

OpenCV

Andrej Lúčný

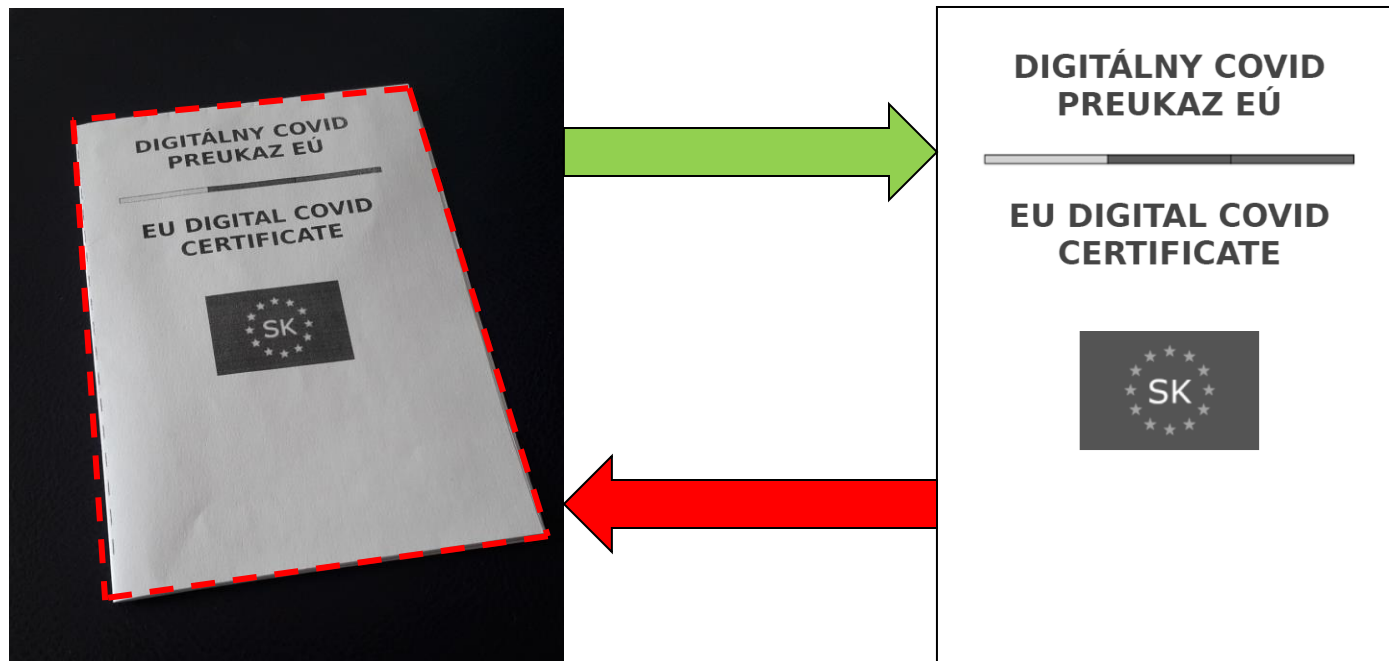
Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

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

www.agentspace.org/opencv

Image **alignment** / **registration**



Alignment je skoro to isté ako registrácia, akurát výstup alignmentu je obrázok objektu v štandardnom tvare a registrácia je deformácia štandardného tvaru do zodpovedajúceho objektu na obrázku

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scaling matrix

$$\begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation matrix

$$\begin{bmatrix} s_x & s_{xy} \\ s_{yx} & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Linear transform

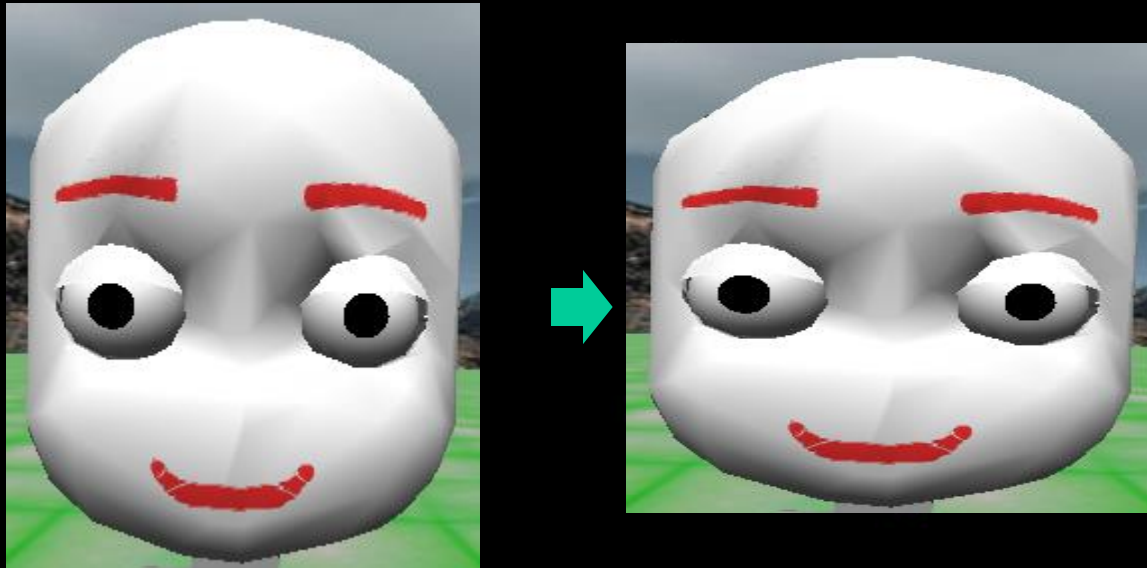
$$\begin{bmatrix} s_x & s_{xy} & t_x \\ s_{yx} & s_y & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affine transform

$$\begin{bmatrix} s_x & s_{xy} & t_x \\ s_{yx} & s_y & t_y \\ t_{yx} & t_{yx} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography

