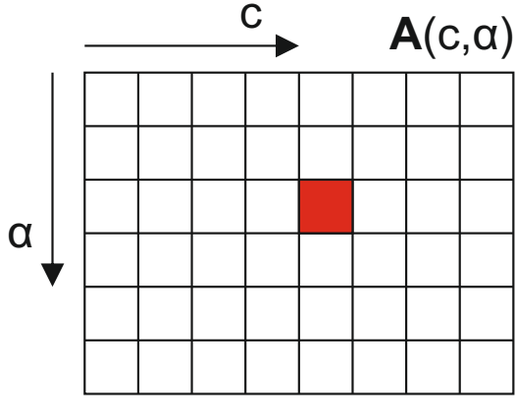




$$u \cos \alpha + v \sin \alpha = c$$

Alternativní parametrizace



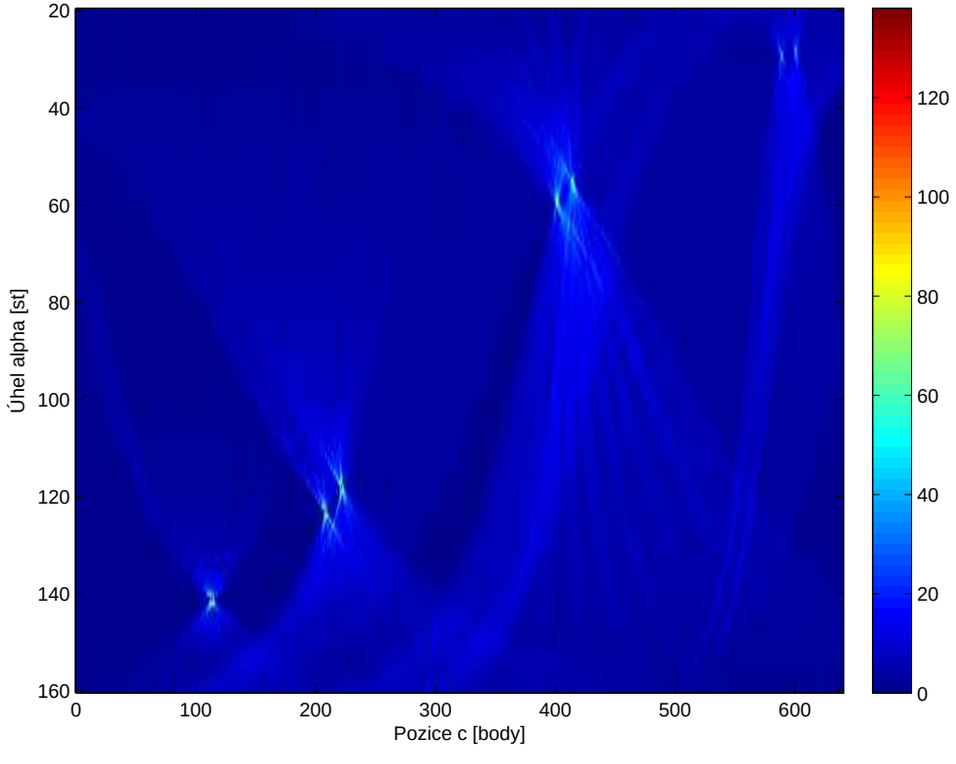


$$u - \cot \alpha (v_0 - v) = c$$

Implementace pro přímky

- Vstupem je seznam bodů na hranách
- Hledáme jízdní parkovací pruh. Proto se omezíme na body s řádkovou souřadnicí > 270.
- Rozsah parametrů volíme $\alpha \in \langle 20°; 160° \rangle$ a $c \in \langle 0; 640 \rangle$ obrazových bodů.
- Kvantizace je $1°$ a 1 obrazový bod.
- Řádek pro měření polohy $v_0 = 350$

