

Robotika: Základy vidění

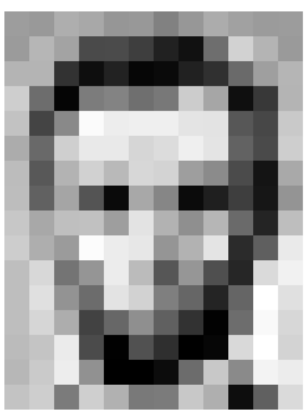
Vladimír Petrík

vladimir.petrik@cvut.cz

12. 10. 2026

Co je to obrázek?

- Kamera připojená k počítači vytváří obrázky
- Obrázek je pole čísel



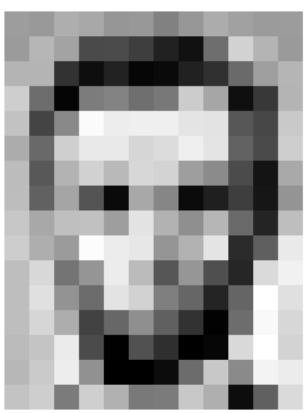
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

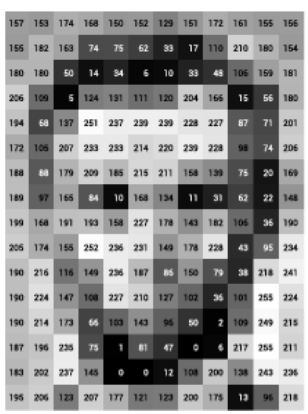
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

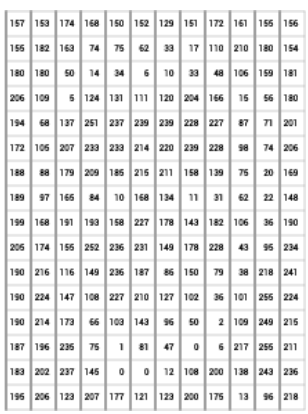
Obrázky jsou z: https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

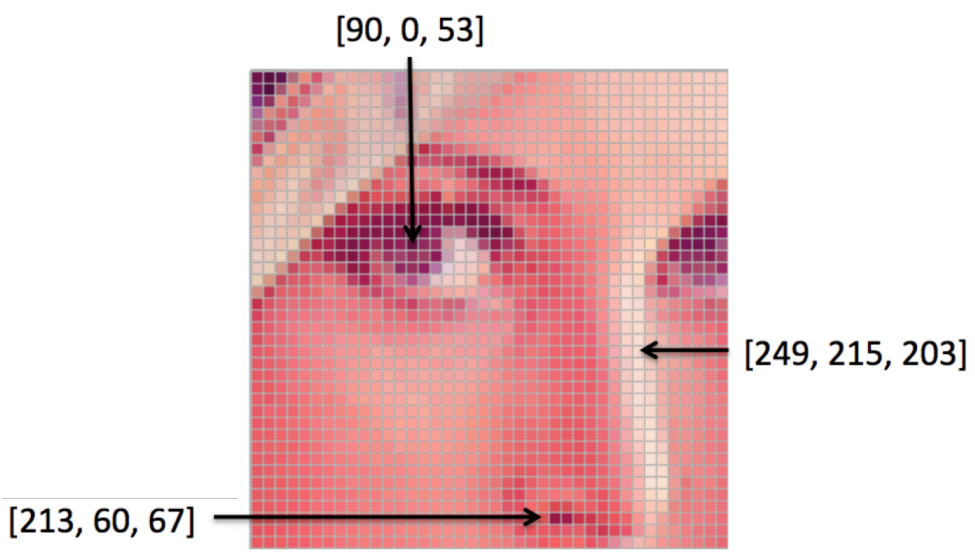
Co je to obrázek?

- Kamera připojená k počítači vytváří obrázky
- Obrázek je pole čísel





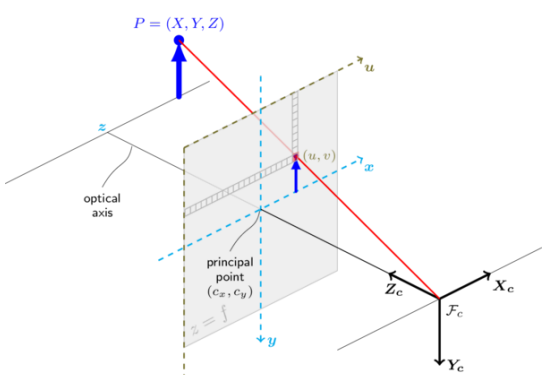




Obrázky jsou z: https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

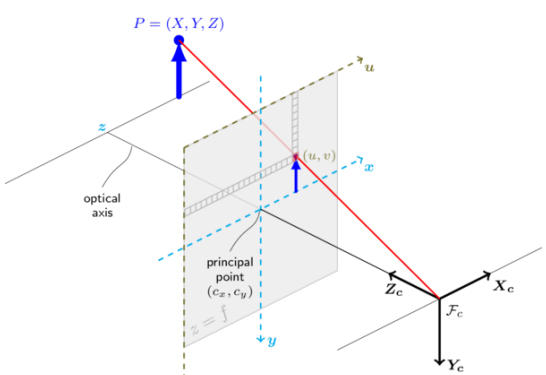
Jak vzniká obraz?

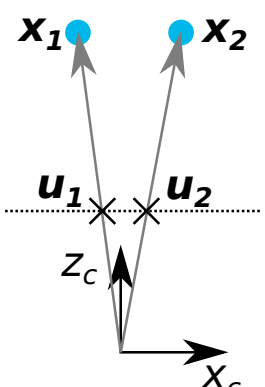
- Perspektivní kamera
  - model dírkové kamery / středové promítání (docs.opencv.org)
  - promítá prostorový bod $\boldsymbol{x}_c$ do obrazového bodu $\boldsymbol{u} = (u, v)^\top$ protnutím
    - obrazové roviny a
    - přímky spojující $\boldsymbol{x}_c$ se středem projekce
  - všechny body na paprsku se promítají do stejného pixelu



Jak vzniká obraz?

- Perspektivní kamera
  - model dírkové kamery / středové promítání (docs.opencv.org)
  - promítá prostorový bod $\boldsymbol{x}_c$ do obrazového bodu $\boldsymbol{u} = (u, v)^\top$ protnutím
    - obrazové roviny a
    - přímky spojující $\boldsymbol{x}_c$ se středem projekce
  - všechny body na paprsku se promítají do stejného pixelu





Projekce

- $\boldsymbol{u}_H = K\boldsymbol{x}_c$
  - $\boldsymbol{u}_H$ je pixel v homogenních souřadnicích
  - pokud $\boldsymbol{u}_H = (u_H, v_H, w_H)^\top$, pak souřadnice pixelu jsou $(u_H/w_H, v_H/w_H)^\top$

Projekce

- $\boldsymbol{u}_H = K\boldsymbol{x}_c$
  - $\boldsymbol{u}_H$ je pixel v homogenních souřadnicích
  - pokud $\boldsymbol{u}_H = (u_H, v_H, w_H)^\top$, pak souřadnice pixelu jsou $(u_H/w_H, v_H/w_H)^\top$
  - alternativně můžeme reprezentovat jako: $\lambda(u, v, 1)^\top = K\boldsymbol{x}_c$

Projekce

- $\boldsymbol{u}_H = K\boldsymbol{x}_c$
  - $\boldsymbol{u}_H$ je pixel v homogenních souřadnicích
  - pokud $\boldsymbol{u}_H = (u_H, v_H, w_H)^\top$, pak souřadnice pixelu jsou $(u_H/w_H, v_H/w_H)^\top$
  - alternativně můžeme reprezentovat jako: $\lambda(u, v, 1)^\top = K\boldsymbol{x}_c$
- K je matice kamery

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

Projekce

- $\boldsymbol{u}_H = K\boldsymbol{x}_c$
  - $\boldsymbol{u}_H$ je pixel v homogenních souřadnicích
  - pokud $\boldsymbol{u}_H = (u_H, v_H, w_H)^\top$, pak souřadnice pixelu jsou $(u_H/w_H, v_H/w_H)^\top$
  - alternativně můžeme reprezentovat jako: $\lambda(u, v, 1)^\top = K\boldsymbol{x}_c$
- K je matice kamery

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

  - co představuje λ?
    - λ je nenulové reálné číslo
    - pokud znáte hodnotu λ, můžete vypočítat kartézské souřadnice $\boldsymbol{x} = \lambda K^{-1}\boldsymbol{u}$
    - v opačném případě lze vypočítat pouze paprsek

Projekce

- $\boldsymbol{u}_H = K\boldsymbol{x}_c$
  - $\boldsymbol{u}_H$ je pixel v homogenních souřadnicích
  - pokud $\boldsymbol{u}_H = (u_H, v_H, w_H)^\top$, pak souřadnice pixelu jsou $(u_H/w_H, v_H/w_H)^\top$
  - alternativně můžeme reprezentovat jako: $\lambda(u, v, 1)^\top = K\boldsymbol{x}_c$
- K je matice kamery

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

  - co představuje λ?
    - λ je nenulové reálné číslo
    - pokud znáte hodnotu λ, můžete vypočítat kartézské souřadnice $\boldsymbol{x} = \lambda K^{-1}\boldsymbol{u}$
    - v opačném případě lze vypočítat pouze paprsek
  - jak zjistit K z bodů?

Co můžeme studovat na obrázcích?



Co můžeme studovat na obrázcích?

- Segmentační masky (kde jsou objekty zájmu)



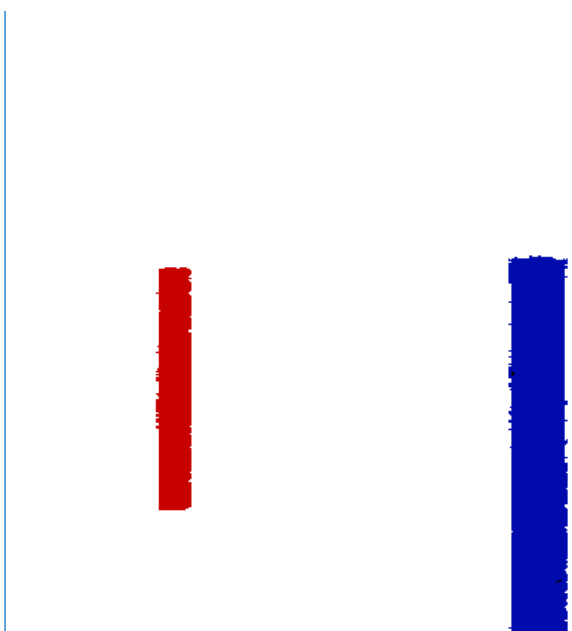


Co můžeme studovat na obrázcích?