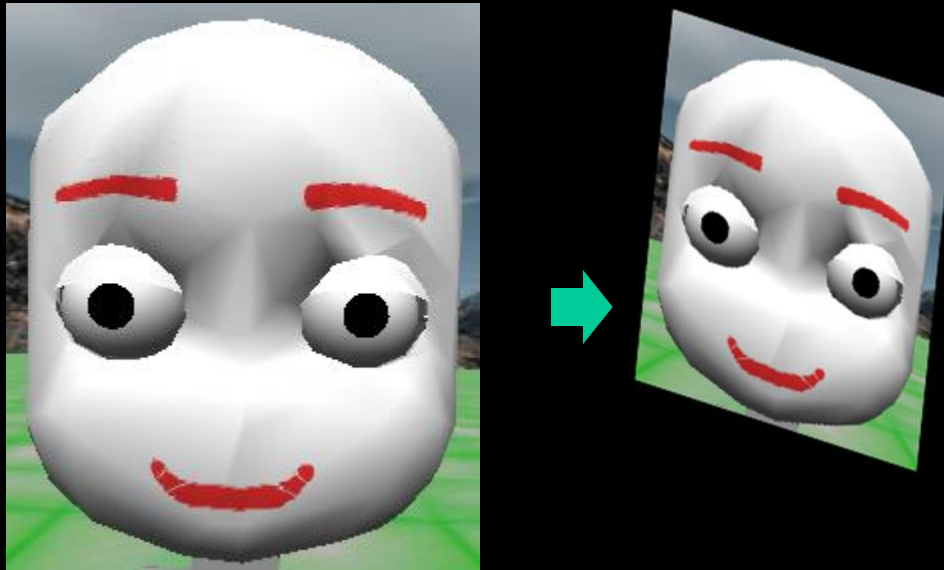
Scaling



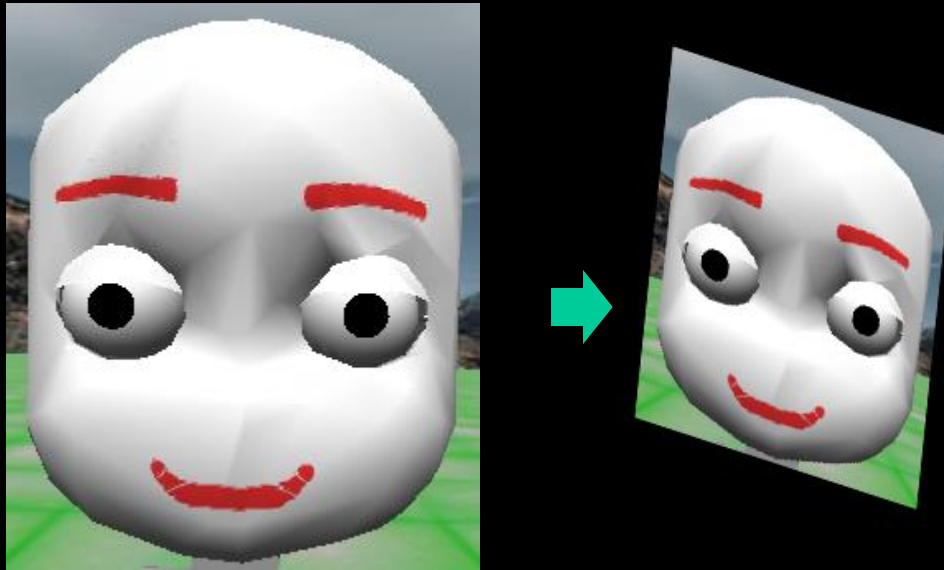
Rotation



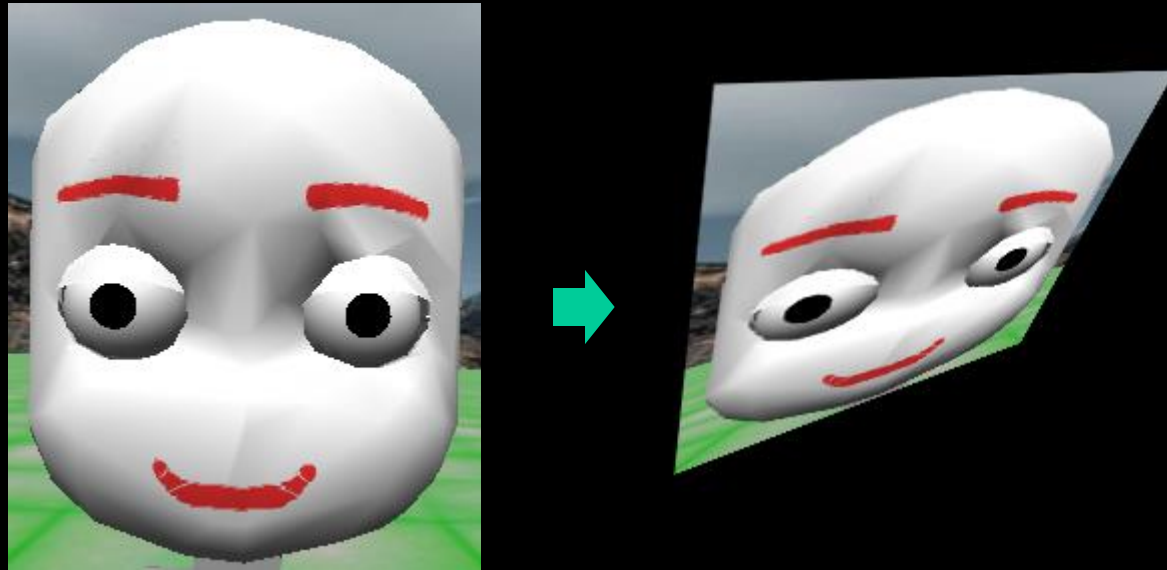
Linear transform



Affine transform



Homography



Identita

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

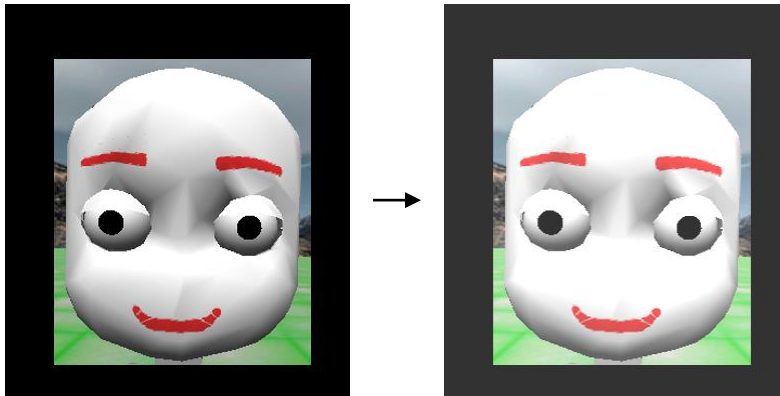
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
h = np.array([[1.3,0.1,1],[0.2,0.9,2],[0.002,0.0015,0.5]])  
img2 = cv2.warpPerspective(img,h,(img.shape[1],img.shape[0]))
```

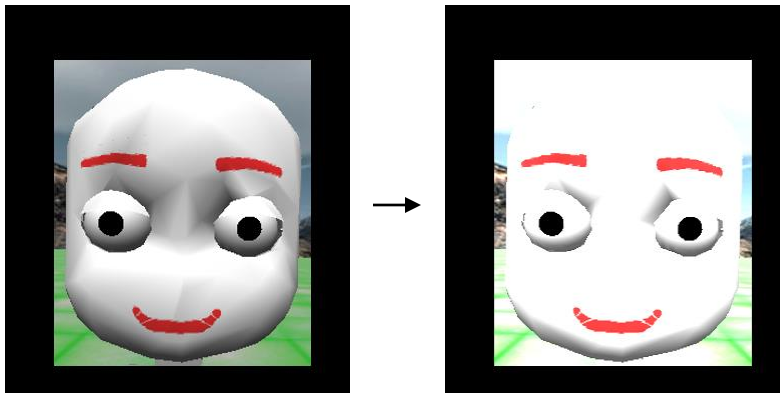
Aritmetické operácie s obrazom

- Jas (brightness) = pripočítanie konštanty



```
cv2.add(  
    img,  
    50*np.ones_like(img)  
)
```

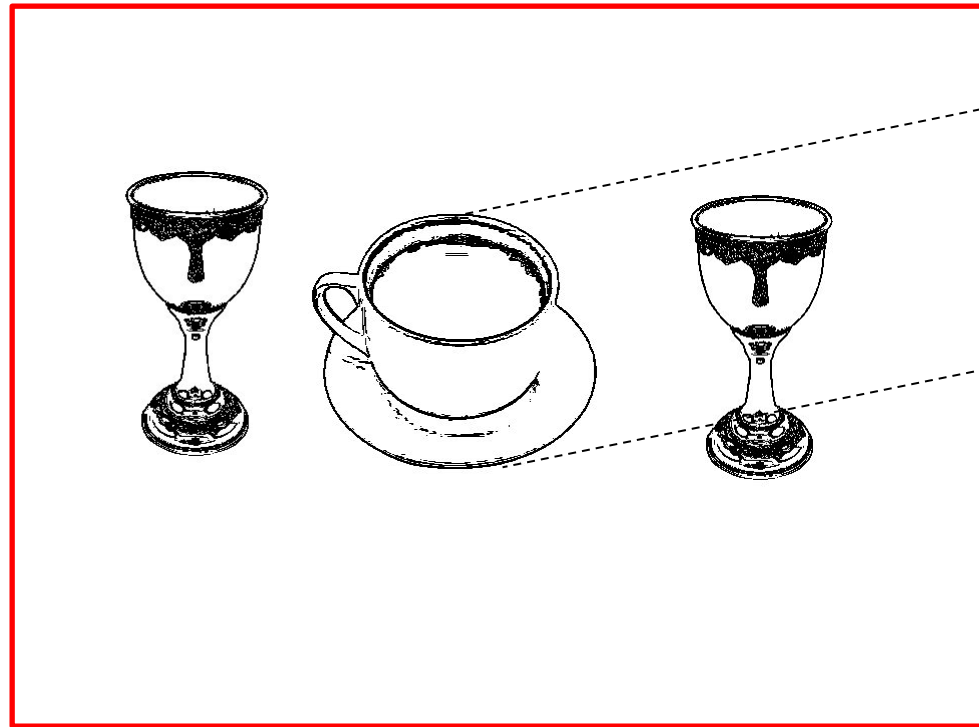
- Kontrast (contrast) = vynásobenie konštantou



