

# OpenCV

*Andrej Lúčny*

*Katedra aplikovanej informatiky FMFI UK*

*lucny@fmph.uniba.sk*

*[http://dai.fmph.uniba.sk/w/Andrej\\_Lucny](http://dai.fmph.uniba.sk/w/Andrej_Lucny)*

*[www.agentspace.org/opencv](http://www.agentspace.org/opencv)*

# Práca s kamerou a videom

```
camera = cv2.VideoCapture(0) #1 #"video.avi"  
#camera=cv2.VideoCapture("http://208.72.70.171/mjpg/video.mjpg")  
  
while True:  
    hasFrame, frame = camera.read()  
    if not hasFrame:  
        break  
    cv2.imshow('camera', frame)  
    key = cv2.waitKey(10)  
    if key == 27:  
        break
```

OpenCV podporuje pripojenie rôznych kamier usb i sieťových

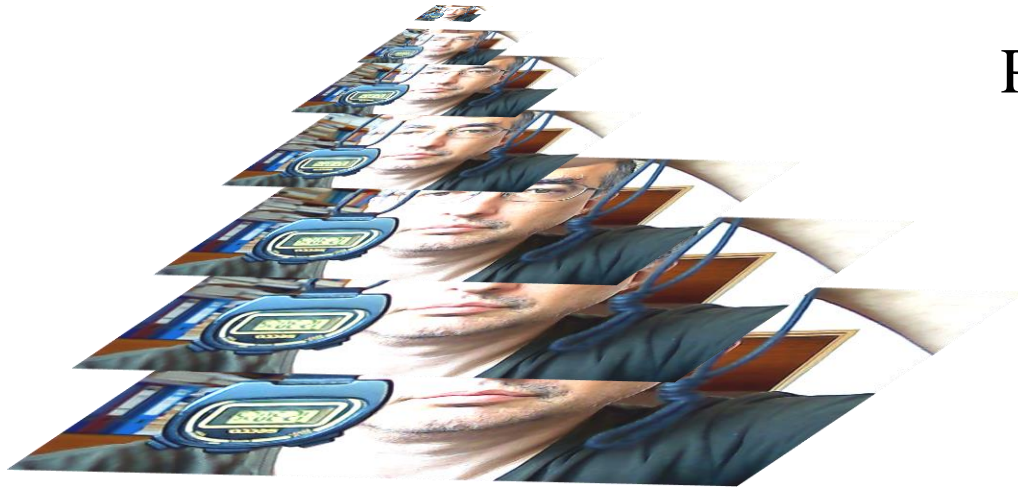
# Optický tok



$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

$(\Delta x, \Delta y)$  ... flow

# Optický tok

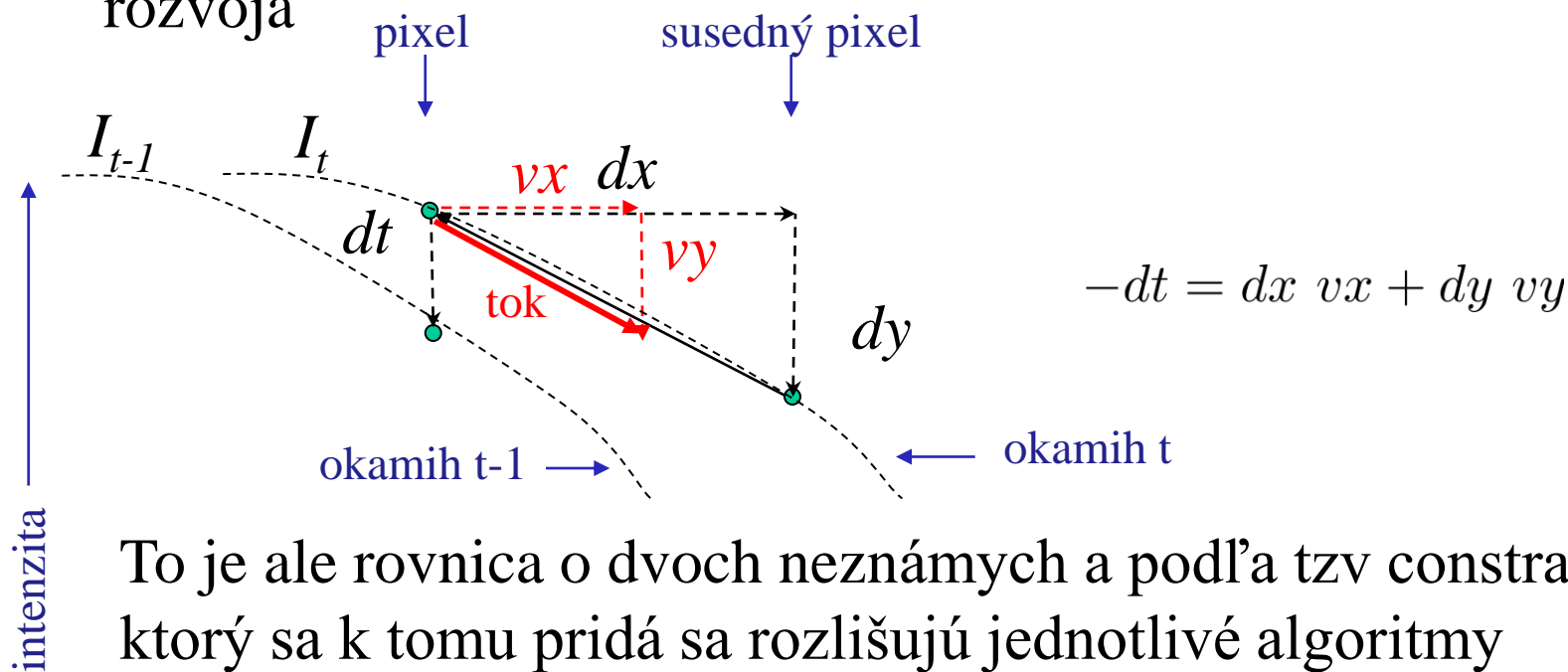


Pyramidal image

Optický tok vieme dobre spočítať pokiaľ je menší ako 1 pixel  
Väčšie toky počítame preto na zmenšenom obraze  
Začneme na vrchole pyramídy a obraz remapujeme podľa  