- Segmentační masky (kde jsou objekty zájmu)
- Klasifikace objektů (označování)







Segmentační masky – prahování barev

- Prahování
  - hodnoty pixelů RGB pro souřadnice $\boldsymbol{u}$: $I_{\mathrm{RGB}}(\boldsymbol{u})$

Segmentační masky – prahování barev

- Prahování
  - hodnoty pixelů RGB pro souřadnice $\boldsymbol{u}$: $I_{\mathrm{RGB}}(\boldsymbol{u})$
  - $M(\boldsymbol{u}) = 1$, pokud $I_{\mathrm{RGB}}(\boldsymbol{u}) = (0, 255, 0)^\top$?

Segmentační masky – prahování barev

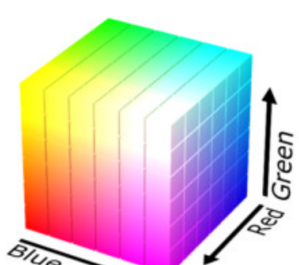
- Prahování
  - hodnoty pixelů RGB pro souřadnice $\boldsymbol{u}$: $I_{\mathrm{RGB}}(\boldsymbol{u})$
  - $M(\boldsymbol{u}) = 1$, pokud $I_{\mathrm{RGB}}(\boldsymbol{u}) = (0, 255, 0)^\top$?
  - $M(\boldsymbol{u}) = 1$, pokud $\boldsymbol{\tau}_l < I_{\mathrm{RGB}}(\boldsymbol{u}) < \boldsymbol{\tau}_u$, pro všechny kanály

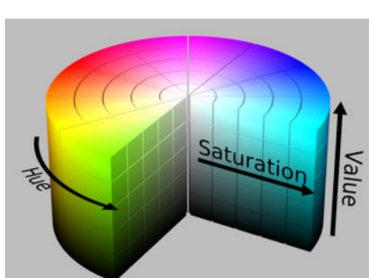


Segmentační masky – prahování barev

- Prahování
  - hodnoty pixelů RGB pro souřadnice $\boldsymbol{u}$: $I_{\mathrm{RGB}}(\boldsymbol{u})$
  - $M(\boldsymbol{u}) = 1$, pokud $I_{\mathrm{RGB}}(\boldsymbol{u}) = (0, 255, 0)^\top$?
  - $M(\boldsymbol{u}) = 1$, pokud $\boldsymbol{\tau}_l < I_{\mathrm{RGB}}(\boldsymbol{u}) < \boldsymbol{\tau}_u$, pro všechny kanály
  - $M(\boldsymbol{u}) = 1$, pokud $\boldsymbol{\varphi}_l < I_{\mathrm{HSV}}(\boldsymbol{u}) < \boldsymbol{\varphi}_u$, pro všechny kanály



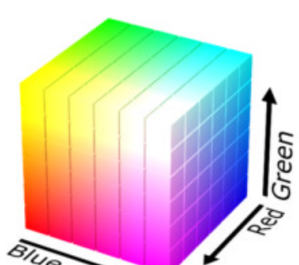


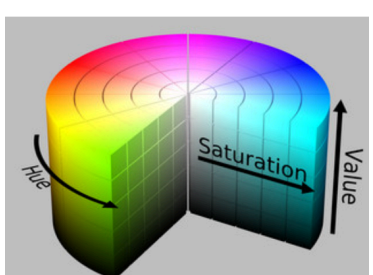


Segmentační masky – prahování barev

- Prahování
  - hodnoty pixelů RGB pro souřadnice $\boldsymbol{u}$: $I_{\mathrm{RGB}}(\boldsymbol{u})$
  - $M(\boldsymbol{u}) = 1$, pokud $I_{\mathrm{RGB}}(\boldsymbol{u}) = (0, 255, 0)^\top$?
  - $M(\boldsymbol{u}) = 1$, pokud $\boldsymbol{\tau}_l < I_{\mathrm{RGB}}(\boldsymbol{u}) < \boldsymbol{\tau}_u$, pro všechny kanály
  - $M(\boldsymbol{u}) = 1$, pokud $\boldsymbol{\varphi}_l < I_{\mathrm{HSV}}(\boldsymbol{u}) < \boldsymbol{\varphi}_u$, pro všechny kanály
- Následné zpracování
  - výpočet spojených komponent
  - odstranit malé nebo deformované segmenty
  - přiřadit označení na základě prahových hodnot







Segmentační masky pro známé 3D objekty

- Neuronová síť (např. Mask R-CNN)

Segmentační masky pro známé 3D objekty

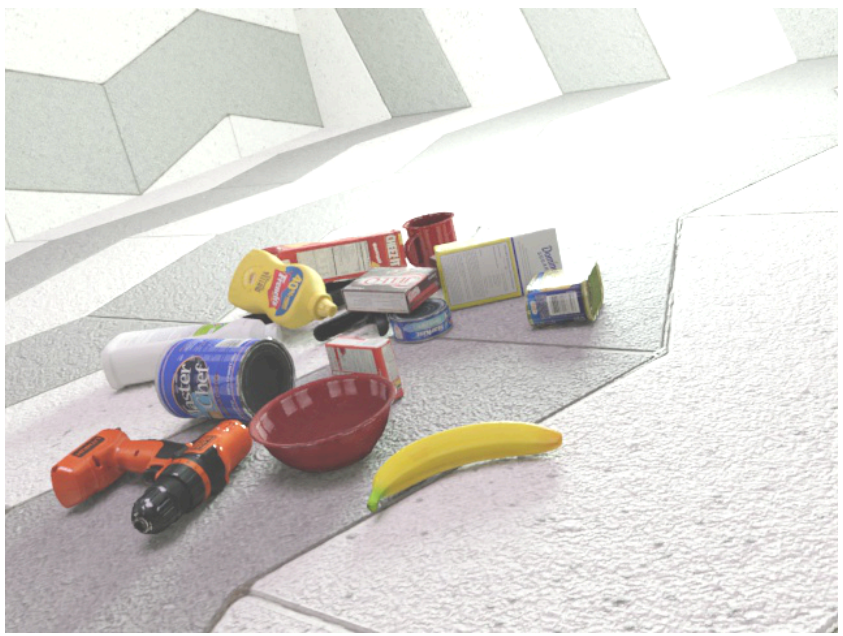
- Neuronová síť (např. Mask R-CNN)
- Trénovací vstupy:
  - datová sada obrázků, masek a štítků nebo

Segmentační masky pro známé 3D objekty

- Neuronová síť (např. Mask R-CNN)
- Trénovací vstupy:
  - datová sada obrázků, masek a štítků nebo
  - datová sada známých 3D objektů (mesh)

Segmentační masky pro známé 3D objekty

- Neuronová síť (např. Mask R-CNN)
- Trénovací vstupy:
  - datová sada obrázků, masek a štítků nebo
  - c





Segmentační masky pro známé 3D objekty

- Neuronová síť (např. Mask R-CNN)
- Trénovací vstupy:
  - datová sada obrázků, masek a štítků nebo
  - c





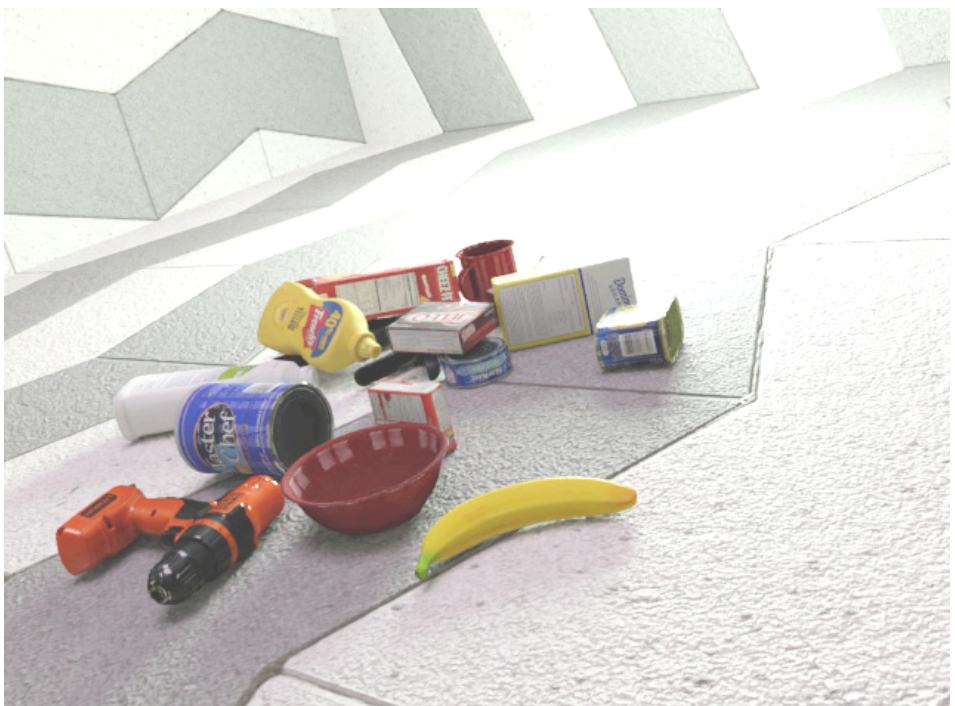
Segmentační masky pro známé 3D objekty

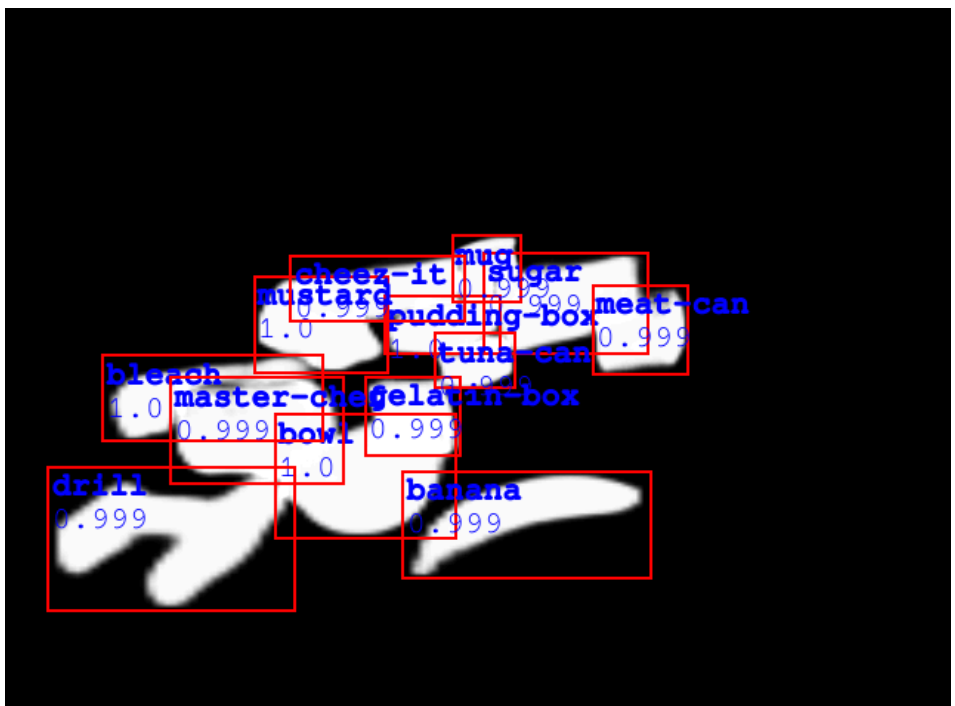
- Neuronová síť (např. Mask R-CNN)
- Trénovací vstupy:
  - datová sada obrázků, masek a štítků nebo
  - datová sada známých 3D objektů (mesh)
  - kvalita závisí na trénovacích datech (augmentace)