```
cv2.multiply(  
    img,  
    2*np.ones_like(img)  
)
```

fyzikálne zodpovedá zmene osvetlenia

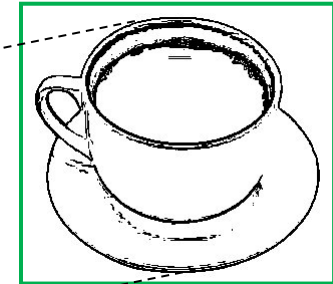
Registrácia: Brute force



P pixelov

frame

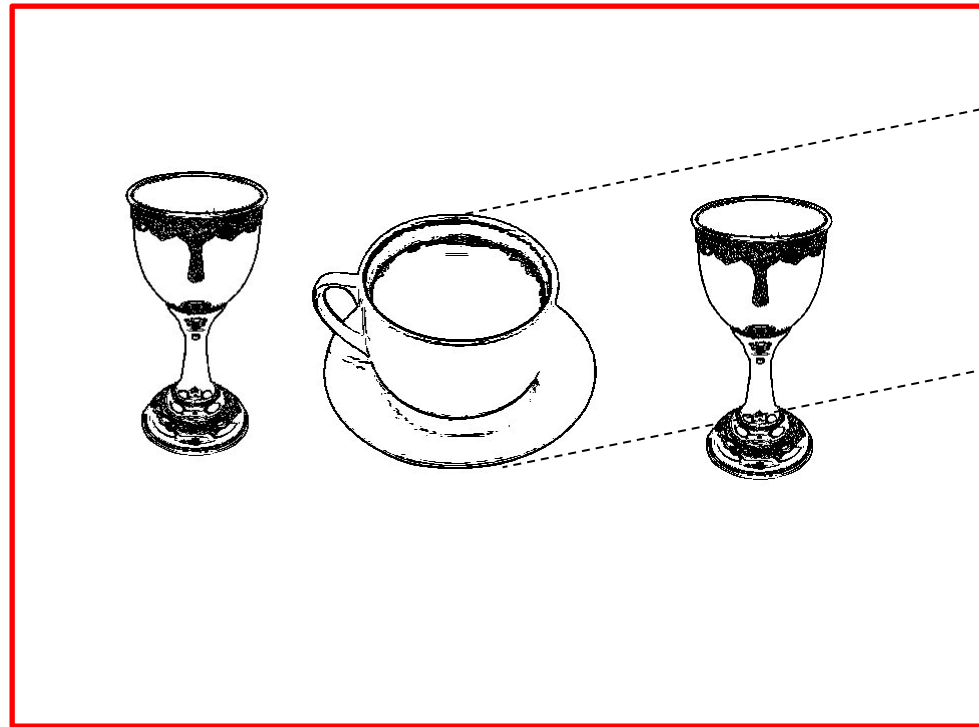
reference



Q pixelov

Polož vzor na
každé miesto a
hľadaj
minimálny
rozdiel $O(PQ)$

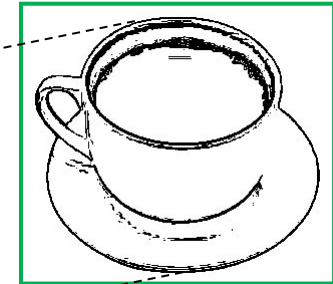
Registrácia: Brute force



P pixelov

frame

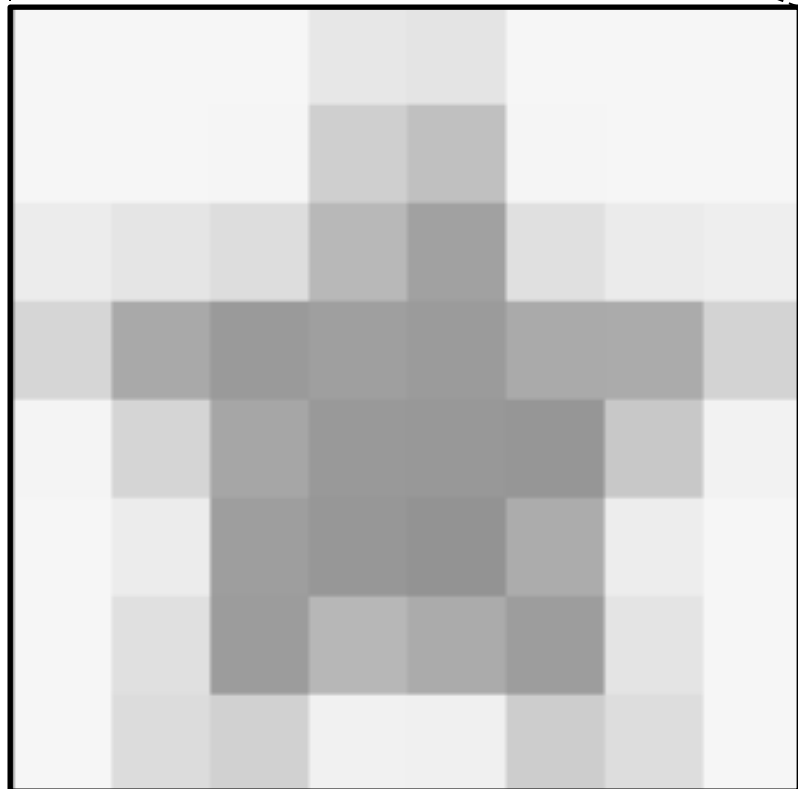
reference



Q pixelov

Pomocou
Fourierovej
transformácie sa to
dá urobiť efektívne
 $O(P \log P)$

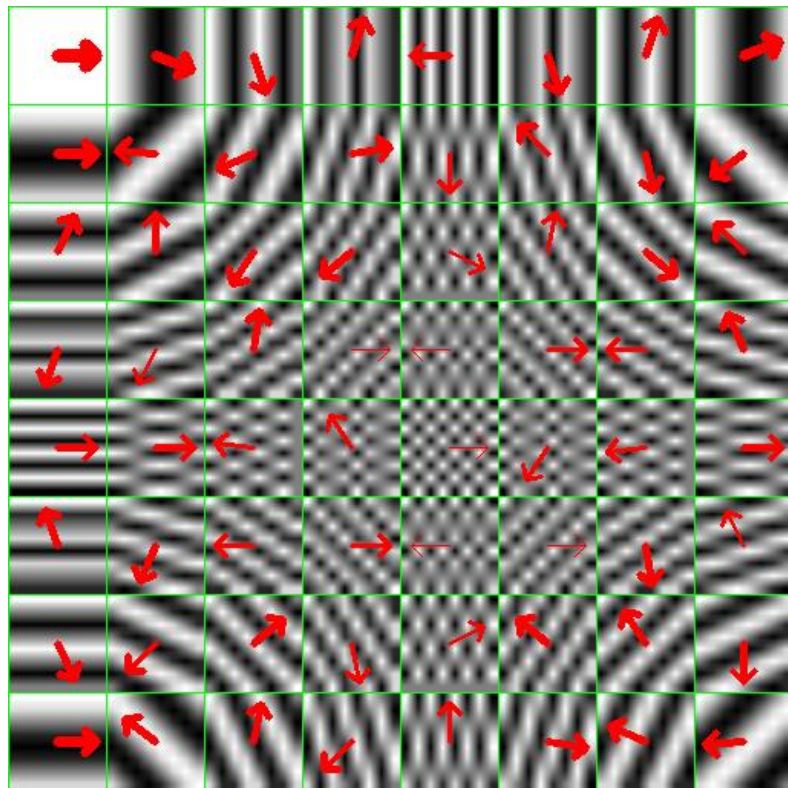
(Fourier Transform)



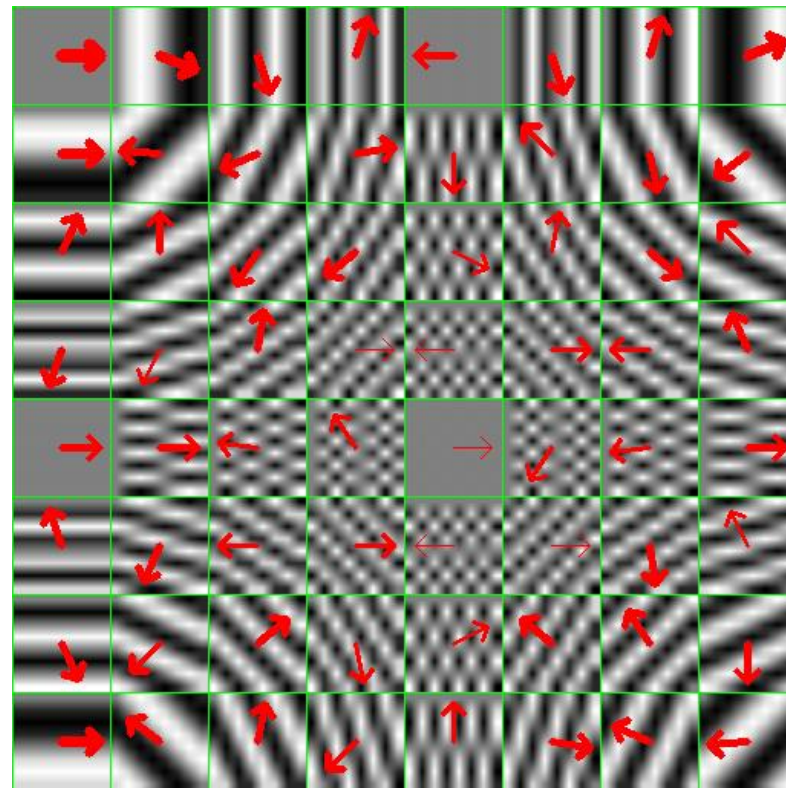
Každý obraz možno vyjadriť ako lineárnu kombináciu (s komplexnými koeficientami) „vlnových“ obrazov (s komplexnými pixelmi)

(Fourier Transform)

Re