toku a zväčšíme, upravíme podľa ďalších zistených tokov  
menších ako 1 pixel, atď a nakoniec toky na jednotlivých  
úrovniach pyramídy zintegrujeme do výsledku.

# Optický tok

Optický tok menší ako 1 pixel počítame pomocou Taylorovho rozvoja



To je ale rovnica o dvoch neznámých a podľa tzv constraint-u ktorý sa k tomu pridá sa rozlišujú jednotlivé algoritmy

Lucas – Kanade alg. predpokladá, že tok podlieha len affinej transformácii.

# Good Features To Follow

Optický tok sa počíta pre jednotlivé body – ako ich vybrať?

Už vieme, že možno z obrazu vybrať „zaujímavé body“, tu ale nepotrebujeme, aby ich deskriptory boli invariantné. Ideálne sú rohy a hrany.

Obe možno detektovať napríklad Harrisovým detektorom



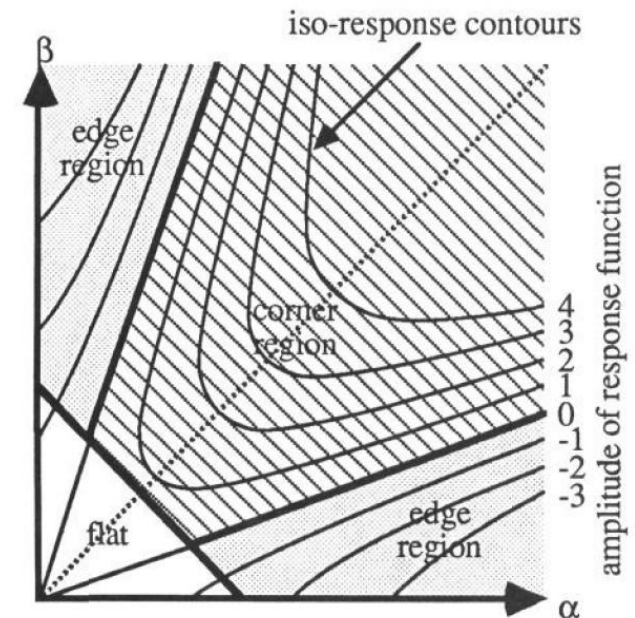
# Harrisov detektor



Druhú deriváciu intenzity obrazu určuje Hessian matrix

$$\begin{bmatrix} d_x^2 * G & d_x d_y * G \\ d_x d_y * G & d_y^2 * G \end{bmatrix}$$

Jej vlastné hodnoty  $\alpha$ ,  $\beta$  udávajú ako sa obraz mení v dvoch nezávislých na seba kolmých smeroch