Segmentační masky pro známé 3D objekty

- Neuronová síť (např. Mask R-CNN)
- Trénovací vstupy:
  - datová sada obrázků, masek a štítků nebo
  - datová sada známých 3D objektů (mesh)
  - kvalita závisí na trénovacích datech (augmentace)
- Inference:
  - Vstup: obrázek
  - Výstup: segmentační maska, ohraničující rámeček, označení a konfidence

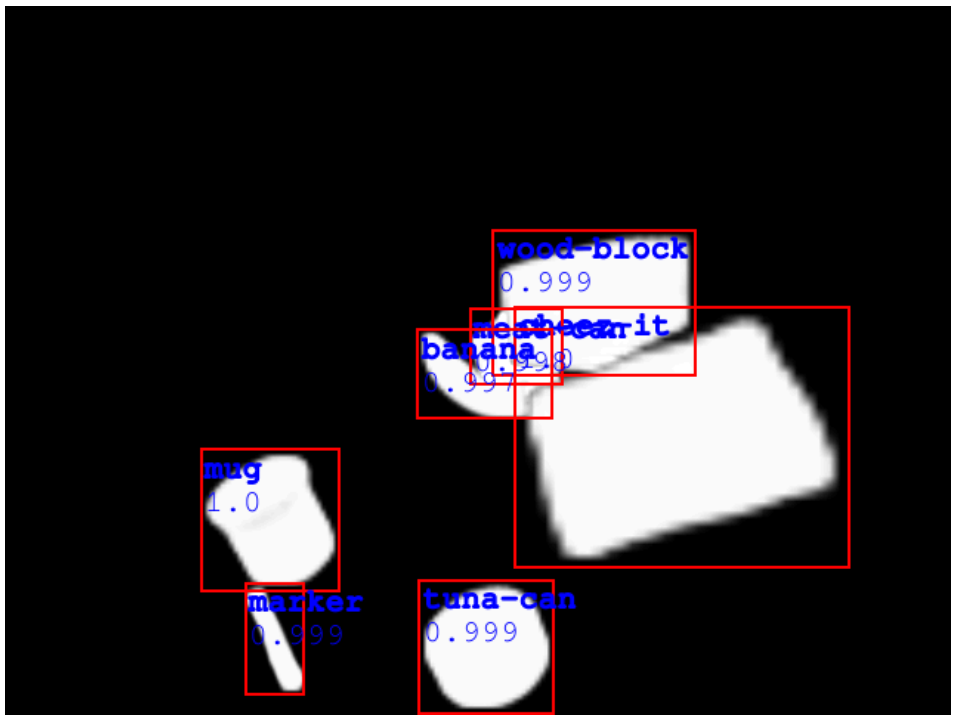
Mask R-CNN výsledky





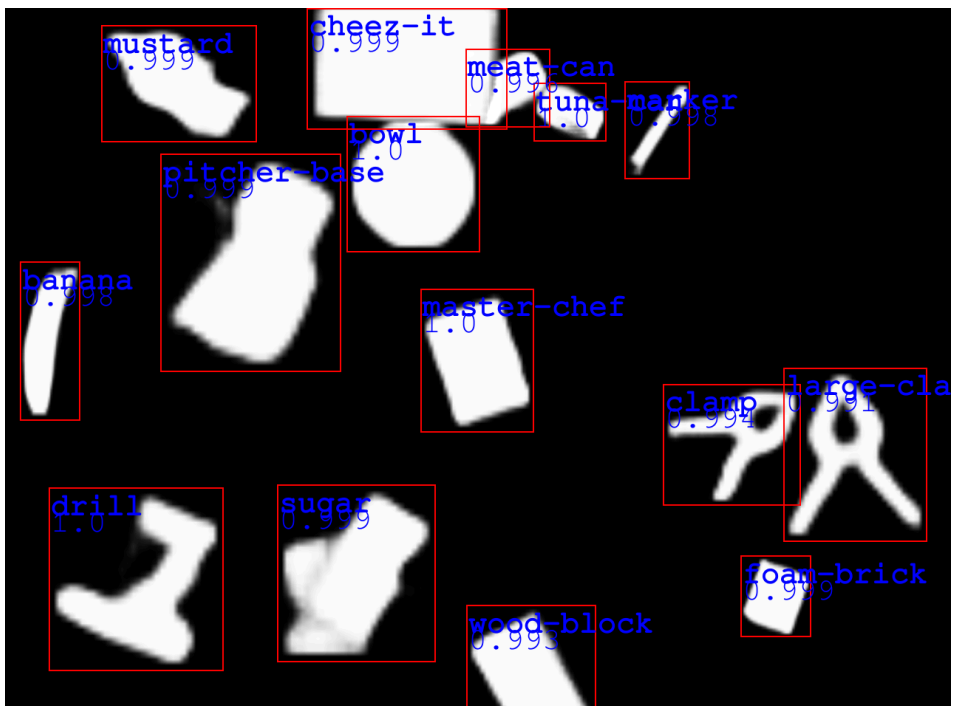
Mask R-CNN výsledky





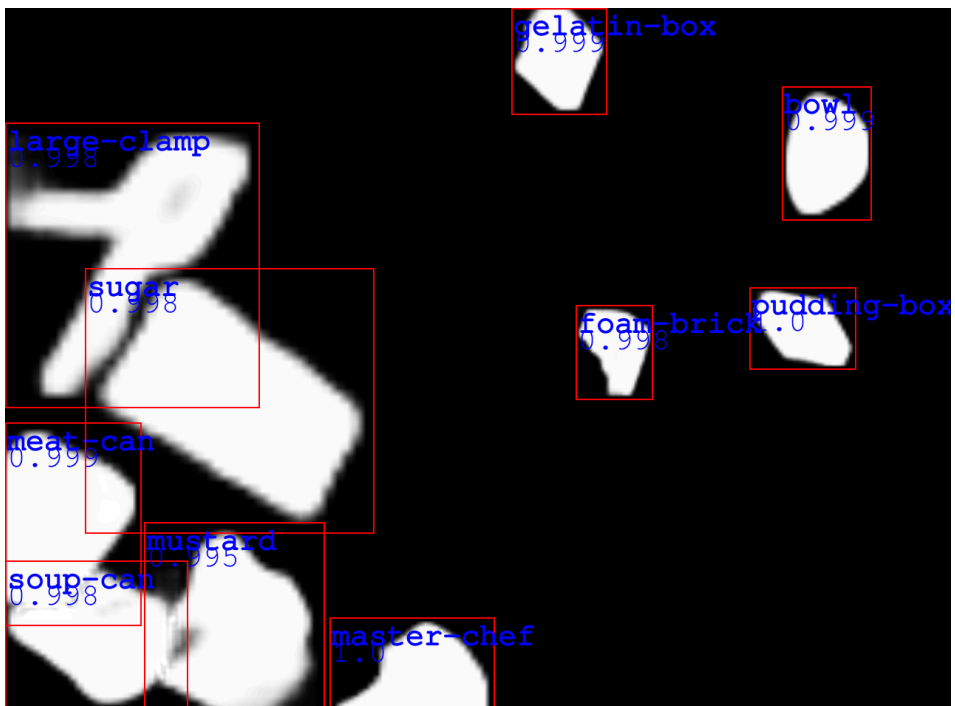
Mask R-CNN výsledky





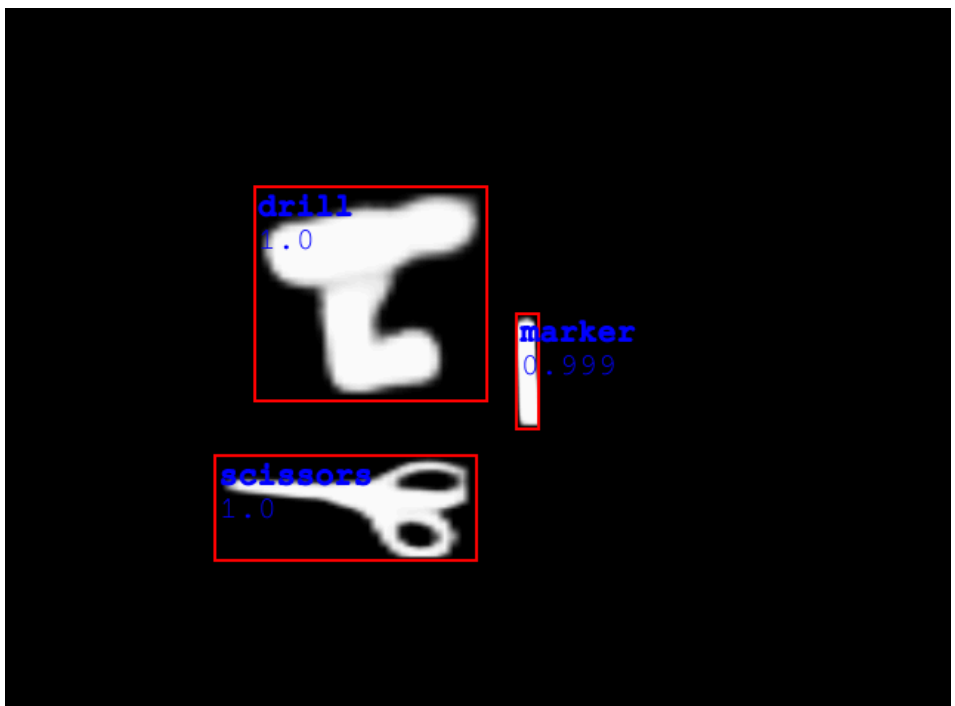
Mask R-CNN výsledky





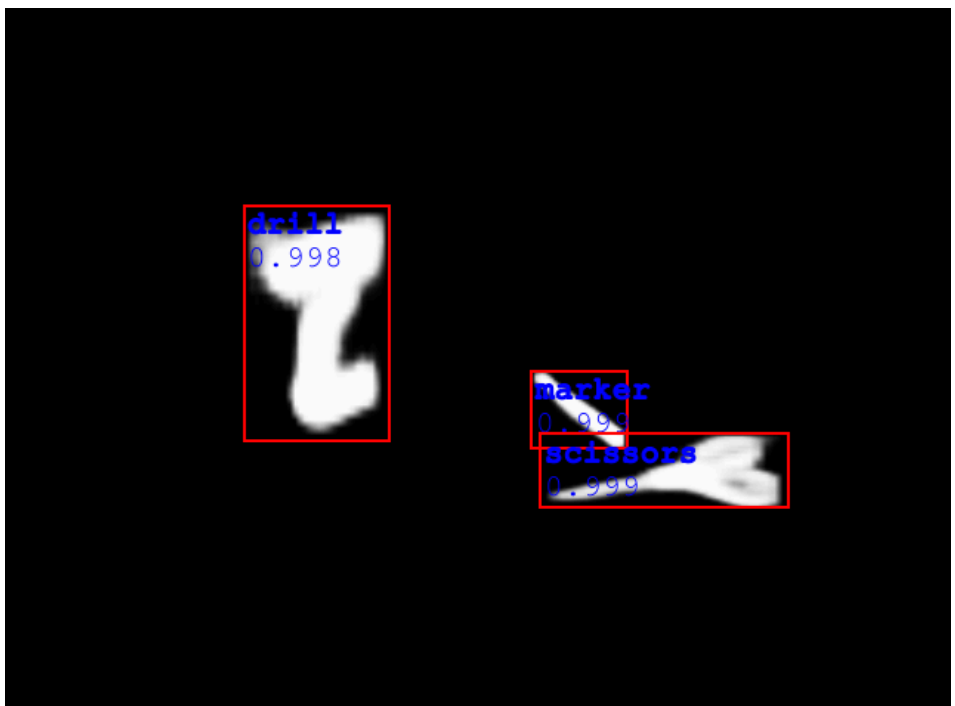
Mask R-CNN výsledky





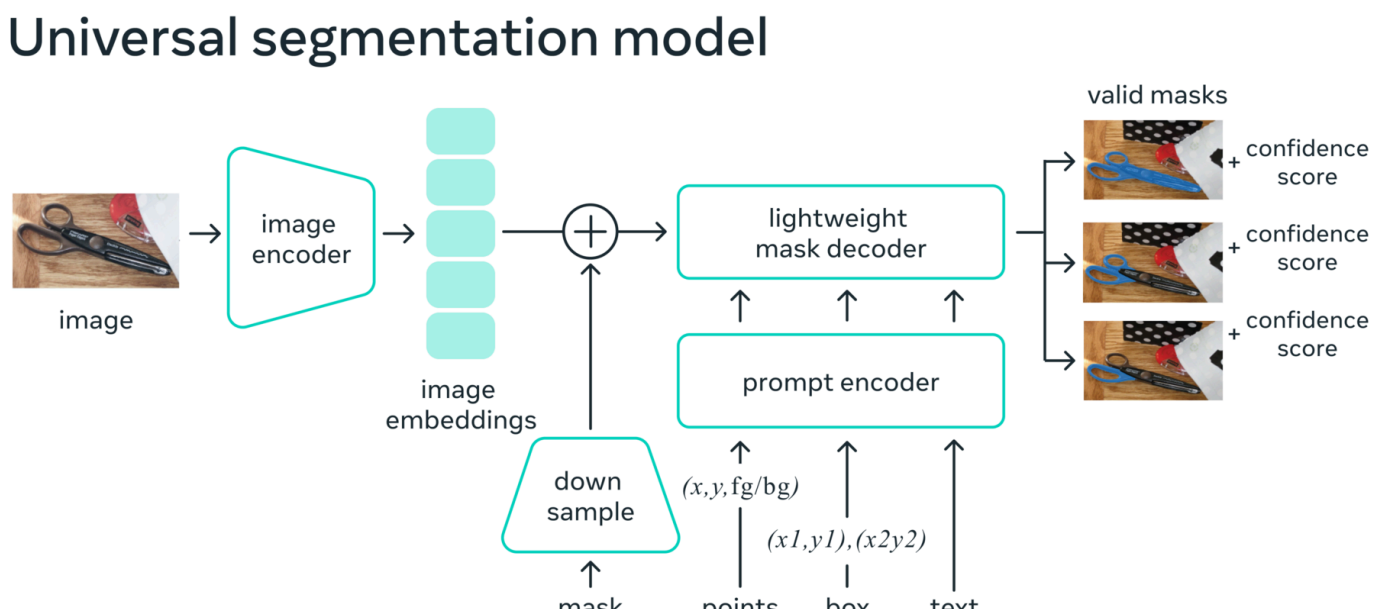
Mask R-CNN výsledky





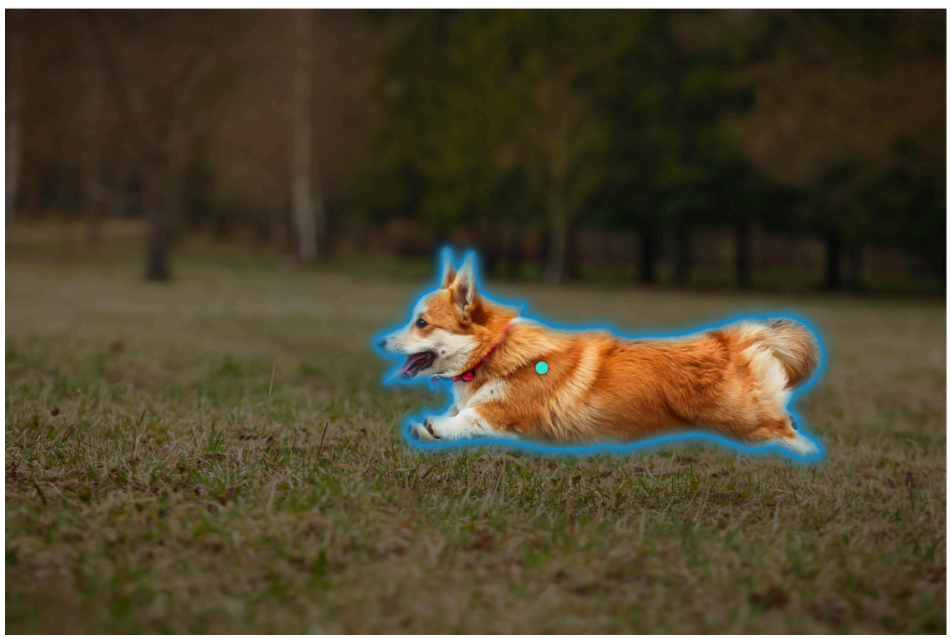
Segmentační masky bez přetrénování

- Segment Anything Model (SAM)
  - segmentace libovolného objektu na libovolném snímku jediným kliknutím
  - datová sada 10 milionů obrázků, 1 miliarda masek