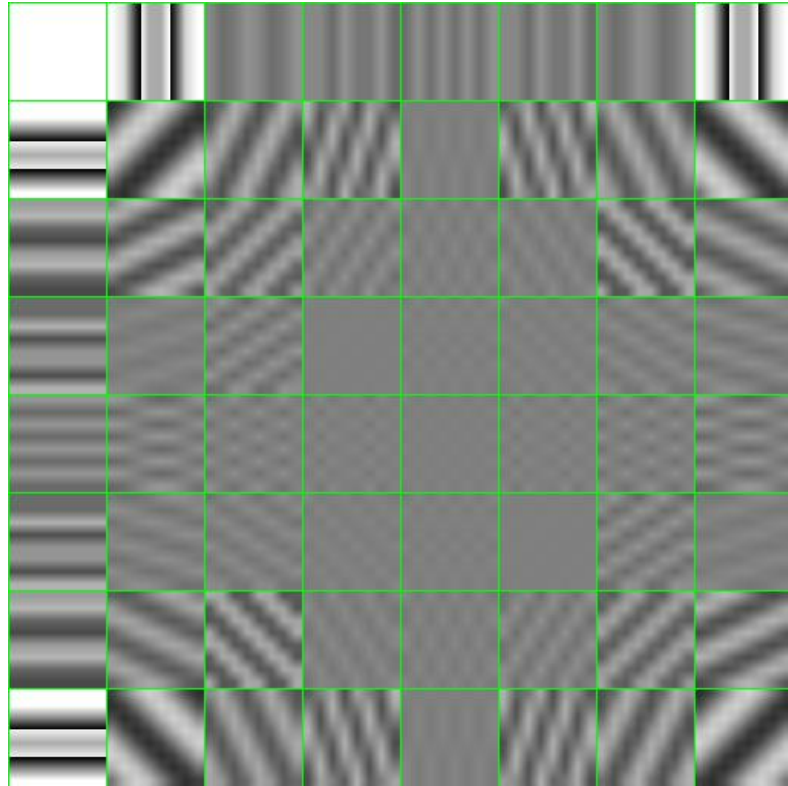
Im



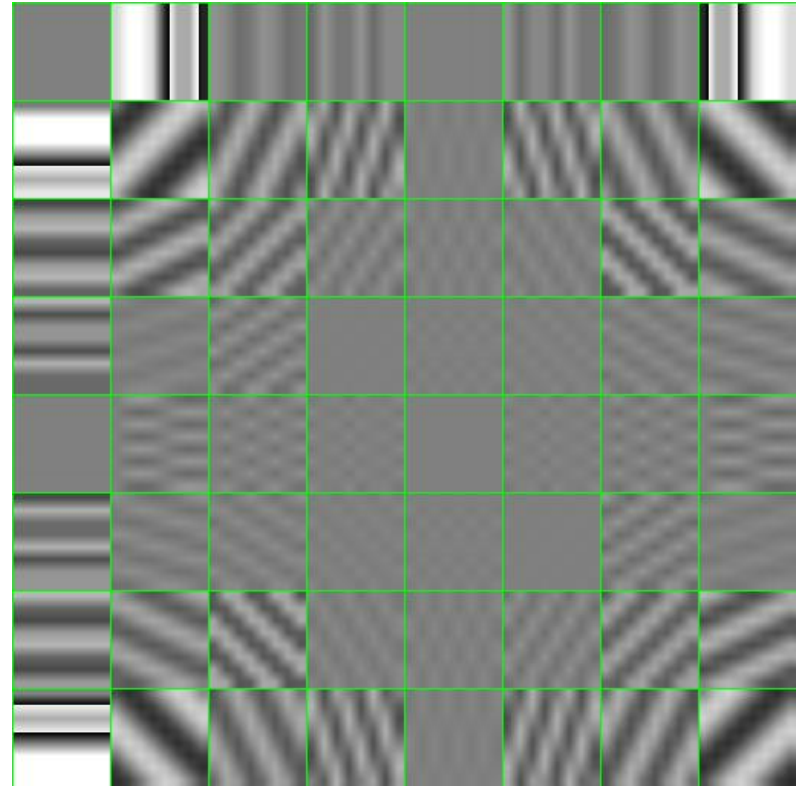
Fourierove koeficienty sú komplexné čísla, majú svoju amplitúdu (hrúbka šípky) a fázu (orietácia). Fáza ostáva rovnaká pokiaľ sa mení len kontrast obrázku (je invariantná na násobenie)

(Fourier Transform)

multiplied: Re



Im

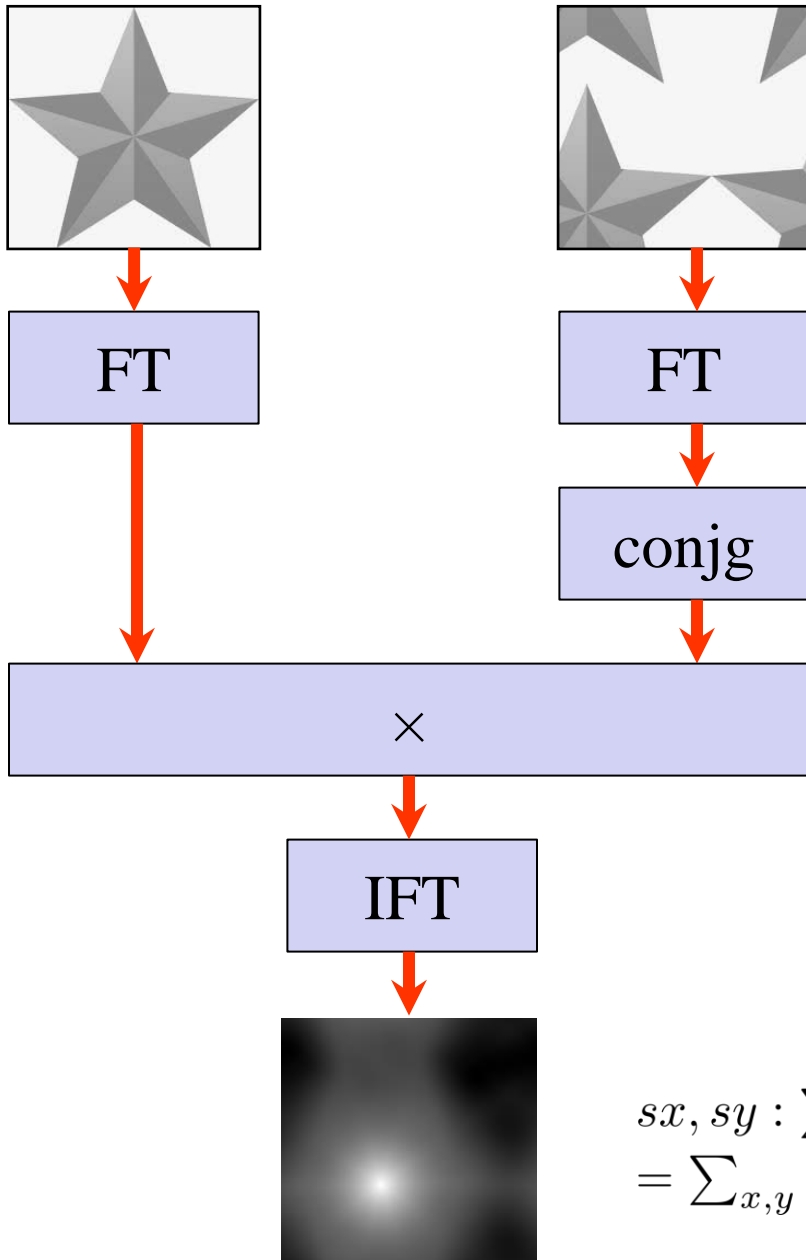


total:



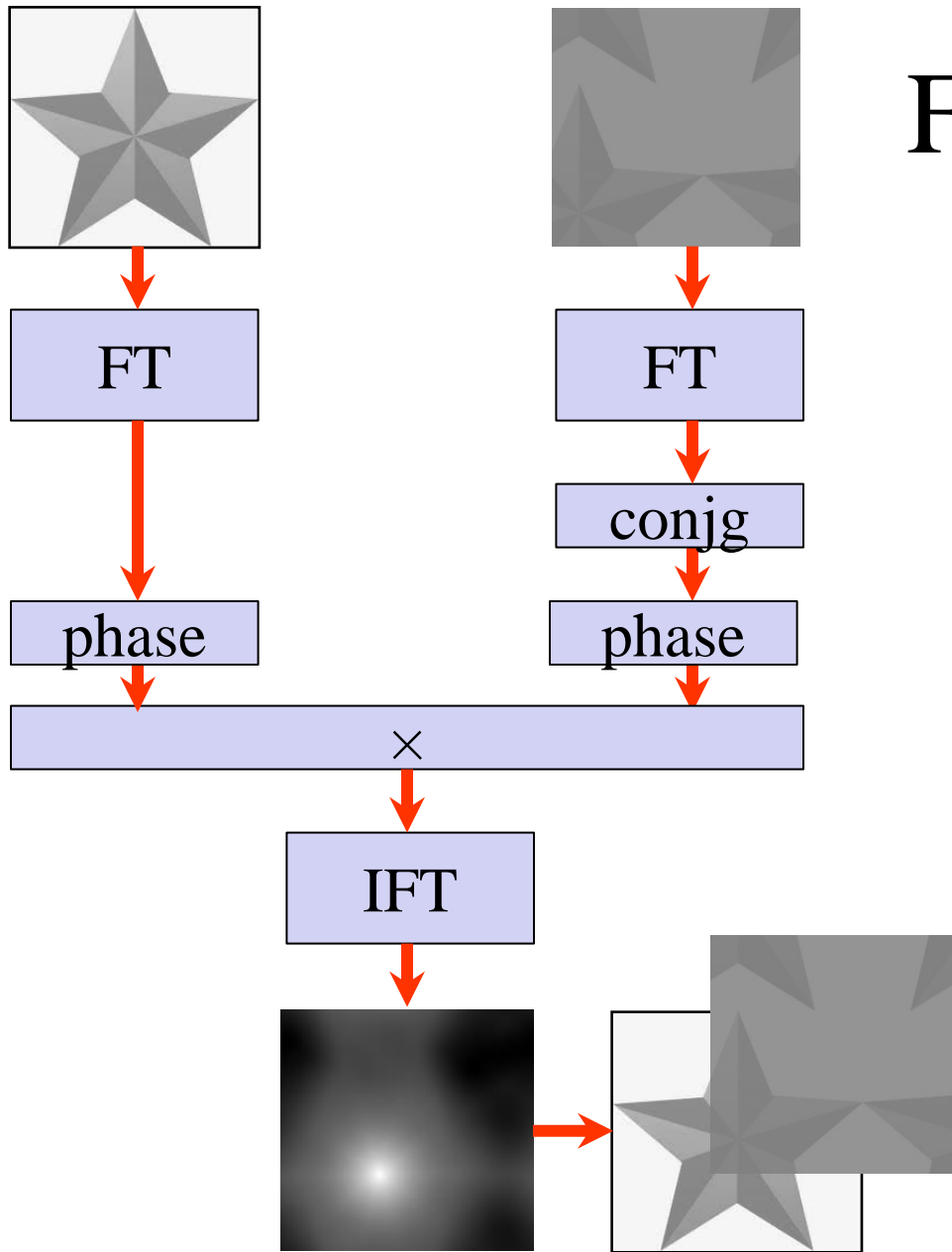
total:





Pomocou Fourierovej transformácie vieme efektívne spočítať sumy súčinov dvoch obrázkov pre každé možné posunutie. Pre to posunutie, kde je suma súčinov najväčšia, je “rozdiel” (MSE) obrázkov najmenší

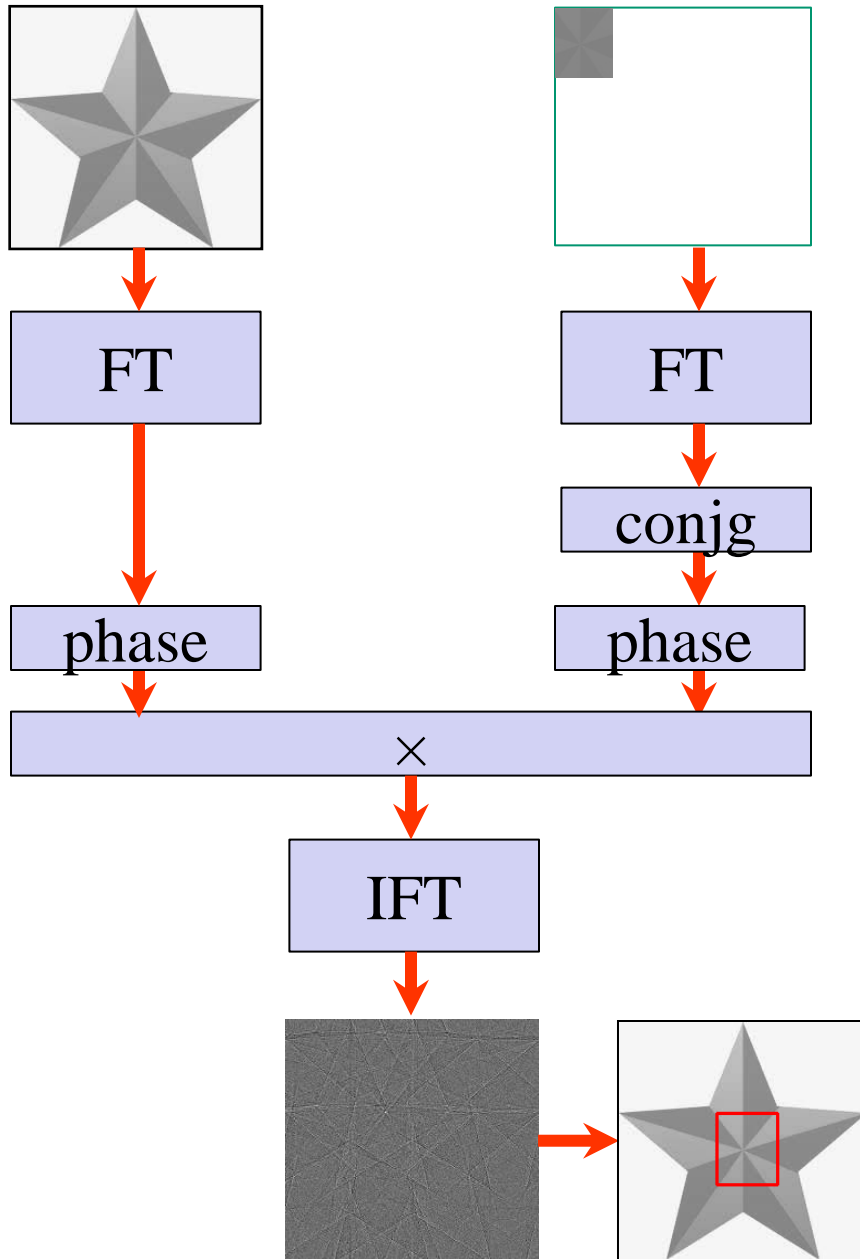
$$\begin{aligned}
 sx, sy : \sum_{x,y} (a_{x-sx,y-sy} - b_{x,y})^2 &= \\
 &= \sum_{x,y} a_{x,y}^2 - 2 \sum_{x,y} a_{x-sx,y-sy} b_{x,y} + \sum_{x,y} b_{x,y}^2
 \end{aligned}$$