Podľa veľkosti  $\alpha$ ,  $\beta$  vieme, či ide o plochu, hranu alebo roh



# Stereovision

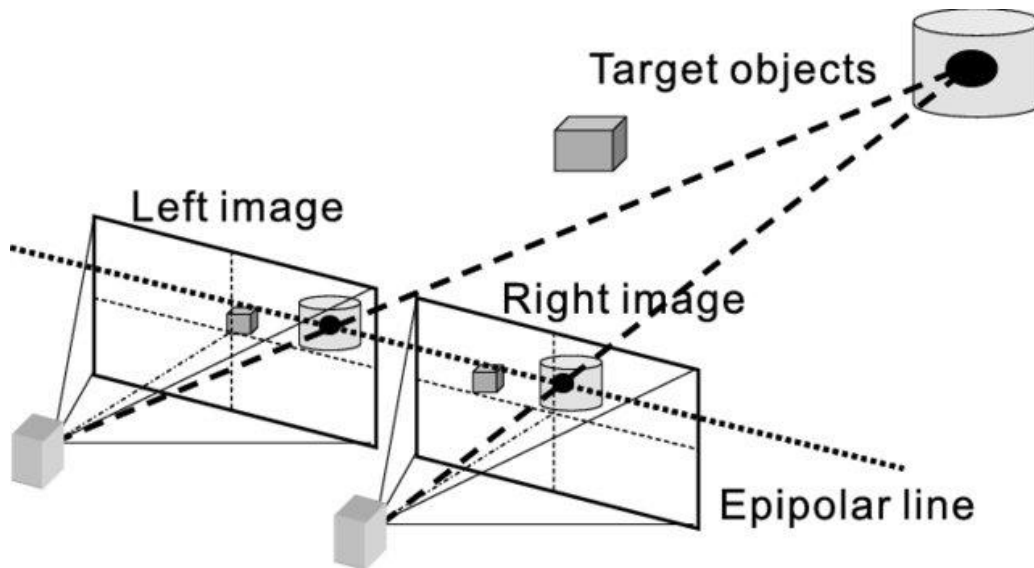


Hĺbka obrazu sa dá vypočítať trianguláciou, pokiaľ sa nám podarí určiť ktoré body na ľavom a pravom obraze si zodpovedajú. To vieme s dostatočnou presnosťou pomocou detektorov črt (feature detector) ako sú SIFT, SURF, ORB...



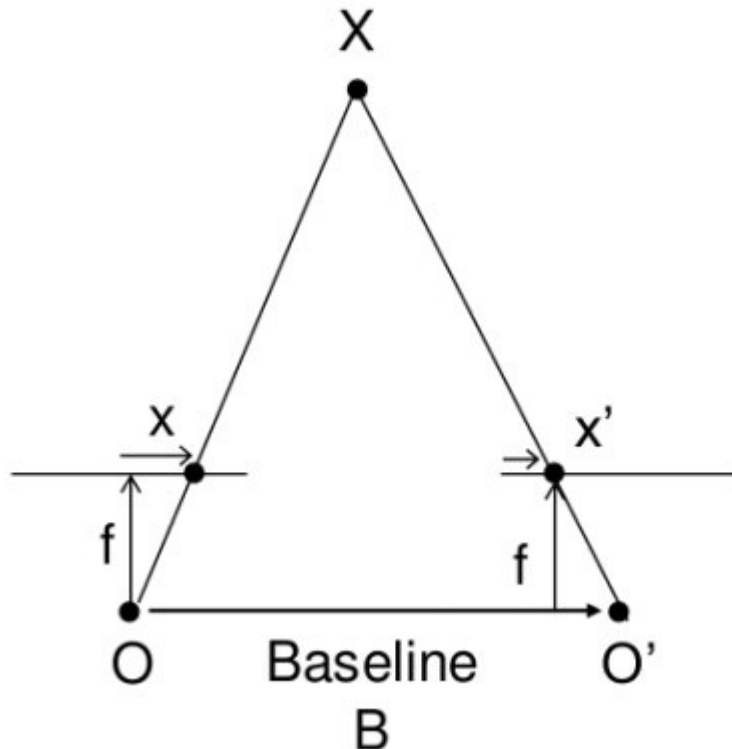
# Stereovision

Párovaniu význačných bodov výrazne pomáha, že nemôžu byť kdekol'vek. Pre každý bod sa dá určiť tzv. epipolárna priamka, na ktorej sa musí zodpovedajúci bod nachádzať.