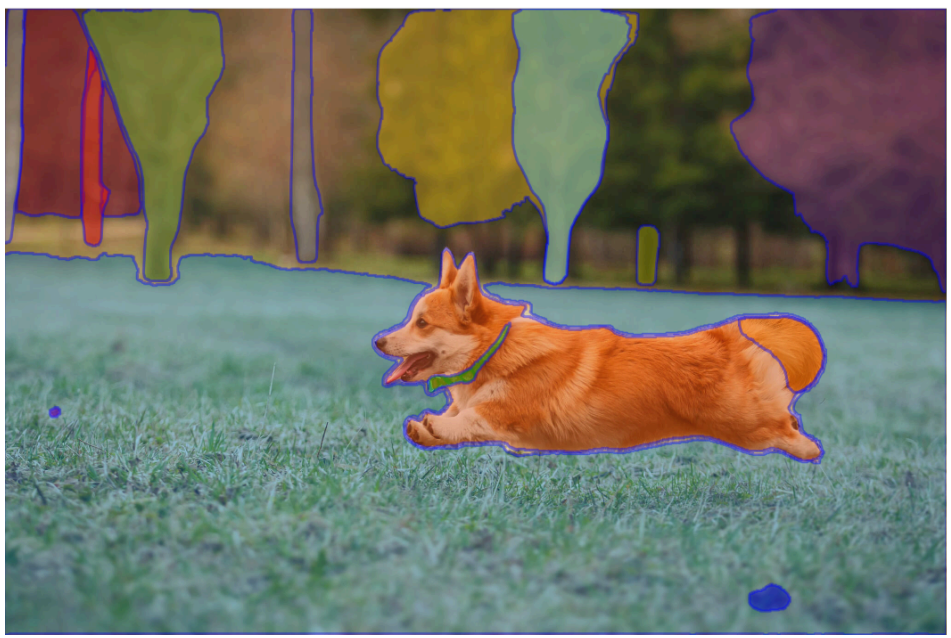
SAM výsledky





SAM výsledky





SAM výsledky



SAM výsledky



SAMv2



Segmentace

- Segmentace nalezne objekty v obraze
  - segmentační maska
  - bounding box (ohraničující rámeček)
  - label (označení)
  - skóre důvěryhodnosti / konfidence

Segmentace

- Segmentace nalezne objekty v obraze
  - segmentační maska
  - bounding box (ohraničující rámeček)
  - label (označení)
  - skóre důvěryhodnosti / konfidence
- Informace pouze v obrazovém prostoru
- Jak ji použít v prostoru robota?

Externí kamera

- Předpokládáme kameru pevně připevněnou k referenčnímu s.s.
  - pokud známe K a T_{RC}, jak promítnout body $\boldsymbol{x}_R$ do obrazu?

Externí kamera

- Předpokládáme kameru pevně připevněnou k referenčnímu s.s.
  - pokud známe K a T_{RC}, jak promítnout body $\boldsymbol{x}_R$ do obrazu?
- Neznámé K a T_{RC} a planární problém
  - např. kostky se stejnou výškou na stole
  - jaká je poloha kostky na 2D stole vzhledem k 2D souřadnicím obrazu/pixelů?