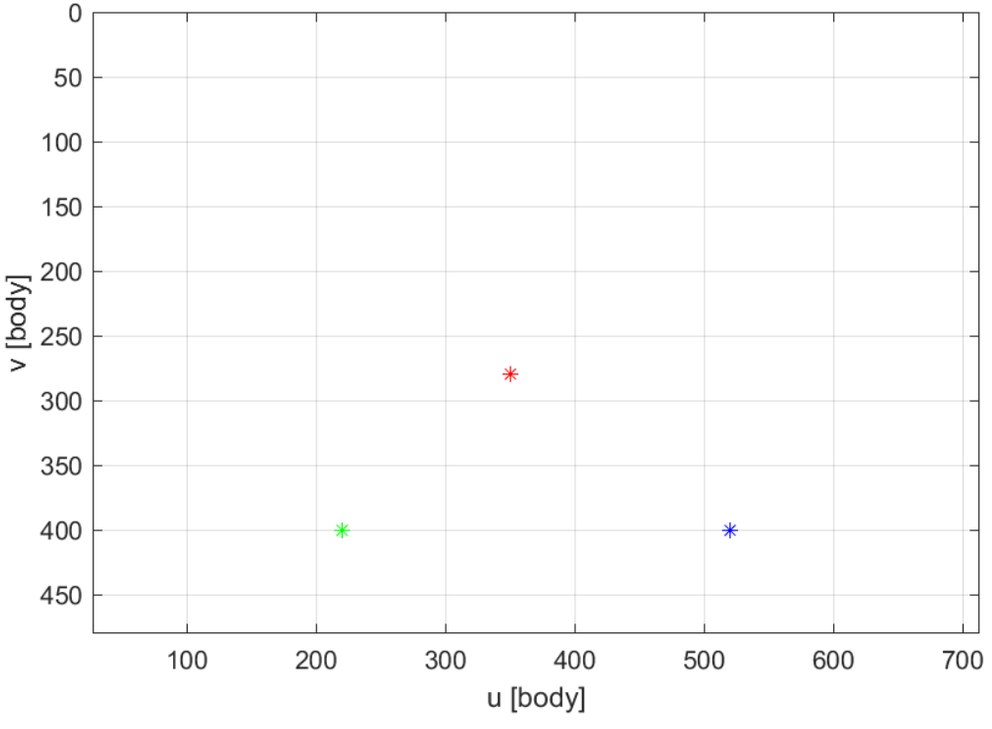
Vstupující hranové body



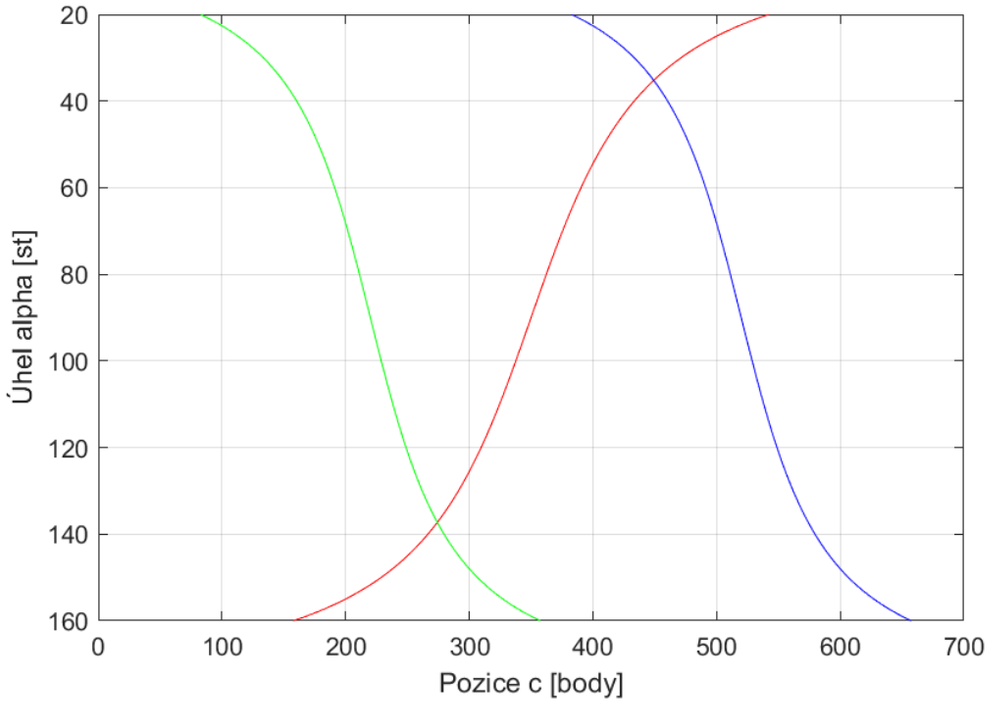
Výsledek transformace

Houghova transformace pro přímky

Body v obraze



Pozice v akumulátoru



1. Provedeme kvantizaci parametrů α, c.
2. Vytvoříme n-rozměrné pole akumulátoru $\mathbf{A}(\alpha, c)$.
3. Pro každý bod hrany $\mathbf{u} = [u, v]$ inkrementuje pozice v akumulátoru $\mathbf{A}(\alpha, c)$, které určíme ze vztahu: $u - \cot\alpha(v_0 - v) = c$
4. Po započtení všech hraničních bodů platí, že lokální maxima v akumulátoru $\mathbf{A}(\alpha, c)$ odpovídají jednotlivým přímkám

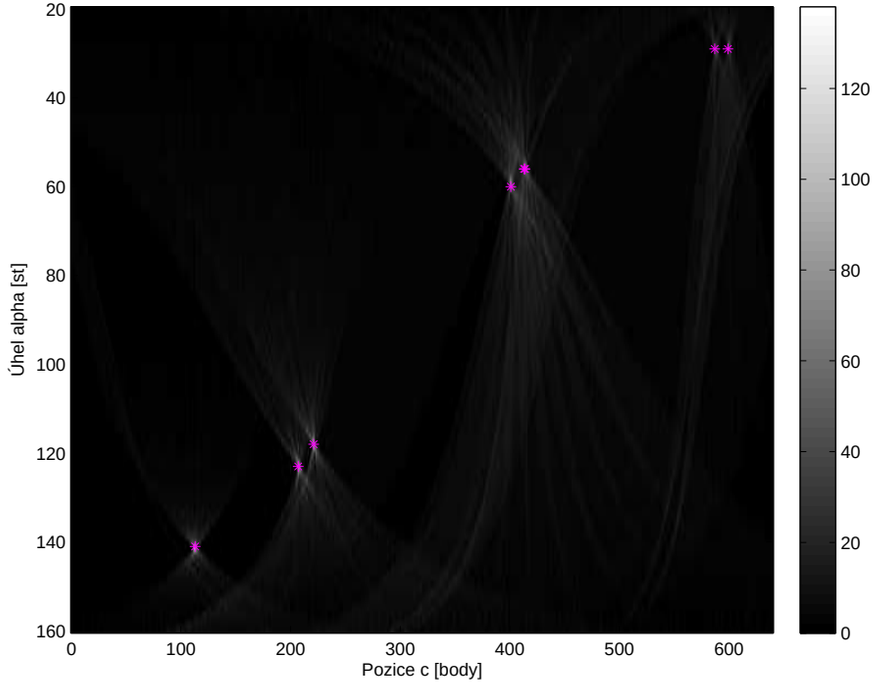
Detekce přímek v transformovaném obrázku