# Stereovision

Hĺbka je potom nepriamo úmerná disparite, t.j. čím sú body od seba ďalej na epipolárnej priamke, tým sú bližšie

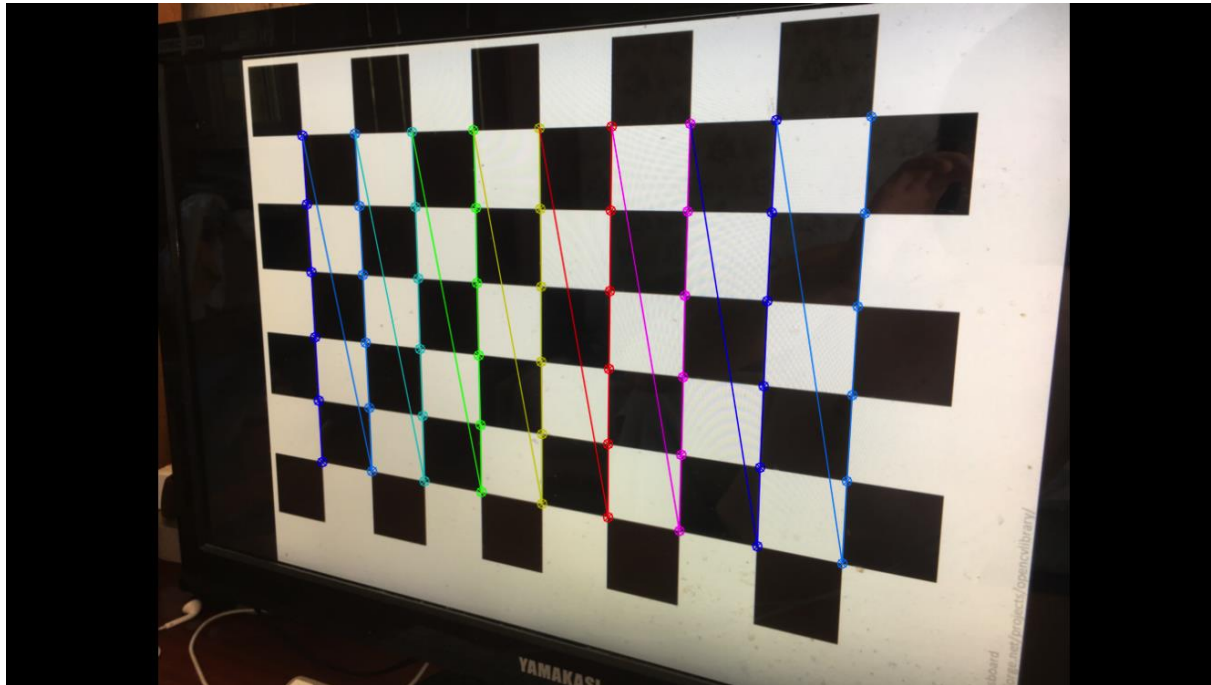


$$disparity = x - x' = \frac{Bf}{Z}$$

$z$

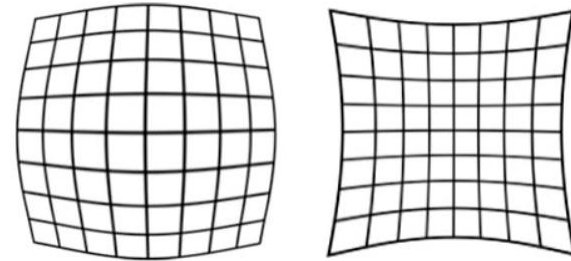
$x$   $x'$  sú pozície bodu na obraze  
 $B$  vzdialenosť kamier  
 $F$  ohnisková vzdialenosť kamier  
 $Z$  hĺbka

# Kalibrácia



Optika kamery na okrajoch obrazu skresľuje. Skreslenie sa dá zmierniť kalibráciou kamery, ktorá zaznamená parametre skreslenia pri predložení známeho objektu

# Kalibrácia



Radiálne  
skreslenie

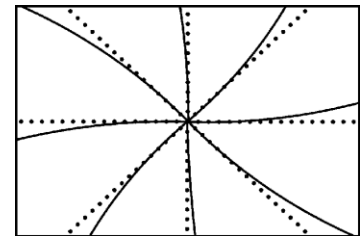
$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Tangenciálne  
skreslenie

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$

$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$



Parametre kalibrácie

$$Distortion\ coefficients = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$