Externí kamera

- Předpokládáme kameru pevně připevněnou k referenčnímu s.s.
  - pokud známe K a T_{RC}, jak promítnout body $\boldsymbol{x}_R$ do obrazu?
- Neznámé K a T_{RC} a planární problém
  - např. kostky se stejnou výškou na stole
  - jaká je poloha kostky na 2D stole vzhledem k 2D souřadnicím obrazu/pixelů?
  - analyzováno pomocí **homografie**

Homografie

- Homografická matice H je matice 3×3, která mapuje body z jedné roviny do druhé
  - obrazová rovina na stůl
  - jedna obrazová rovina do jiné obrazové roviny (jiný pohled)

Homografie

- Homografická matice H je matice 3×3, která mapuje body z jedné roviny do druhé
  - obrazová rovina na stůl
  - jedna obrazová rovina do jiné obrazové roviny (jiný pohled)
- $s(x, y, 1)^\top = H(u, v, 1)^\top$
  - x, y jsou souřadnice v první rovině
  - u, v jsou souřadnice ve druhé rovině

Homografie

- Homografická matice H je matice 3×3, která mapuje body z jedné roviny do druhé
  - obrazová rovina na stůl
  - jedna obrazová rovina do jiné obrazové roviny (jiný pohled)
- $s(x, y, 1)^\top = H(u, v, 1)^\top$
  - x, y jsou souřadnice v první rovině
  - u, v jsou souřadnice ve druhé rovině
- 9 prvků, ale pouze 8 DoF, obvykle se přidává omezení $h_{33} = 1$
- Jak najít H?

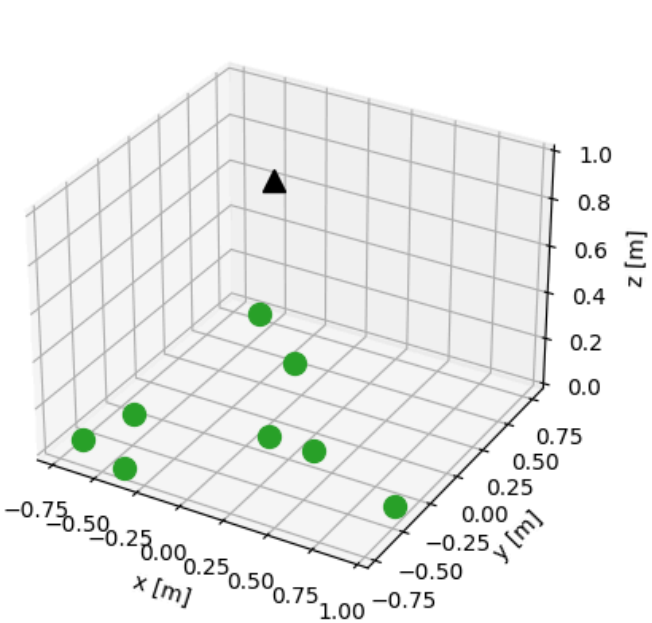
Homografie

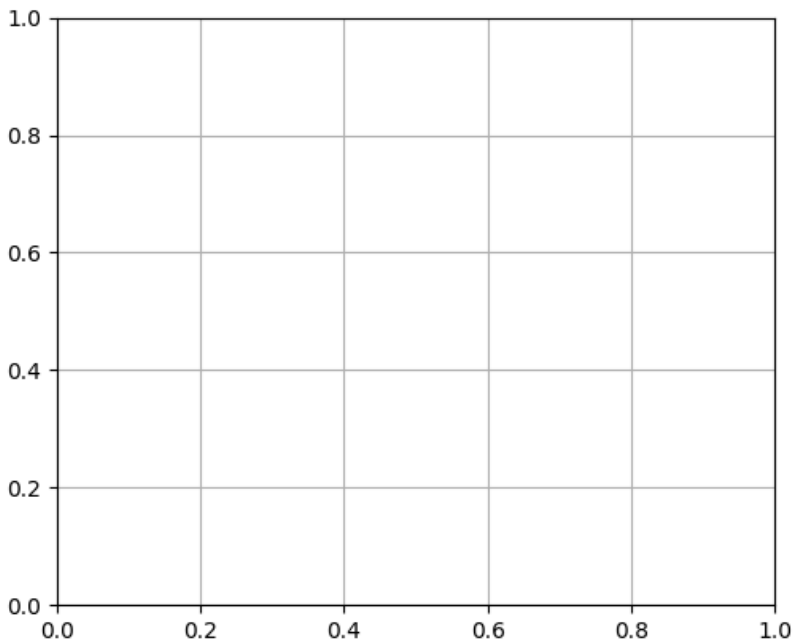
- Homografická matice H je matice 3×3, která mapuje body z jedné roviny do druhé
  - obrazová rovina na stůl
  - jedna obrazová rovina do jiné obrazové roviny (jiný pohled)
- $s(x, y, 1)^\top = H(u, v, 1)^\top$
  - x, y jsou souřadnice v první rovině
  - u, v jsou souřadnice ve druhé rovině
- 9 prvků, ale pouze 8 DoF, obvykle se přidává omezení $h_{33} = 1$
- Jak najít H?
  - `H, _ = cv2.findHomography(U, X)`
  - U, X jsou $N \times 2$ korespondenční body

Homografie

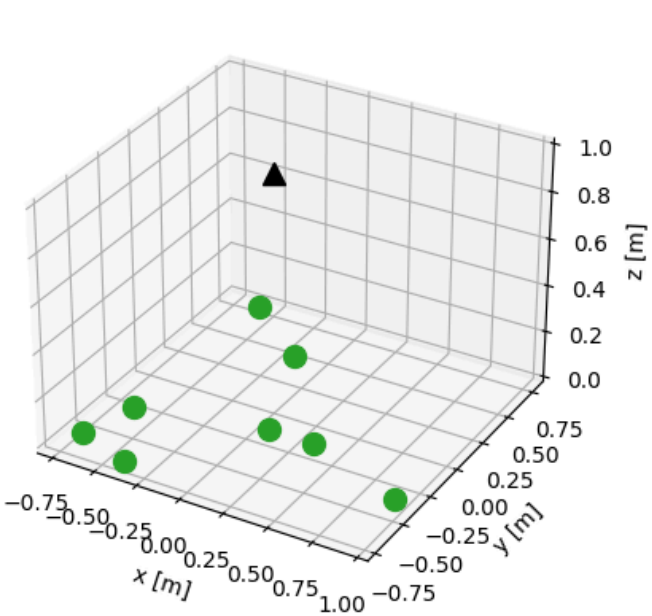
- Homografická matice H je matice 3×3, která mapuje body z jedné roviny do druhé
  - obrazová rovina na stůl
  - jedna obrazová rovina do jiné obrazové roviny (jiný pohled)
- $s(x, y, 1)^\top = H(u, v, 1)^\top$
  - x, y jsou souřadnice v první rovině
  - u, v jsou souřadnice ve druhé rovině
- 9 prvků, ale pouze 8 DoF, obvykle se přidává omezení $h_{33} = 1$
- Jak najít H?
  - `H, _ = cv2.findHomography(U, X)`
  - U, X jsou $N \times 2$ korespondenční body
  - např. ručně změříme
    - poloha středu krychle vůči rohu stolu
    - pozice středu krychle v obrázku