Fázová korelácia

Pokiaľ všetky amplitúdy nastavíme na 1, beriem v úvahu len fázu a tá je nezávislá na kontraste, takže nájdeme aj taký výskyt, ktorý ma iný kontrast než vzor

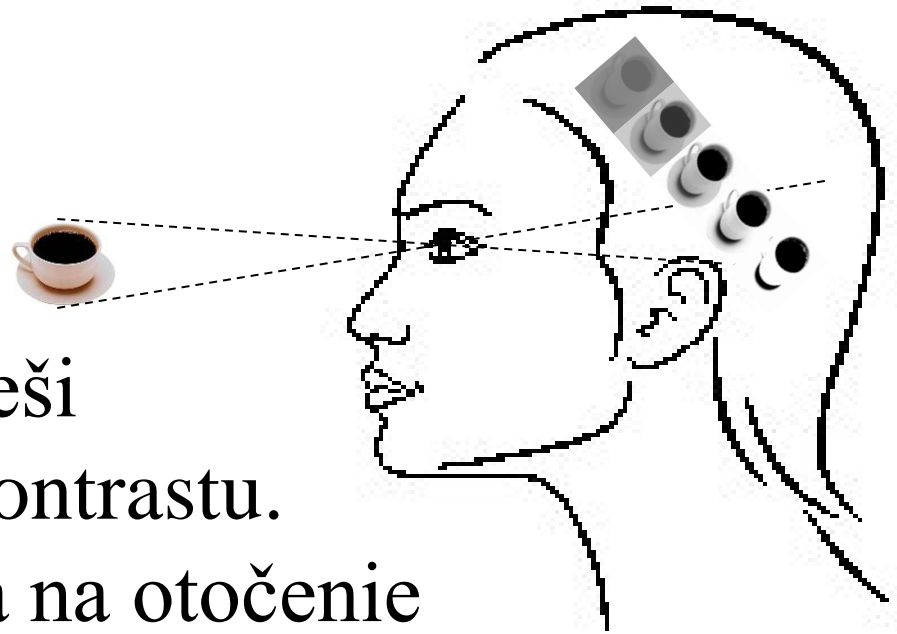
Fázová korelácia



Spravidla máme ako vzor len časť obrázka, tu vieme s istou mierou rizika vhodne doplniť, aby to stále fungovalo (bezpečné pre binárne obrázky, spoľahlivé pre centrovane obrázky)

Fázová korelácia

- Jej klasická verzia rieši posunutie a zmenu kontrastu. Existujú aj rozšírenia na otočenie a zmenu škály (rovnako vo všetkých dimenziách)



Enhanced Correlation Coefficient

- Gradient descent s obrázkami
- Definuje chybovú funkciu zodpovedajúcu podobnosti dvoch na seba položených obrázkov, jeden je pôvodný a druhý podrobený afinnej transformácii
