

# Pokročilé spracovanie obrazu - Kalibrácia a histogramy

Ing. Viktor Kocur  
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

18.10.2018

# Postup kalibrácie

## Kalibračný vzor

Na kalibráciu v matlabe používame kalibračný vzor šachovnice. Dajú sa použiť aj iné vzory, je však nutné naimplementovať rozpoznávanie vzoru samostatne. Vzor musí byť na fotke celý!

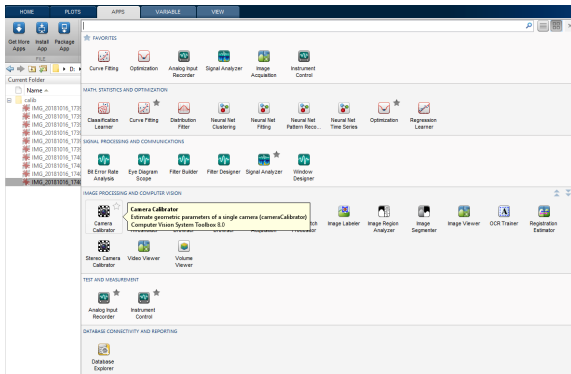
## Fotenie

Pri fotení sa nesmie meniť konfigurácia kamery. Teda je nutné vypnúť autofocus a nezoomovať. Takéto zmeny totiž menia ohniskovú vzdialonť zariadenia!

# Kalibráčná appka

## cameraParams

Pomocou appky v matlabe získame štruktúru cameraParams obsahujúcu parametre kamery.



# Úprava obrazu

## undistort

`[im, newOrigin] = undistortImage(I, cameraParams)` - vráti obrázok bez skreslenia. `newOrigin` označuje posunutie súradníc obrazu oproti pôvodnému obrazu. V prípade, že je to `[0,0]` ho môžeme ignorovať.

## Úloha

Zobrazte si neskreslený obraz pre nejaký snímok. Otestujte aj príkaz `undistortImage(I, cameraParams, 'OutputView', 'full')`.

# Detekcia vzoru

## detectCheckerboardPoints

`[imagePoints, boardSize] = detectCheckerboardPoints(im)` - vráti pozície rohov šachovnice na obrázku a veľkosť šachovnice

## detectCheckerboardPoints

Ak sme mali posun začiatku obrazu, tak musíme posunúť aj nájdené body tj. `imagePoints = imagePoints + newOrigin`.

## generateCheckerboardPoints

`worldPoints = generateCheckerboardPoints(boardSize, squareSize)`  
- vráti rozloženie bodov na šachovnici, `squareSize` určuje šírku políčka v milimetroch.

# Vonkajšie parametre

## extrinsics

$[R, t] = \text{extrinsics}(\text{imagePoints}, \text{worldPoints}, \text{cameraParams})$  -  
vráti rotačnú maticu  $R$  a translačný vektor  $t$  pre dané body a  
parametre kamery.

## Poznámka

Ak sme vnútorne parametre kamery zisťovali aj z obrázku kde  
chceme vedieť vonkajšie parametre, tak nám stačí zobrať tieto  
údaje zo štruktúry `cameraParams`.

# Transformácia bodov

## pointsToWorld

`pointsToWorld(cameraParams, R, t, imagePoints)` - vráti súradnice bodov v rovine kalibračného vzoru (v milimetroch). Na vstupe sú parametre kamery, rotačná matica, translačný vektor a matica  $m \times 2$  kde v prvom stĺpci sú x-ové súradnice bodov a v druhom stĺpci sú y-ové súradnice bodov na obrázku.

## Pozor!

Ak máme body napr. z `ginput()`, tak je nutné k ním opäť prirábať `newOrigin`, ak nieje nulový.

## Úloha

Použite `ginput()` a odmerajte vzdialenosť, alebo rozmery nejakého objektu pomocou tohto postupu.

# Histogram

## imhist

imhist(I) - zobrazí histogram, v prípade že výstup zapíšeme do premennej, tak nič nevykreslí ale vráti nam vektor s histogramom.

## Úloha

Preveďte zatisie.jpg na šedotónový obrázok. Na jeho histograme sú tri peaky. Upravte obrázok, tak aby pixely približne patriace len jednému z peakov boli úplne biele, ostatné nechajte tak.



# Úprava jasu

## Gamma korekcia

Kontrast v obraze môžeme meniť pomocou gamma korekcie:

$i_{out} = A \cdot i^\gamma$ , kde  $i$  predstavuje jas jednotlivých pixelov obrázka. Pozor jas musí byť medzi 0 a 1!

## Lineárne roztiahnutie

Pre lineárne roztiahnutie môžeme použiť nasledujúcu úpravu:

$$i_{out} = \frac{i - \max(I)}{\max(I) - \min(I)},$$

kde  $i$  sú hodnoty jasu pre jednotlivé pixely a  $I$  predstavuje množinu jasov všetkých pixelov. Tiež chceme hodnoty jasu medzi 0 a 1.

# Ekvalizácia histogramu

## Ekvalizácia

Ekvalizácia histogramu je metóda, ktorá mení jas v obraze tak, aby výsledný histogram vyzeral čo najrovnomernejšie.

## histeq

`histeq(I)` - vráti obrázok po ekvalizácii histogramu.

## Úloha

Pre obrázok `krajinka.png` vyskúšajte rôzne metódy úpravy kontrastu. Po úpravách si zobrazte obrázky aj histogramy.

# Prahovanie

## imbinarize

imbinarize(l) - vráti binarizovaný obraz s prahom určeným Otsuho metódou.

## imbinarize

imbinarize(l, t) - vráti binarizovaný obraz s prahom t.

## Úloha

Vyskúšajte prahovanie na obrázkoch coins.png, qr.jpg a zatisie.jpg.

# Adaptívne prahovanie

## imbinarize

`imbinarize(I, 'adaptive')` - vráti binarizovaný obraz s použitím adaptívneho prahovania.

## Úloha

Vyskúšajte adaptívne prahovanie na obrázku `coins.png` a `qr.jpg`.