Lokalizace hran

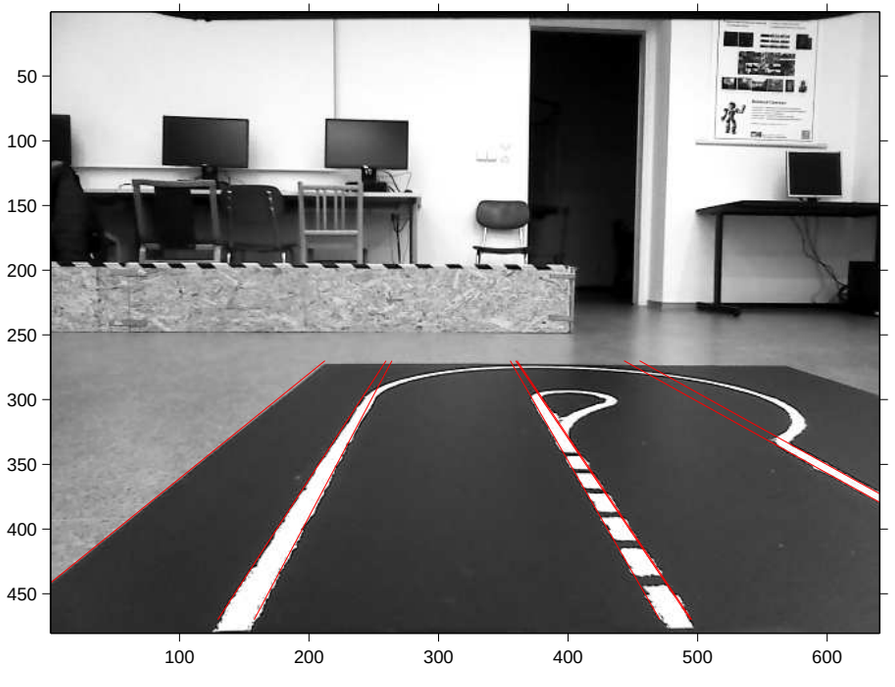
- Hledáme maxima v transformované obrázku.
- Maximu splňuje požadavek na minimální podporu > 30 hranových bodů.
- Na základě parametrů α a c nalezených maxim konstruujeme přímky.

Realizace

```
cv2.HoughLines(image, rho, theta, threshold)
```



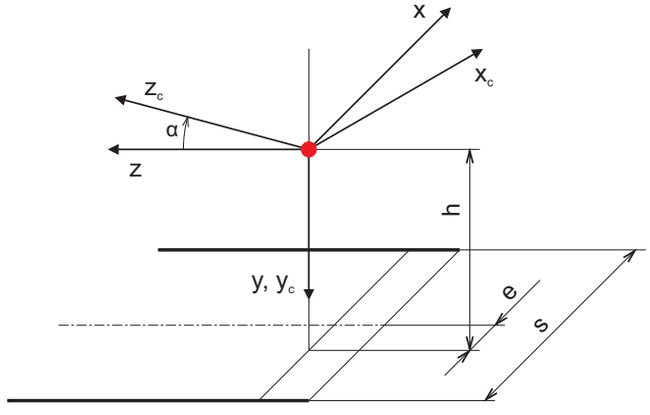
Detekovaná maxima



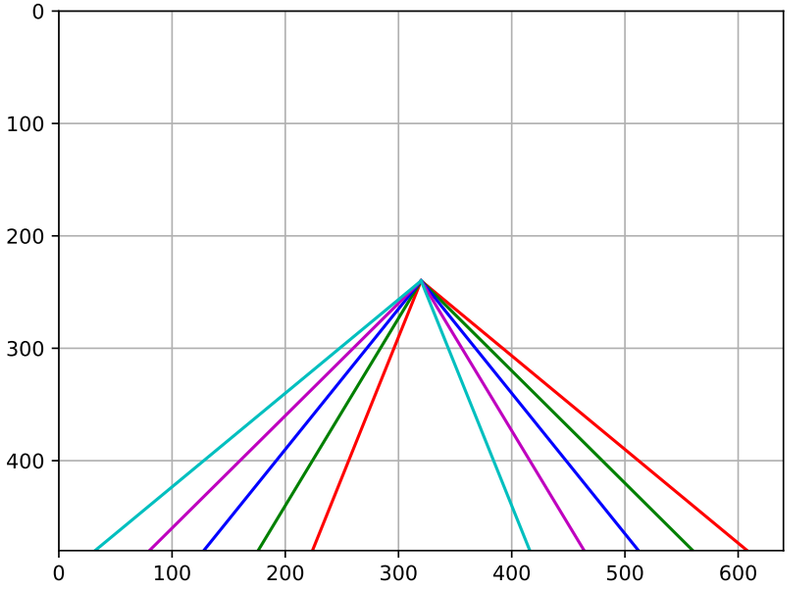
Odpovídající přímky

Interpretace výsledků

- Jízdní pruh, parkovací místo tvoří rovnoběžky
- Průsečík rovnoběžek v obraze - úběžník (vanishing point)
- Jak ovlivní promítané rovněběžky poloha robotu?