- Chybovú funkciu symbolicky derivuje a počíta gradient – zmenu parametrov affínnej transformácie
- Iteratívne nájde afínnu transformáciu

Enhanced Correlation Coefficient

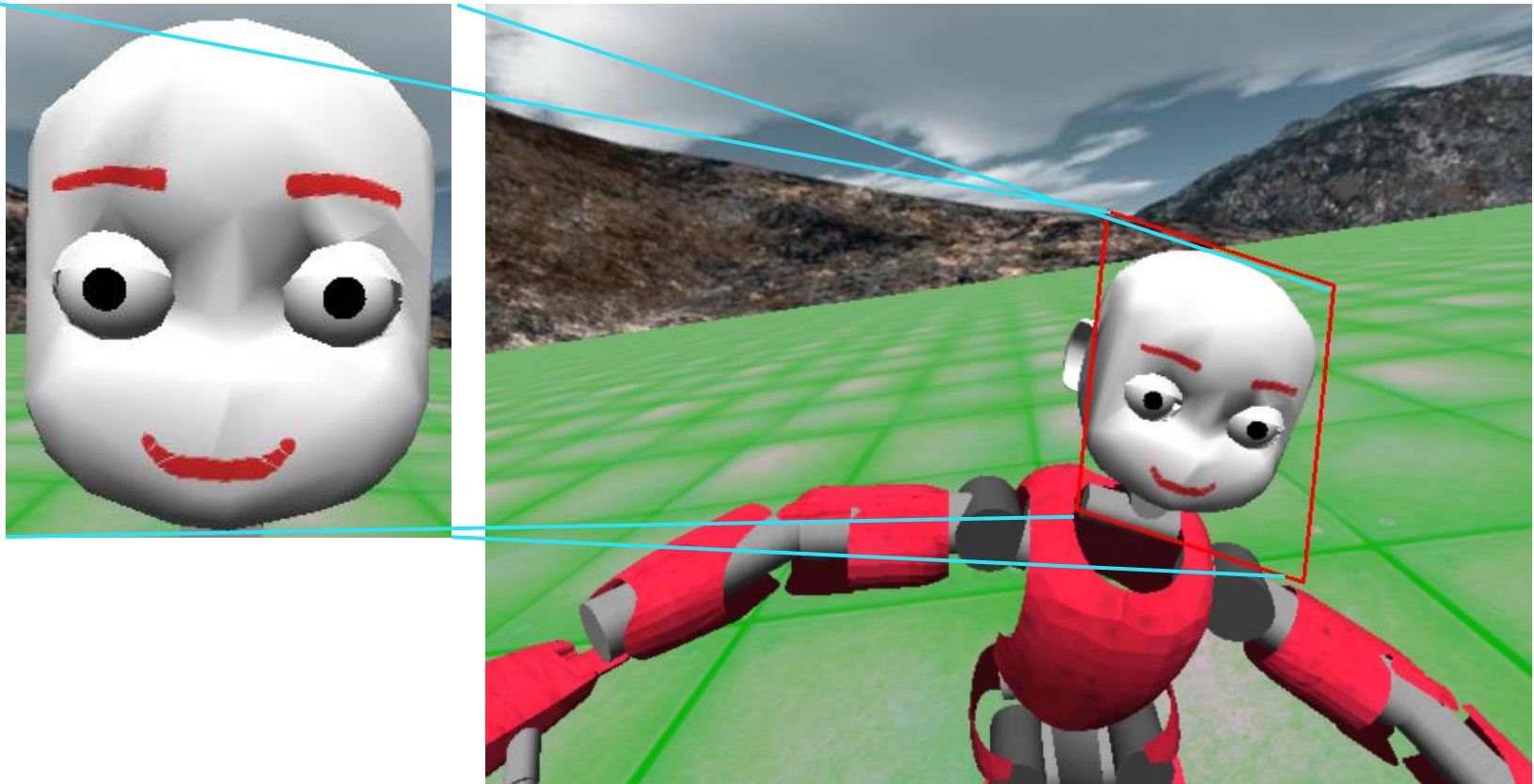
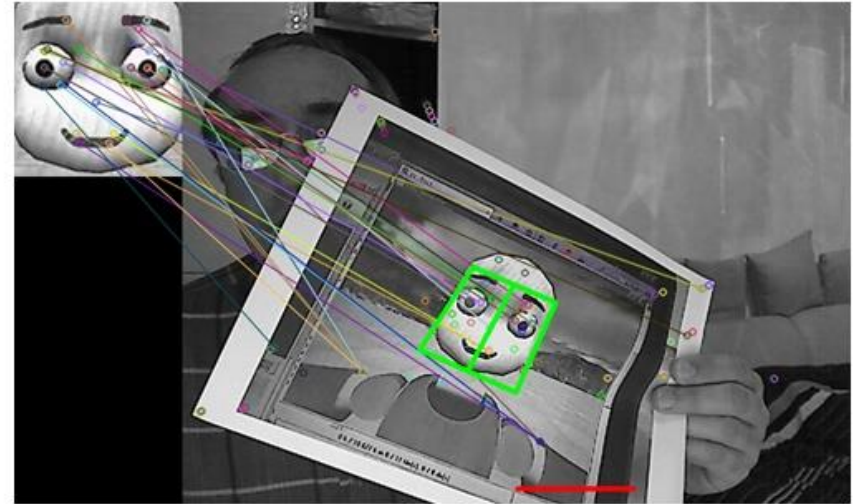


Image features

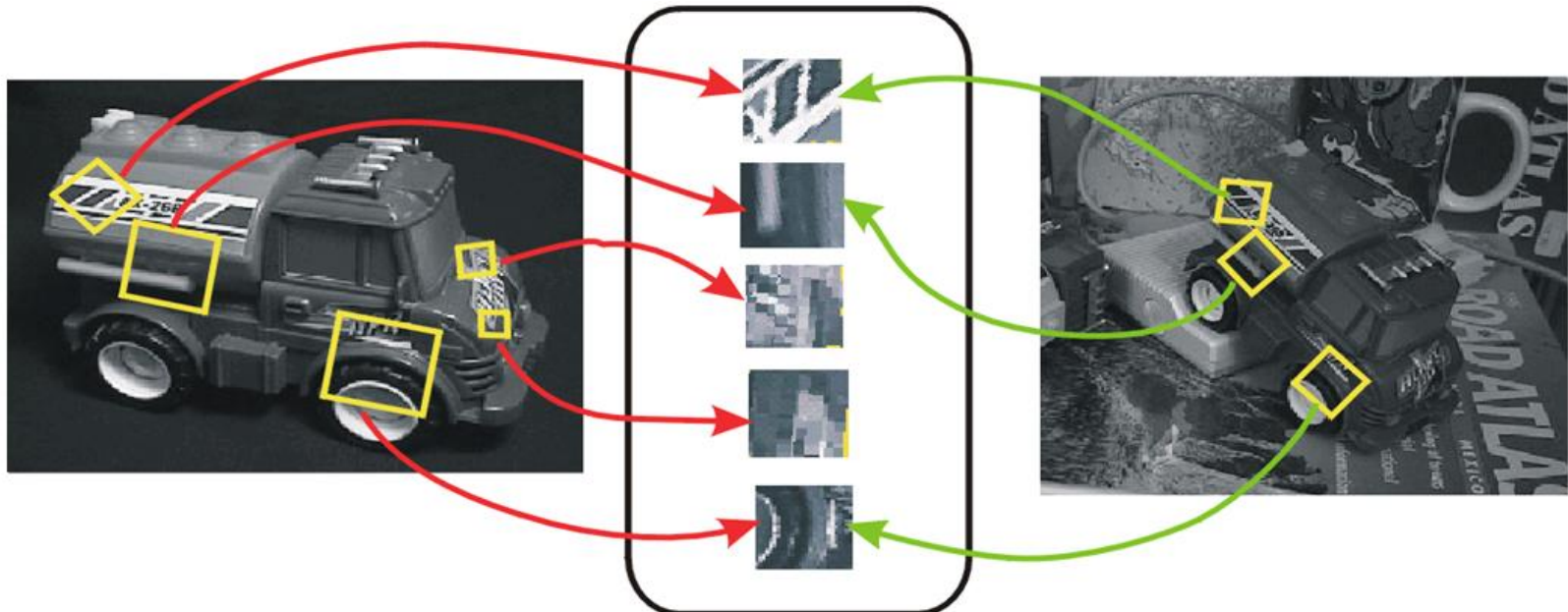


Obrazu priradujú zoznam zaujímavých bodov a ich deskriptorov, ktoré sú určitým spôsobom invariantné k posunutiu, otočeniu, zväčšeniu, zmenšeniu a pod.

- SIFT
- SURF
- FAST, BRIEF, ORB

SIFT (Scale Invariant Feature Transform)

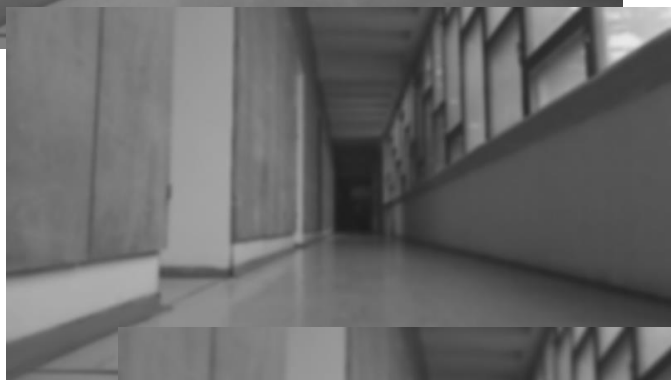
Vzor predstavujú význačné body, ktorých okolie nachádzame v inej mierke a otočení na inom obraze



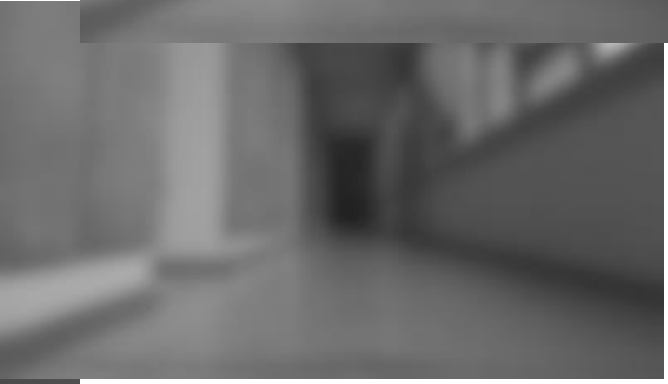
SIFT (Scale Invariant Feature Transform)



Význačné
body
spočítame ich
ako dôležité
detaily, ktoré
sa stratia pri
podvozokovaní