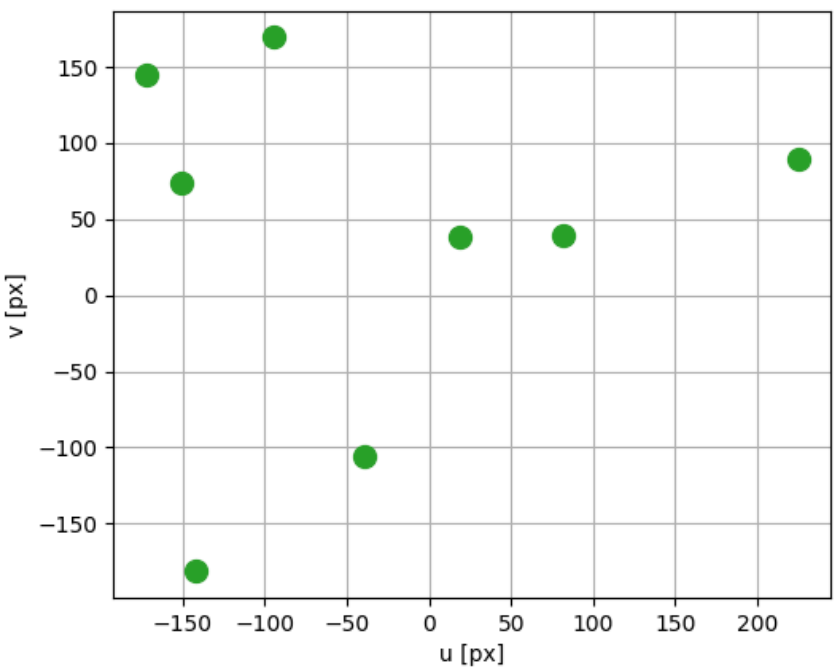
Příklad homografie



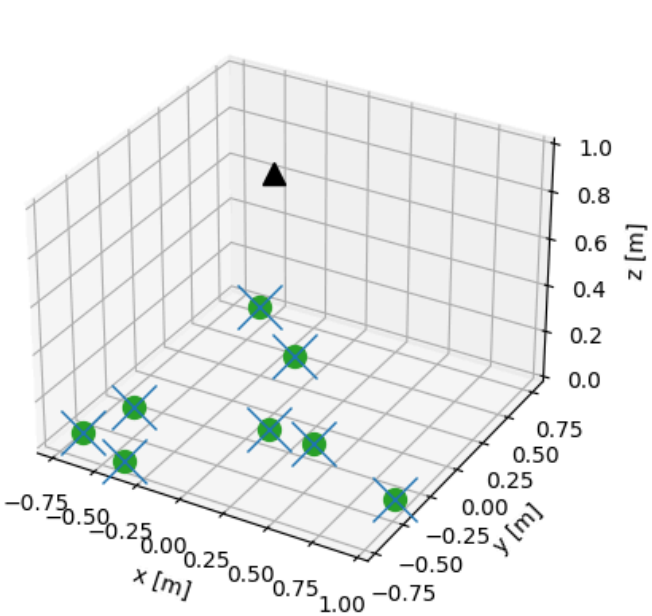


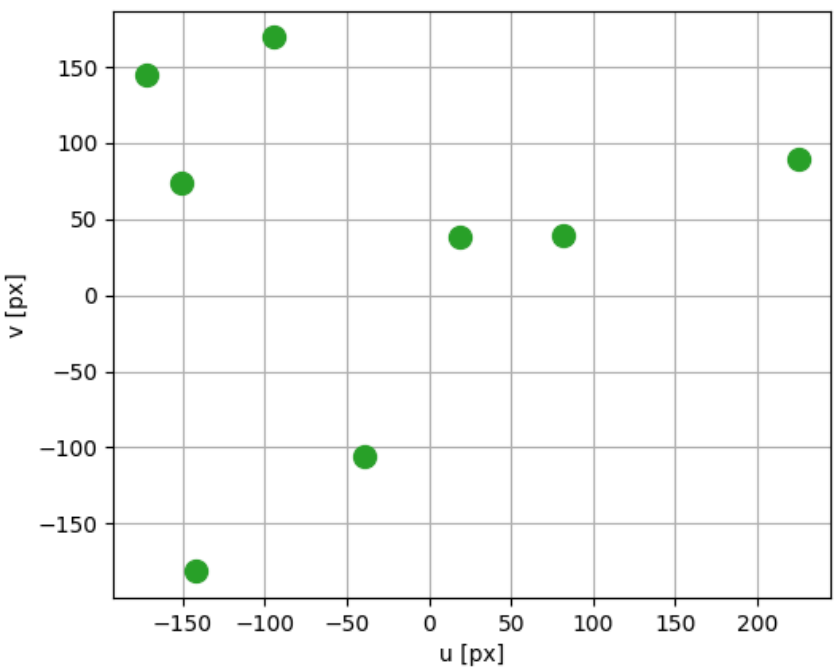
Příklad homografie



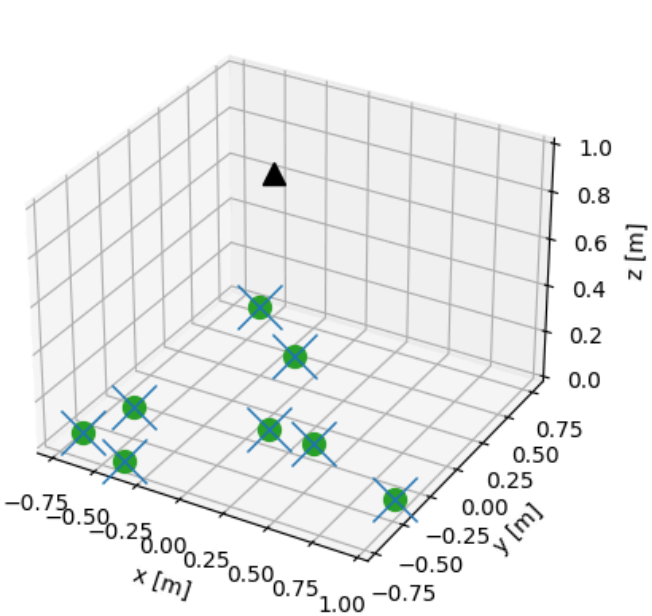


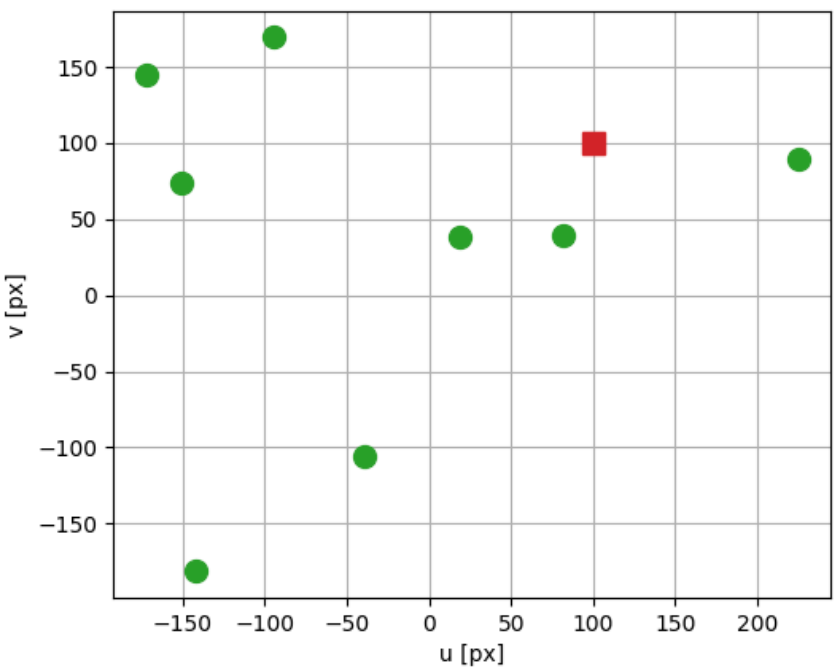
Příklad homografie



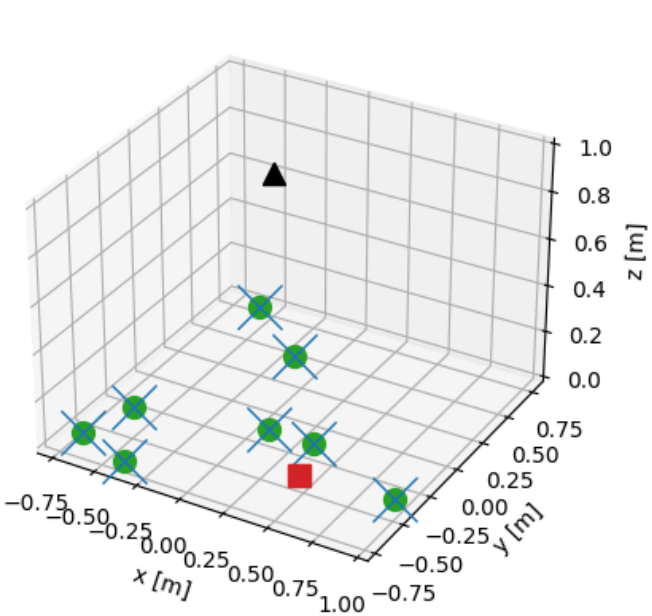


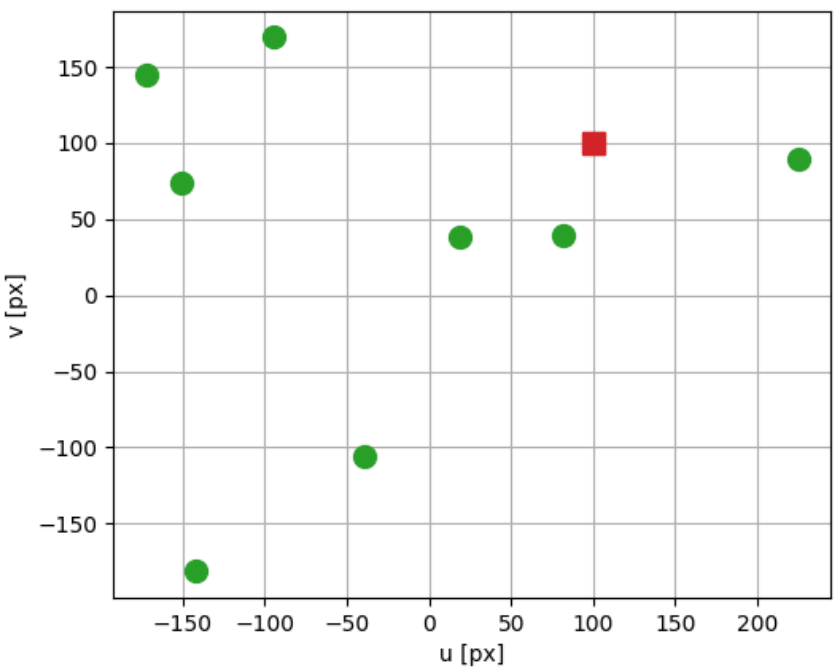
Příklad homografie





Příklad homografie



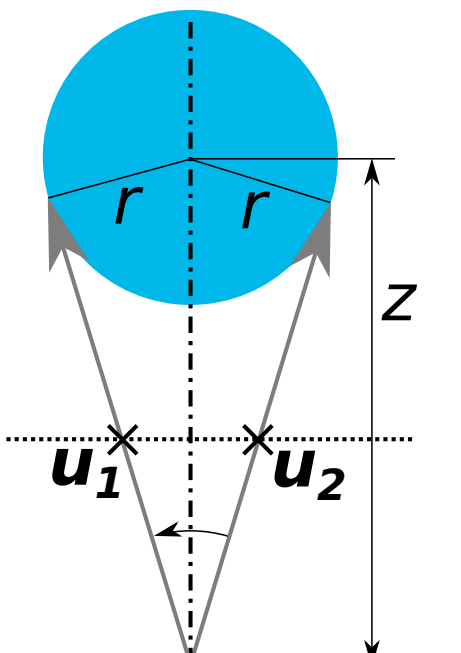


Neplanární odhad polohy/orientace

- Homografie mapuje pouze rovinu na rovinu
- Obecnější odhad polohy objektu v s.s. **kamery**
  - získání hloubky mapováním z plochy v pixelech na hloubku pro objekty pevné velikosti
  - získání hloubky pomocí dodatečných informací o scéně, např. známá velikost/model objektu
  - RGBD kamera
  - dodatečné značky (markery)