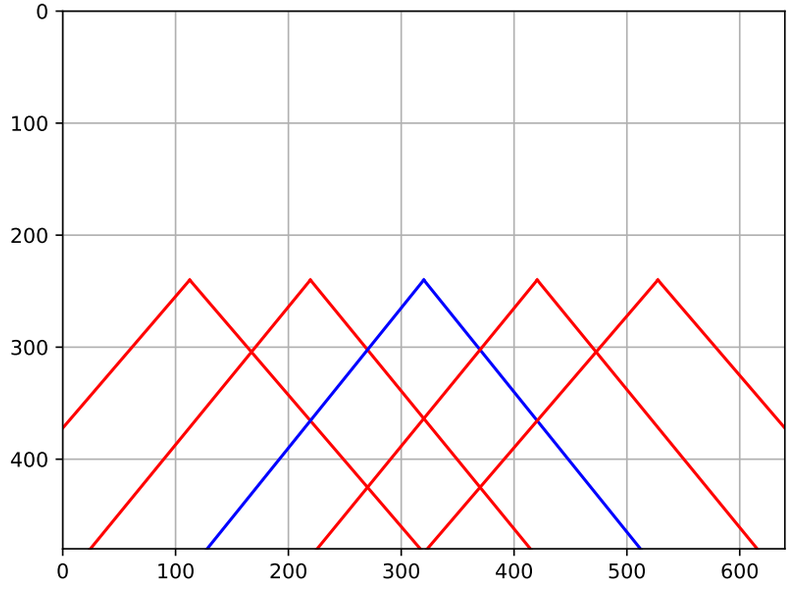
Příčný posun



$s = 0.4$ m, $h = 0.25$ m, $\alpha = 0°$

$e \in \{-0.1,\ -0.05,\ 0.0,\ 0.05,\ 0.1\}$ m

Otáčení (svislá osa)



$s = 0.4$ m, $h = 0.25$ m, $e = 0$ m

$\alpha \in \{-20°,\ -10°,\ 0°,\ 10°,\ 20°\}$

Segmentace a popis oblastí

Prahování v prostoru HSV, spojité oblasti, popis oblastí

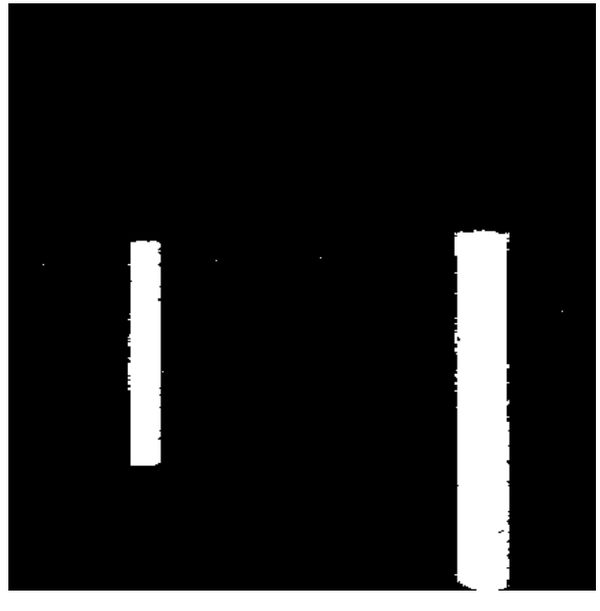
Segmentace

Princip segmentace

- Podstatou je rozdělit obraz na popředí (objekty zájmu) a pozadí.
- Výstupem segmentace je binární obrázek.
- Obrazové body popředí/pozadí jsou obvykle reprezentovány log. 1/0.
- Existuje řada různých metod segmentace

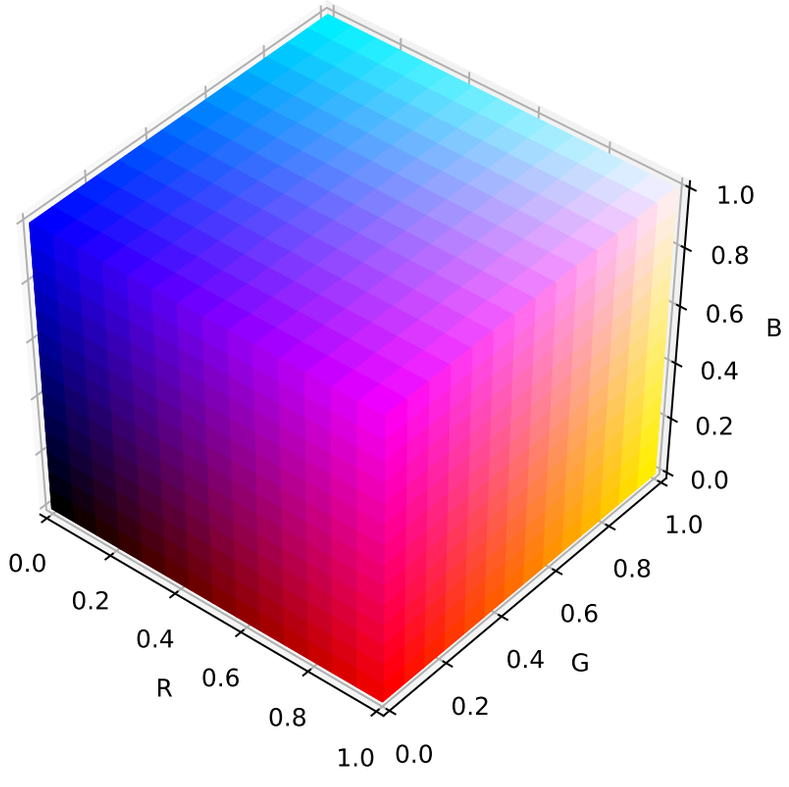
Příklady segmentačních metod:

- Prahování (tresholding)
- Shluková analýza (K-means)
- Hranová detekce (edge detection)
- Narůstání oblastí na základě podobnosti bodů
- Aktivní kontury (active contours)
- Statistické metody (Markov random fields,...)



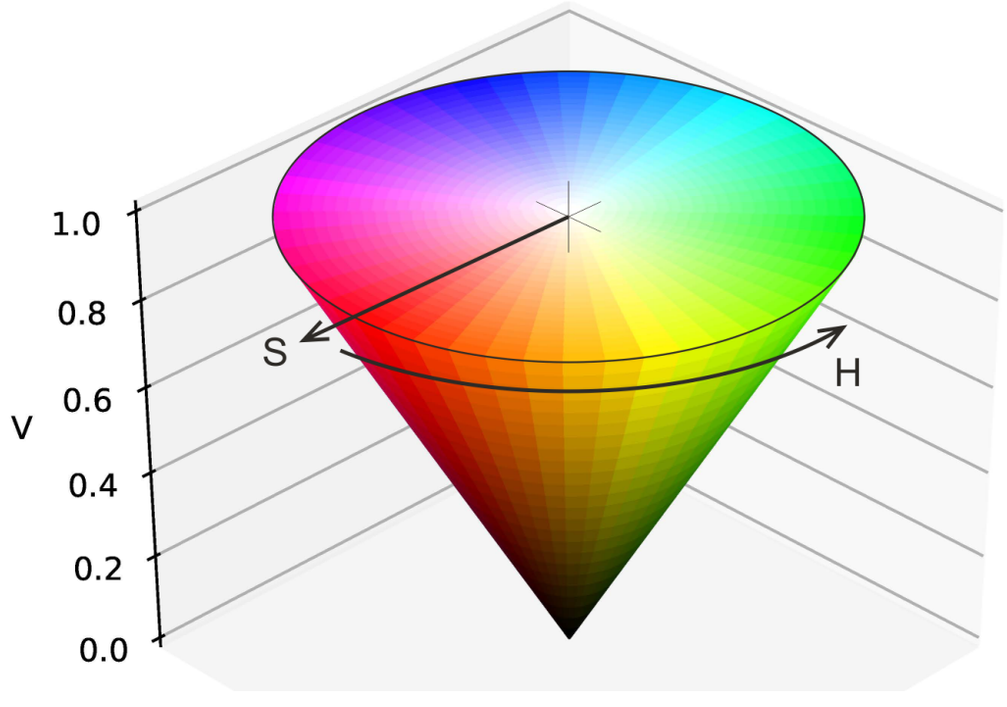
Barevný prostor

RGB



- Red - intenzita červeného kanálu.
- Green - intenzita zeleného kanálu.
- Blue - intenzita modrého kanálu.

HSV



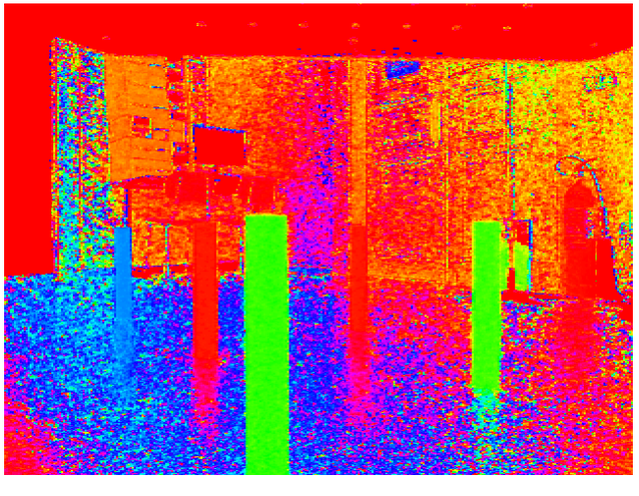
- Hue - odstín barvy odpovídá dominantní vlnové délce (spektrální barvě).
- Saturation - sytost barvy popisuje, jak je barva vzdálena od neutrální šedé/bílé.
- Value - hodnota jasu vyjadřuje kolik světla barva odráží.