obrazu



1. oktáva



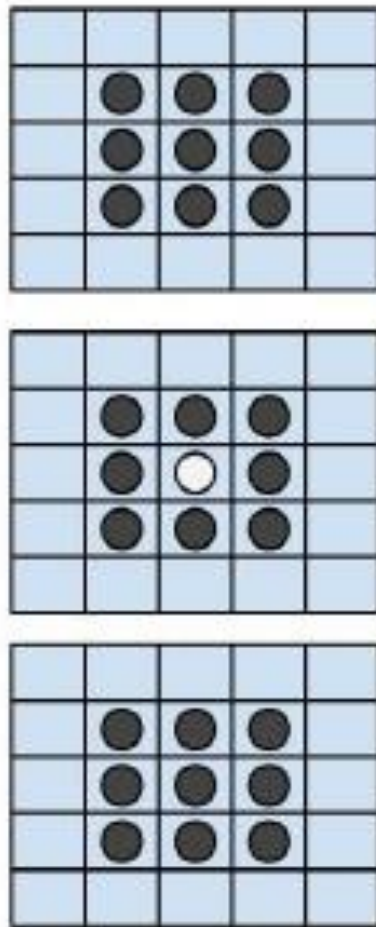
2. oktáva



3. oktáva

4. oktáva

SIFT (Scale Invariant Feature Transform)

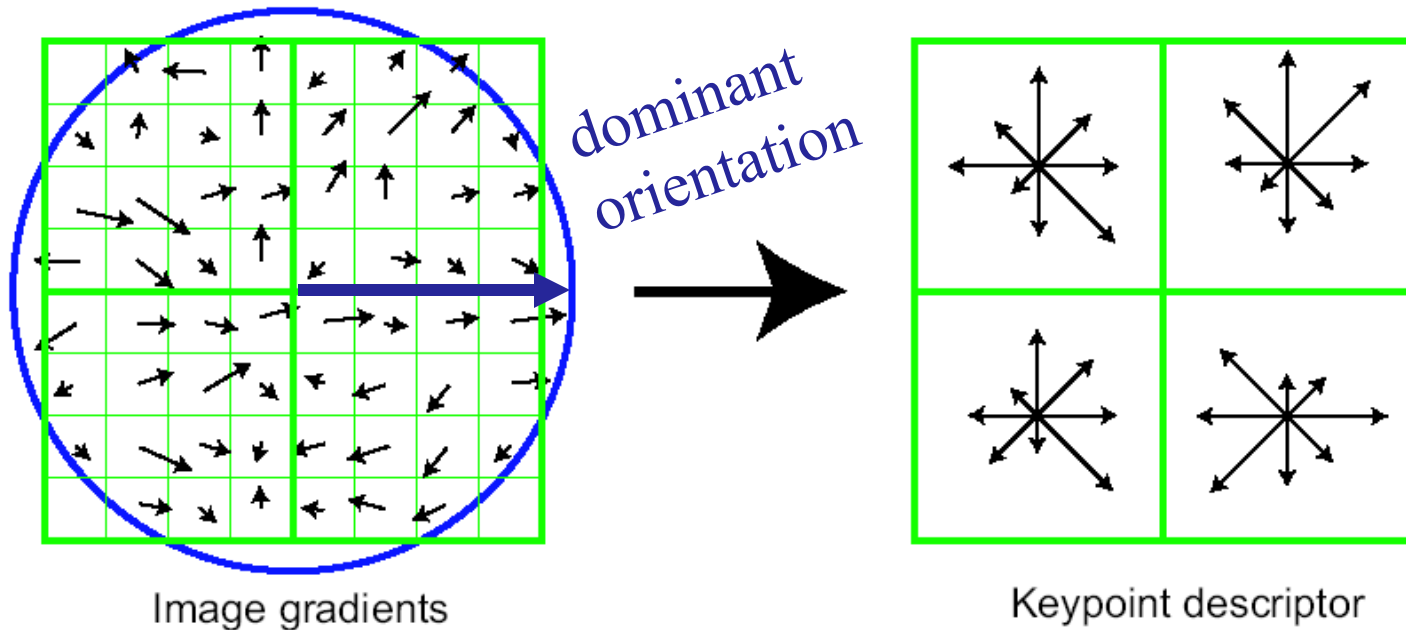


Oktáva

Význačný bod
hľadáme ako
extrém v 3D

(Difference of
Gaussians)

SIFT (Scale Invariant Feature Transform)

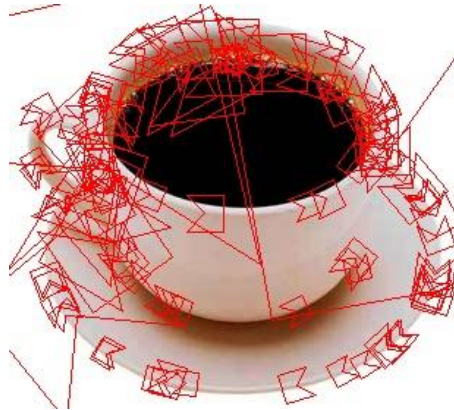


Pre každý bod urobíme popis jeho okolia –
deskriptor (podobný šablóne)

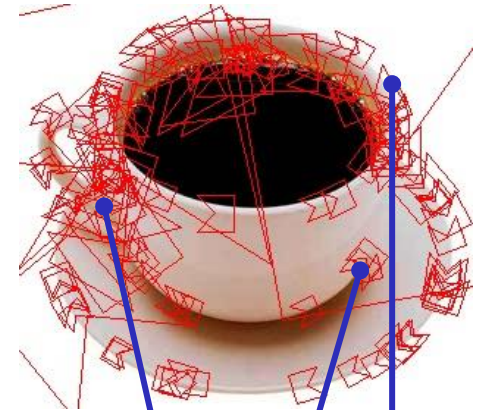
Párovanie je založené na porovnávaní deskriptorov
Algoritmus RANSAC (RANdom SAmple Consensus)



obraz



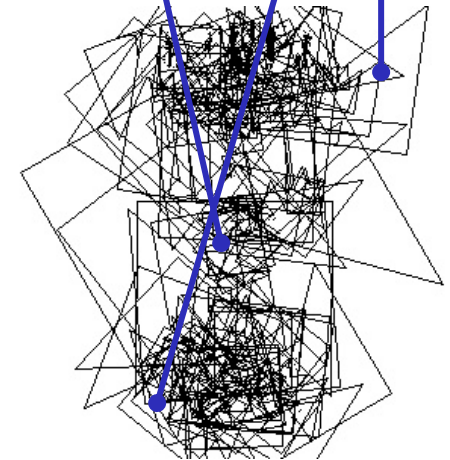
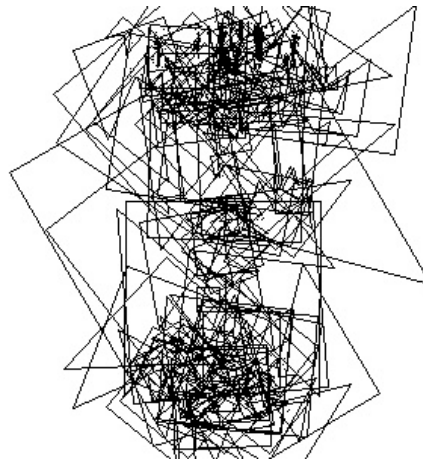
SIFT



párovanie bodov



VZOR

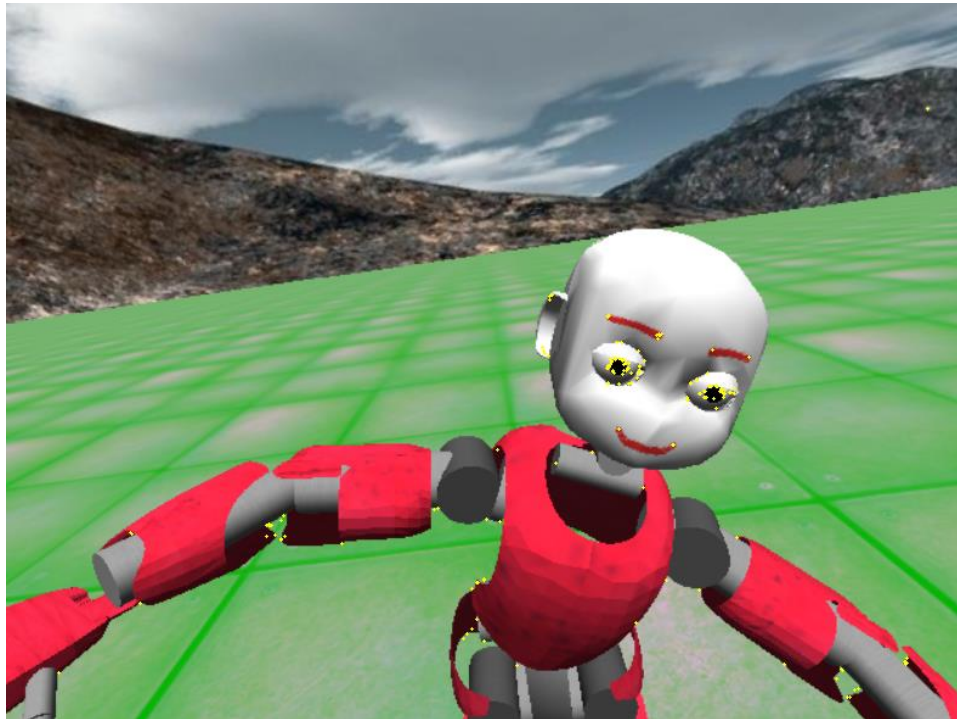


Non-free

- SIFT a SURF boli patentované a OpenCV ich malo len v non-free časti (patent vypršal 2020/03)