Využití znalosti o velikosti

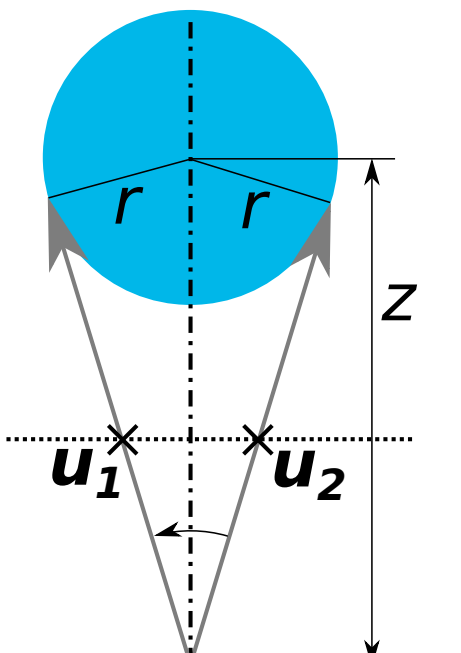
- Víme, že poloměr r je pevný



Využití znalosti o velikosti

- Víme, že poloměr r je pevný
- Z detekovaných pixelů u_1, u_2 můžeme vypočítat paprsky x_1, x_2:

$$\frac{1}{\lambda_i} x_i = K^{-1} u_i$$



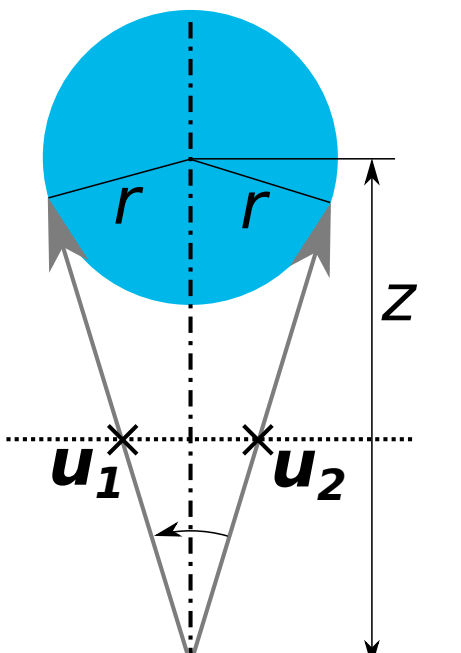
Využití znalosti o velikosti

- Víme, že poloměr r je pevný
- Z detekovaných pixelů u_1, u_2 můžeme vypočítat paprsky x_1, x_2:

$$\frac{1}{\lambda_i} x_i = K^{-1} u_i$$

- Úhel mezi vektory (nezávislý na λ_1, λ_2):

$$\cos \alpha = \frac{x_1 \cdot x_2}{\|x_1\| \|x_2\|}$$



Využití znalosti o velikosti

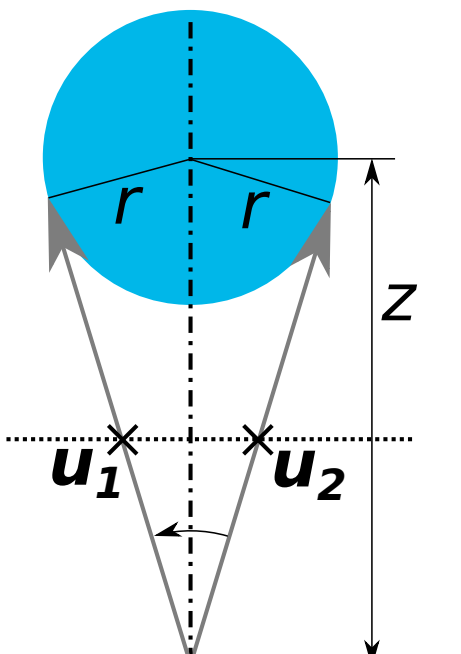
- Víme, že poloměr r je pevný
- Z detekovaných pixelů u_1, u_2 můžeme vypočítat paprsky x_1, x_2:

$$\frac{1}{\lambda_i} x_i = K^{-1} u_i$$

- Úhel mezi vektory (nezávislý na λ_1, λ_2):

$$\cos \alpha = \frac{x_1 \cdot x_2}{\|x_1\| \|x_2\|}$$

- Hloubka: $z = \frac{r}{\sin(\alpha/2)}$



Použití hloubkového senzoru

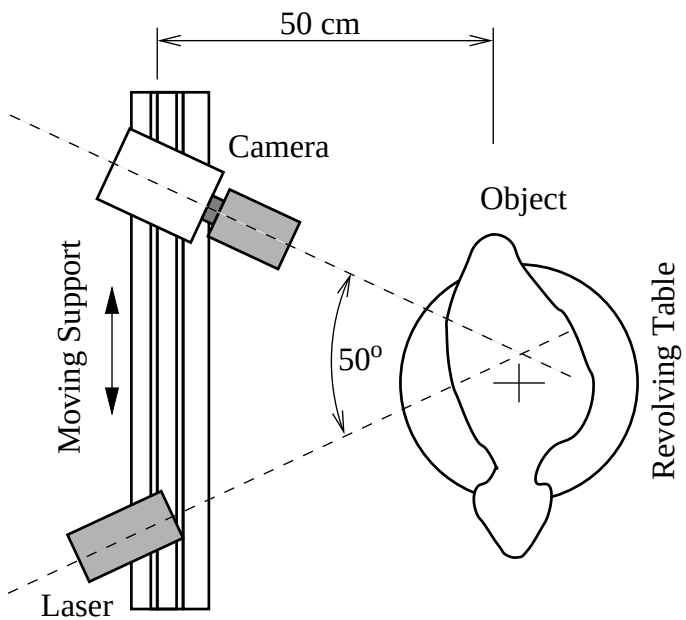
- RGB**D** senzory
  - RGB obrázek ($H \times W \times 3$)
  - Hloubková mapa ($H \times W \times 1$), vzdálenost v metrech pro každý pixel
  - Strukturované mračno bodů ($H \times W \times 3$), (x_c, y_c, z_c) pro každý pixel

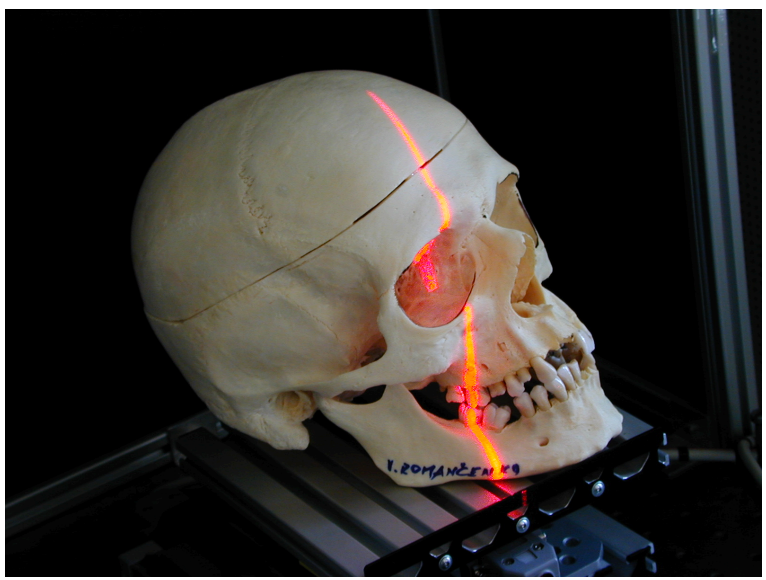




Jak funguje hloubkový senzor

- Laser promítá vzor a kamera ho rozpozná
- Informace o hloubce se vypočítá pomocí triangulace





2D hloubkové senzory

- Založené na strukturovaném světle
- Promítají 2D infračervené vzory
- Jeden projektor a dvě kamery (RGB + IR)

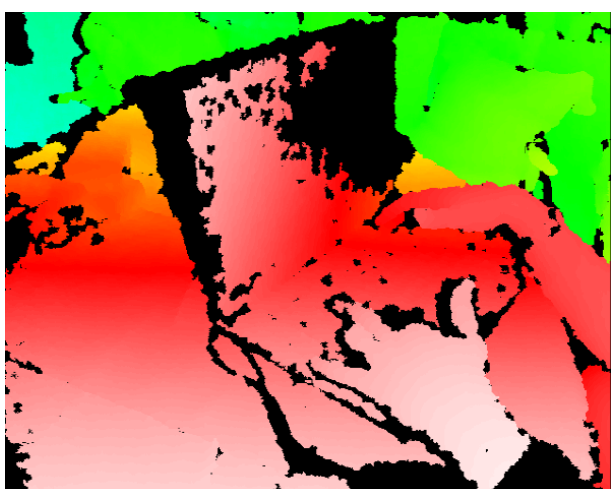




Problémy se snímači hloubky

- Rekonstrukce hloubky není dokonalá (černé oblasti na snímku)
- V Pythonu reprezentováno hodnotou NaN
- Ne každý pixel v RGB má rekonstruovanou hodnotu hloubky
- Data RGB a hloubky nejsou zarovnána (je třeba je kalibrovat)

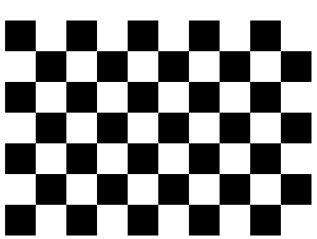


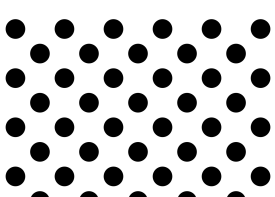


commons.wikimedia.org, User:Kolossos

Dodatečné značky

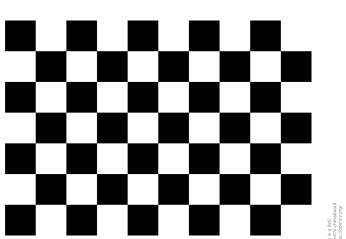
- Můžeme vypočítat polohu/orientaci obrazců? (docs.opencv.org)

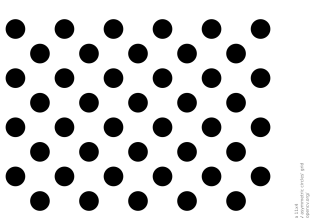


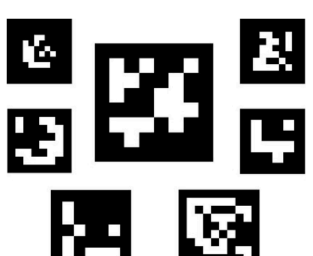


Dodatečné značky

- Můžeme vypočítat polohu/orientaci obrazců? (docs.opencv.org)
  - je třeba znát velikost a strukturu
  - subpixelová přesnost
  - musí být zcela viditelný
- Můžeme vypočítat polohu značek ArUco?