Snímek v prostoru HSV

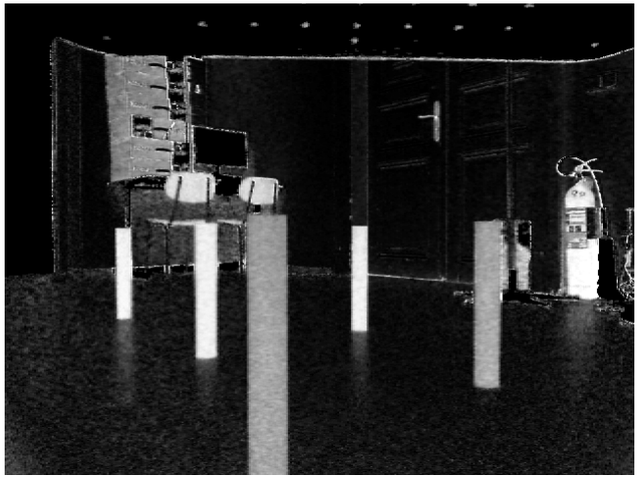
element $= I^{HSV}$ [řádek, sloupec, barva], barva $\in [0, 1, 2] \equiv$ H, S, V

```
hsv =  cv2.cvtColor(im, cv2.COLOR_BGR2HSV)
```

	BGR	HSV
Modrá:	[255, 0, 0]	[120, 255, 255]
Zelená:	[0, 255, 0]	[60, 255, 255]
Červená:	[0, 0, 255]	[0, 255, 255]
Bílá:	[255, 255, 255]	[0, 0, 255]



H $\in \langle 0; 179 \rangle$



S $\in \langle 0; 255 \rangle$

V $\in \langle 0; 255 \rangle$

Prahování (thresholding)

- **Zvolíme referenční barvu (zelená)**

  $I_{ref}^{BGR} = [45, 130, 55] \rightarrow I_{ref}^{HSV} = [56, 167, 130]$

- **Definujeme podmínky**

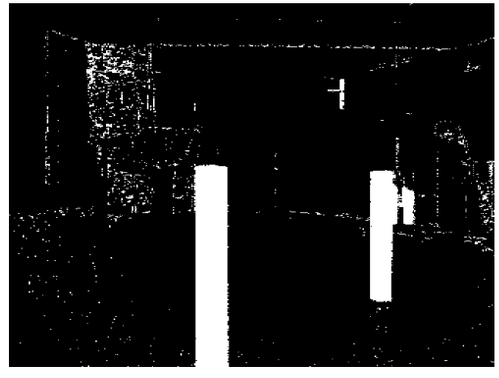
  1. Odstín barvy je podobný referenční:
     $\mid I_{ref}^{H} - I^{H} \mid < t_1$
  2. Tmavé oblasti (nejistá barva) nejsou popředím:
     $I^{V} > t_2$
  3. Hledané objekty jsou barevně saturované:
     $I^{S} > t_3$

- **Stanovíme jednotlivé prahy**

  - Prahy nastavujeme na základě výsledku segmentace
  - Vždy nutno testovat na větším počtu obrázků (různé podmínky)

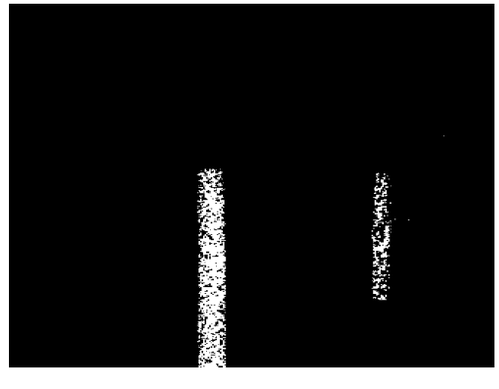
- **Realizace**

```
ret,mask = cv2.threshold(img,threshold,255,cv2.THRESH_BINARY)
binary_mask = img > threshold
```



Mírný práh

Optimální práh



Přísný práh

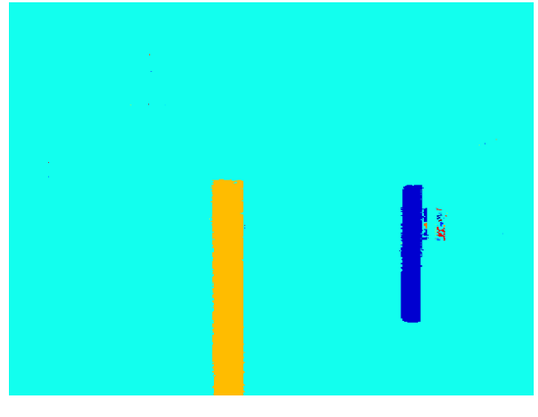
Označení spojitých oblastí (labeling)

- Vstupem je binární obraz (pozadí/popředí)
- Výstupem je obraz s indexy (labels) spojitých oblastní
- Obrazové body každé spojité oblasti jsou označena vlastním indexem
- Spojitost oblasti: 4-okolí, 8-okolí

Realizace v OpenCV

```
out = cv2.connectedComponentsWithStats( binary_mask.astype(np.uint8) )
```

out[0] ... počet oblastí, out[1] ... obraz s index oblastí
out[2] ... parametry oblastí [bod nejvíce vlevo, bod nejvýše, šířka, výška, plochy],
out[3] ... střed / těžiště oblasti.
POZOR: První oblast je pozadí (index 0)



Popis oblastí

OpenCV počítá většinu parametrů na základě obrysu (countour).

```
contours, hierarchy =
  cv2.findContours(interest, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
cnt = contours[k]
```

Parametry spojitých oblastí (Area & Contour Properties)