- OpenCV podporovalo preto hlavne podobný, o niečo menej výkonný, ale dostupný ORB

FAST

- keypoint detector
- Zaujímavé sú body v ktorých okolí je dostatočné množstvo tmavších i svetlejších



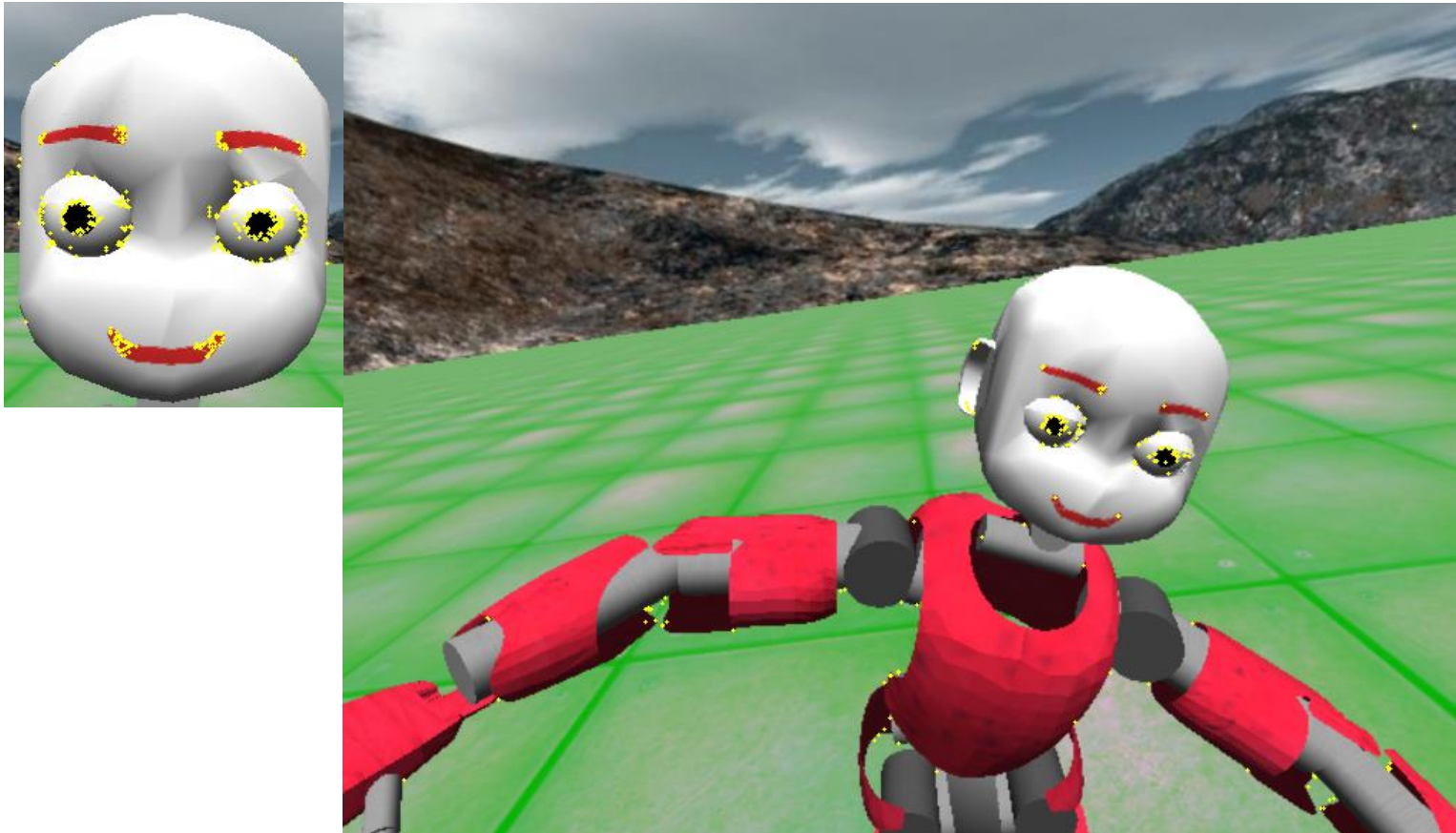
- Tie pretriedime cez Non-Maximum suppression

BRIEF

- Descriptor
- Binary Robust Independent Elementary Features
- Na rozdiel of SIFT a SURF, kde je descriptor vektor floatov, tento deskriptor je vektor bitov
- Rýchlejšie sa porovnáva, zaberá menej pamäte
- Nie je však odolný voči rotácii
- Bity sú výsledky porovnaní intenzít reprezentatívnych bodov v okolí keypointu

- Oriented FAST and Rotated BRIEF
- FAST keypointy sa pretriedia podľa Harrisovej mierky a zrátajú sa BRIEF pre každých 15 stupňov.

ORB



- Matcher nájde z dvojíc podobných keypointov homografiu

ORB