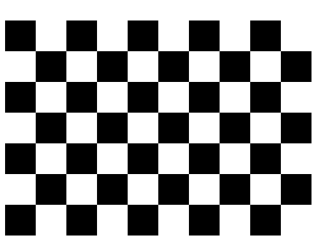


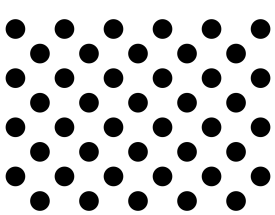


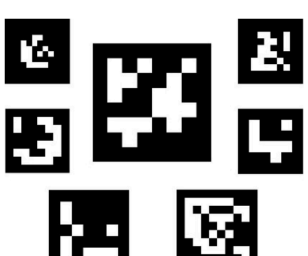


Dodatečné značky

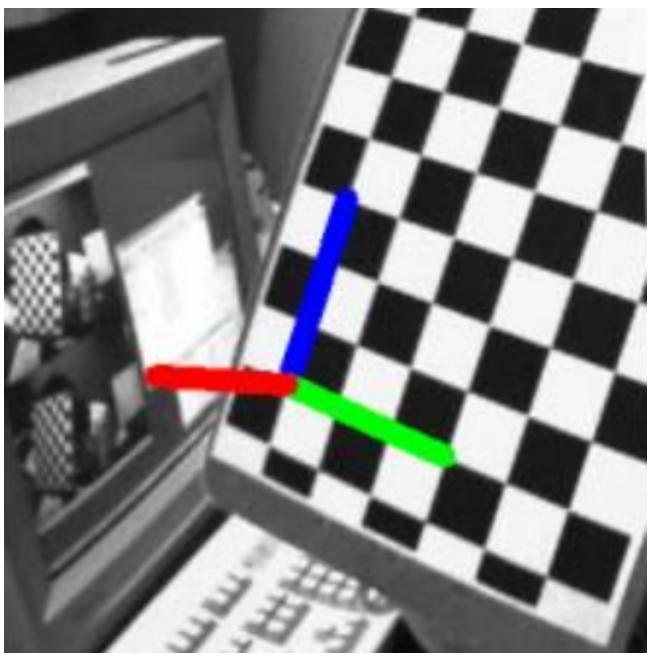
- Můžeme vypočítat polohu/orientaci obrazců? (docs.opencv.org)
  - je třeba znát velikost a strukturu
  - subpixelová přesnost
  - musí být zcela viditelný
- Můžeme vypočítat polohu značek ArUco?
  - méně přesné než pravidelné obrazce
  - poskytují identifikátor značky a pózu
  - musí být zcela viditelný

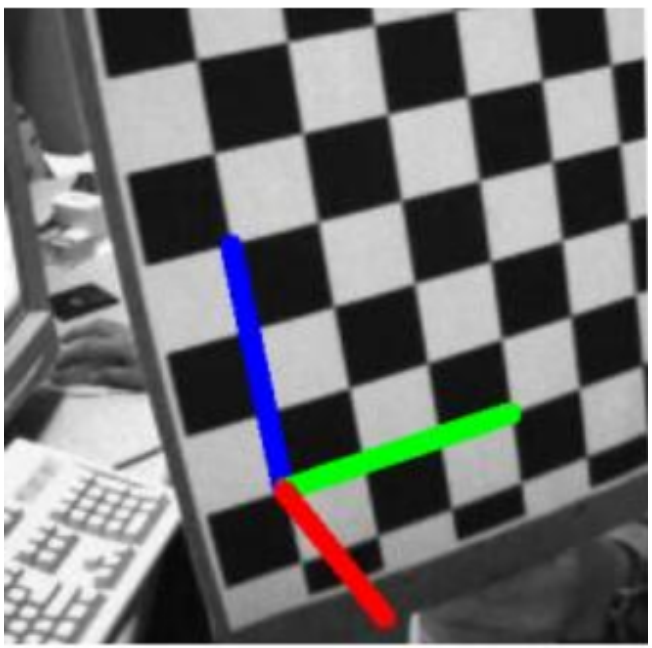


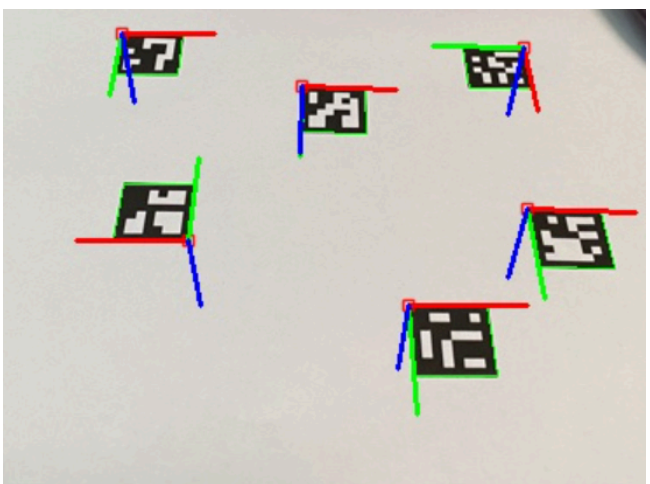




Příklad odhadu polohy značek





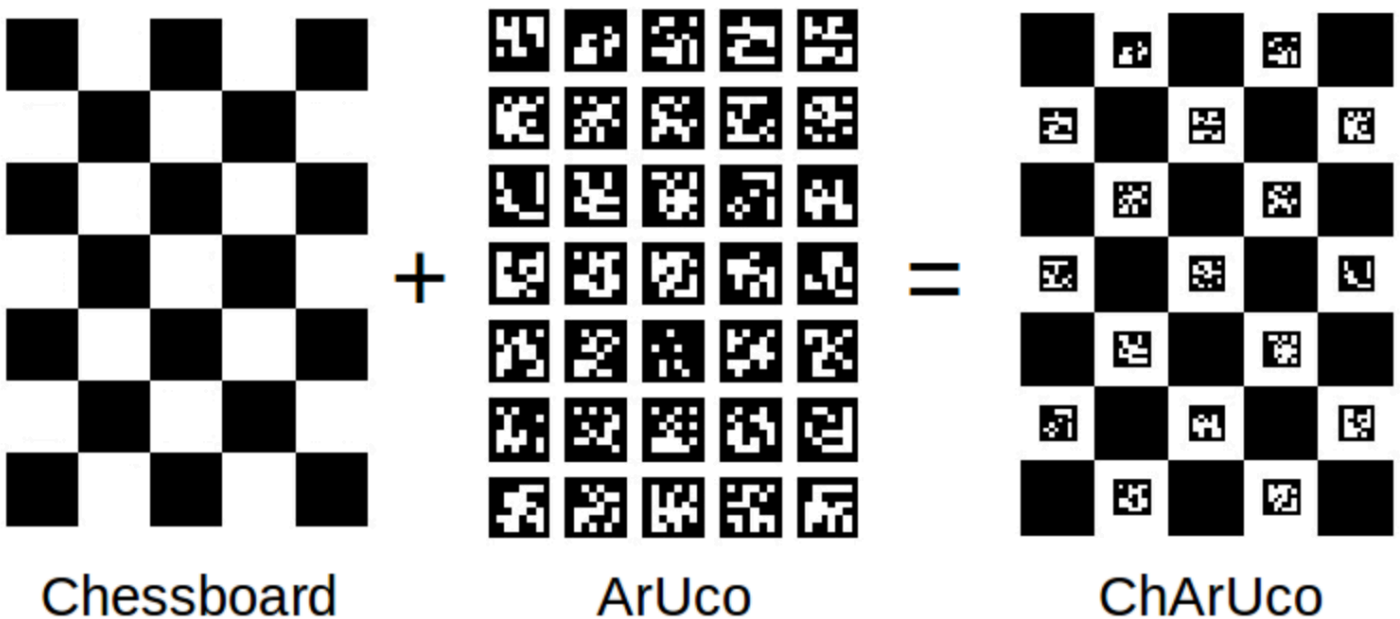


Příklad značek z reálného světa



ChArUco deska pro kalibraci

- Kombinuje přesnost pravidelného vzoru s detekcí ArUco
- Detekce při částečné viditelnosti



Charuco definition

Odhad matice kamery pomocí kalibračních desek

- Můžeme odhadnout matici kamery z korespondencí mezi obrazovým a prostorovým prostorem
  - nasbíráme snímky desky z různých pohledů
  - detekujeme desky
  - vypočítáme korespondence mezi obrazovými body a body v s.s. desky
  - `_, K, dist_coeffs, rvecs, tvecs = cv2.calibrateCamera(obj_points, img_points, img_shape)`
- Kromě toho získáme
  - koeficienty zkreslení, které kompenzují vady objektivu
  - ```
    Knew, roi = cv2.getOptimalNewCameraMatrix(K, dist_coeffs, img_shape, 1, img_shape)
    img_undistorted = cv2.undistort(img, K, dist_coeffs, None, Knew)
    ```
  - $SE(3)$ pozice desek v s.s. kamery
    ```

Odhad pozice z RGB(D)

- Metody odhadu polohy
  - používají předchozí znalosti o úloze, např. objekty s pevnou výškou v rovině
  - používají předchozí znalosti o objektech (velikost)
  - používají hloubkový senzor
  - používají značky ArUco

Odhad pozice z RGB(D)

- Metody odhadu polohy
  - používají předchozí znalosti o úloze, např. objekty s pevnou výškou v rovině
  - používají předchozí znalosti o objektech (velikost)
  - používají hloubkový senzor
  - používají značky ArUco
- Kde je robot?

Odhad pozice z RGB(D)

- Metody odhadu polohy
  - používají předchozí znalosti o úloze, např. objekty s pevnou výškou v rovině
  - používají předchozí znalosti o objektech (velikost)
  - používají hloubkový senzor
  - používají značky ArUco
- Kde je robot?
  - homografie odhaduje polohy objektů vzhledem k s.s. roviny

Odhad pozice z RGB(D)

- Metody odhadu polohy
  - používají předchozí znalosti o úloze, např. objekty s pevnou výškou v rovině
  - používají předchozí znalosti o objektech (velikost)
  - používají hloubkový senzor
  - používají značky ArUco
- Kde je robot?
  - homografie odhaduje polohy objektů vzhledem k s.s. roviny
  - ostatní metody odhadují polohy v s.s. kamery

Odhad pozice z RGB(D)

- Metody odhadu polohy
  - používají předchozí znalosti o úloze, např. objekty s pevnou výškou v rovině
  - používají předchozí znalosti o objektech (velikost)
  - používají hloubkový senzor
  - používají značky ArUco
- Kde je robot?
  - homografie odhaduje polohy objektů vzhledem k s.s. roviny
  - ostatní metody odhadují polohy v s.s. kamery
  - potřebujeme odhadnout/kalibrovat T_{RC}

Shrnutí

- Reprezentace obrázku
- Promítání do/z obrazu
- Segmentace v obrazovém prostoru
- Homografie
- Odhad polohy z obrazu
- Kalibrace kamery

Cvičení

- Transformace kamera–robot, homografie (hw03, odevzdat do 25. 10.)
- Zadání semestrální práce: viz stránka předmětu

Co dál?

- Za týden začneme testem
  - otázky z prvních tří přednášek
  - transformace $SE(2)$ a $SE(3)$ a jejich vlastnosti
  - přímá kinematika a DoF