- Plocha S a obvod O
  Kompaktnost (compactness) $= \frac{O^2}{S}$

```
area = cv2.contourArea(cnt)
lng = cv2.arcLength(cnt,True)
```

- Střed oblasti (těžiště)

```
M = cv2.moments(cnt)
cx = (M['m10']/M['m00'])
cy = (M['m01']/M['m00'])
```

- Momenty spojité oblasti
  $m_{pq} = \sum_u \sum_v u^p v^q I_b(u,v)$
  $\mu_{pq} = \sum_u \sum_v (u-\bar{u})^p (v-\bar{v})^q I_b(u,v)$

```
x,y,w,h = cv2.boundingRect(cnt)
```

- Nejmenší opsaný obdélník (strany a, b)
  Pravoúhlost (rectangularity) $= \frac{a\ b}{S}$

```
rect = cv2.minAreaRect(cnt)
box = cv2.boxPoints(rect)
```

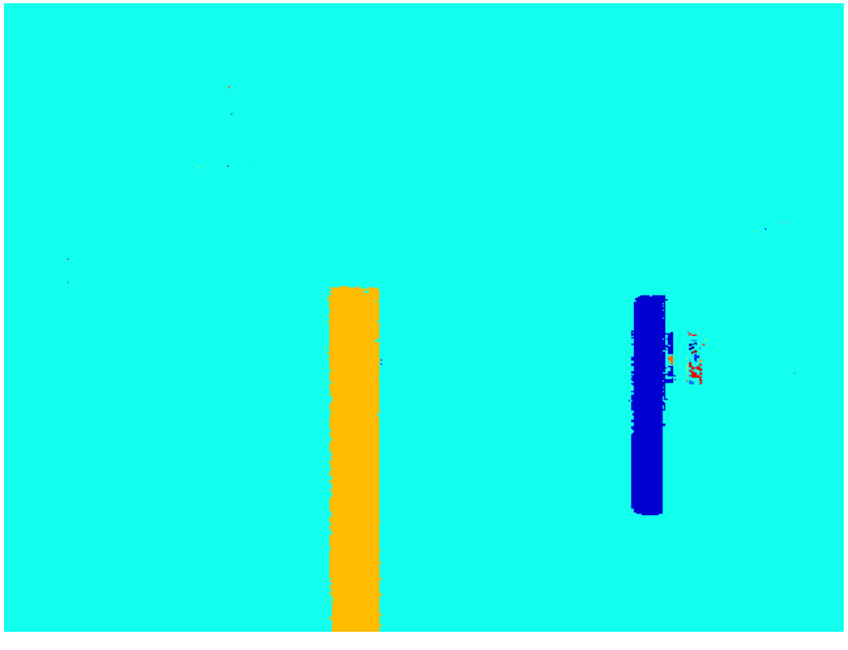
- Konvexní obal (plocha H, obvod)
  Členitost (solidity) $= \frac{S}{H}$

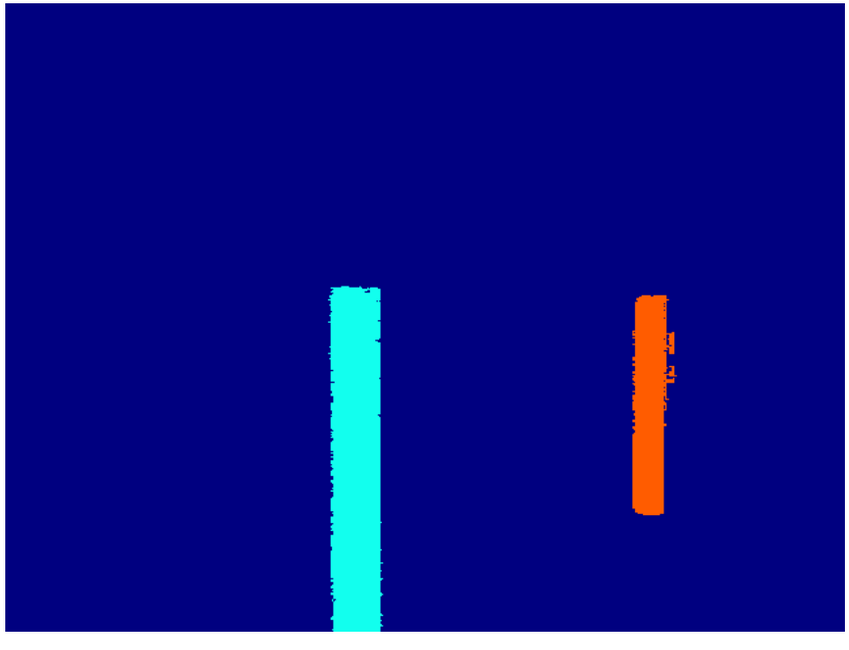
```
hull = cv2.convexHull(cnt)
```

Výběr oblastí zájmu

Parametry nalezených oblastí

Objekt	1	2	3	čtverec	kruh
Těžiště [px]	266.8, 347.6	490.9, 304.9	526.6, 281.4		
Šířka, výška [px]	42, 264	34, 168	10, 17		
Plocha [px]	9528.0	3934.5	33.5		
Výška / šířka [-]	6.29	4.94	1.70	1.0	1.0
Kompaktnost $\frac{O^2}{S}$ [-]	52.43	64.58	205.70	16.0	12.57
Pravoúhlost $\frac{a\,b}{S}$	1.09	1.40	4.29	1.0	1.27
Členitost $\frac{S}{H}$	0.92	0.83	0.29	1.0	1.0

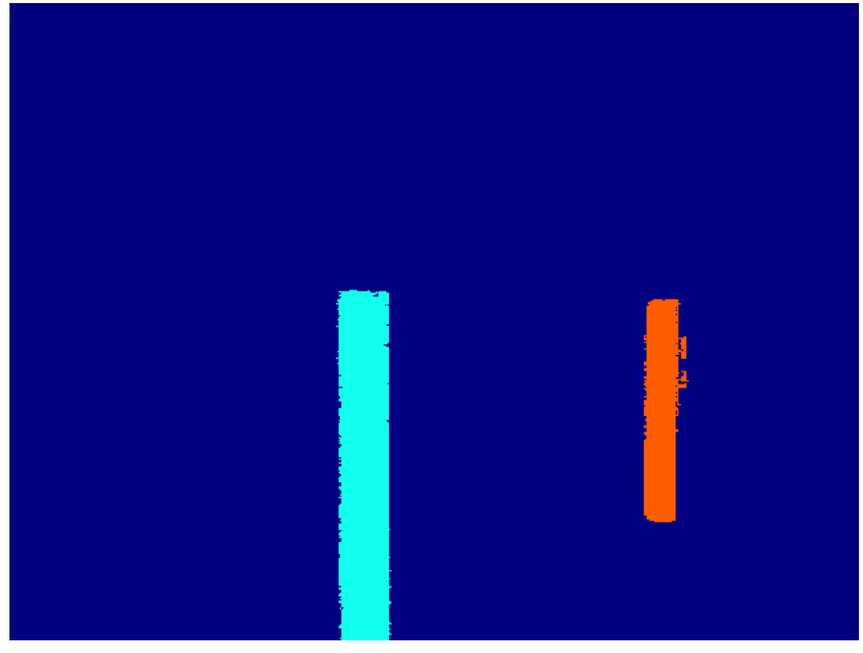




Nalezení sloupků, využití výsledku

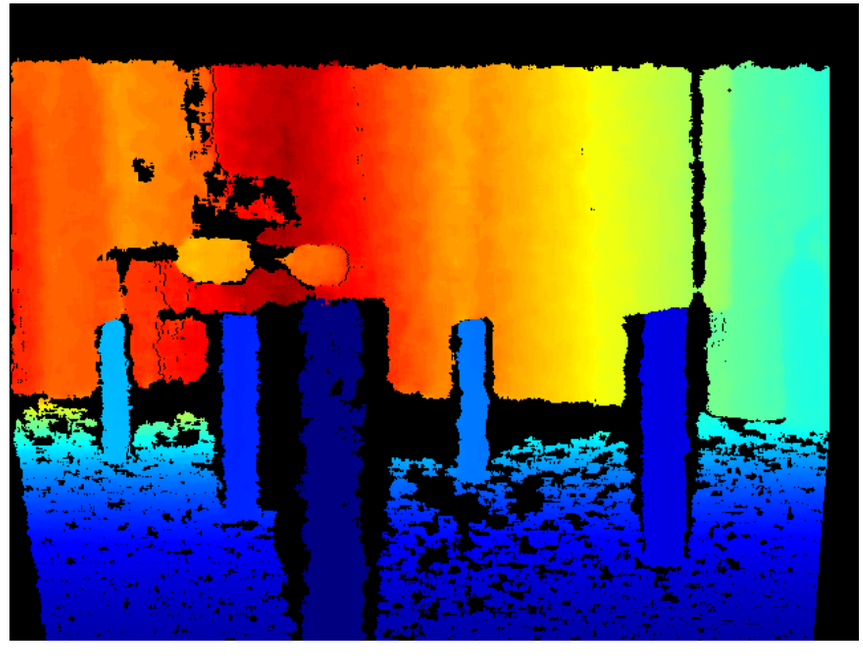
Nalezení sloupků

- Hledané objekty jsou relativně velké.
- Stanovíme podmínku na plochu: $S > t_4$ (500).
- Sloupky jsou v obraze svislé a tenké.
- Stanovíme podmínku na poměr šířky a výšky: $\frac{h}{w} > t_5$ (4).



Využití výsledků

- Kamera míří ze středu robotu vpřed.
- Co vidím ve středu obrazu, tam mířím.
- Vzdálenost sloupku
  - Rozměry lineárně závisí na vzdálenosti.
  - Experimentem je možné vztah najít.



Literatura

Zpracování obrazu

[Sonka1993] Mialn Sonka, Vaclav Hlavac a Roger Boyle. *Image Processing, Analysis and Machine Vision*. ISBN: 978-0-412-45570-4, Springer US, 1993.

[Zanuttigh2016] Pietro Zanuttigh, Giulio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto a Guido Maria Cortelazzo. *Time-of-Flight and Structured Light Depth Cameras: Technology and Applications*. ISBN: 978-3-319-30971-2, Springer, 2016.

Houghova transformace

[DudaHart1972] R.O. Duda a P.E. Hart. *Use of the Hough transformation to detect lines and curves in pictures*. Communications of the ACM, 15(1):11–15, 1972.

[Ballard1981] D.H. Ballard. *Generalizing the Hough transform to detect arbitrary shapes*. Pattern Recognition, ISSN: 0031-3203, 13(2):111 - 122, 1981