# Image derivatives, edges, corners, . . .

**Tomáš Svoboda**, svoboda@cmp.felk.cvut.cz
Czech Technical University in Prague, Center for Machine Perception
http://cmp.felk.cvut.cz

**Last update:** October 26, 2009

## Talk Outline

- why corners?
- combined edge and corner detector
- Demo: step by step in Matlab

## Dense tracking



**Video: Dense tracking. Note that the coverage is not complete**

## What are good features (windows) to track?



How to select good templates $T(\mathbf{x})$ for image registration, object tracking.

$$\Delta\mathbf{p} = \mathtt{H}^{-1} \sum_{\mathbf{x}} \left[ \nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\top} \left[ T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]$$

where $\mathtt{H}$ is the Hessian matrix

$$\mathtt{H} = \sum_{\mathbf{x}} \left[ \nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\top} \left[ \nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]$$

The stability of the iteration is mainly influenced by the inverse of Hessian. We can study its eigenvalues. Consequently, the criterion of a good feature window is $\min(\lambda_1, \lambda_2) > \lambda_{min}$ (texturedness).

## What are good features (windows) to track?



Consider translation $\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$. The Jacobian is then $\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$
\begin{aligned}
\mathtt{H} &= \sum_{\mathbf{x}} \left[\nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^\top \left[\nabla I \, \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right] \\
&= \sum_{\mathbf{x}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial x}\right] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \sum_{\mathbf{x}} \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}
\end{aligned}
$$

The image windows with varying derivatives in both directions. Homeogeneous areas are clearly not suitable. Texture oriented mostly in one direction only would cause instability for this translation.

## What are the good points for translations?



The Hessian matrix

$$
\mathtt{H} = \sum_{\mathbf{x}} \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}
$$

Should have large eigenvalues. Harris and Stephens were motivated differently, yet they ended with their famous[1] corner detector [1].

**Few questions arise:**

- why corners?
- what was the motivation for corners?
- what are corners, actually?

[1] 3635 times cited according to Google scholar on 2009-10-26.

## Image matching



- uniqueness
- repeatability

Derivation of the corner detector on the blackboard. Details can be found in [1]. Its implementation with demonstration of use in chapter 5 of [2].

## Image derivatives

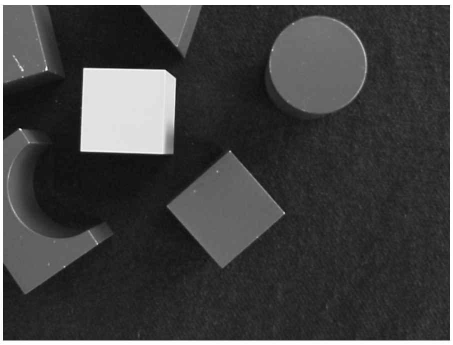
In discrete space we aproximate them by local differences:

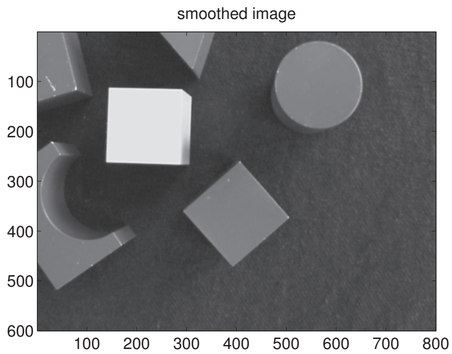
$$
\begin{aligned}
\frac{\partial I}{\partial x}(x,y) &= I(x+1,y) - I(x-1,y) \\
\frac{\partial I}{\partial y}(x,y) &= I(x,y+1) - I(x,y-1)
\end{aligned}
$$

See the on-line demonstration in Matlab



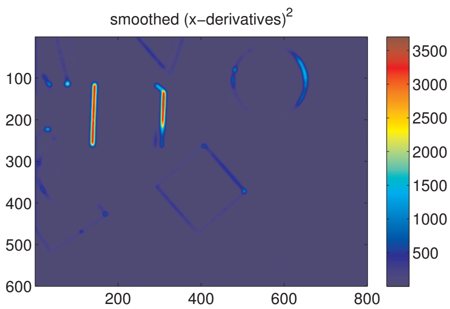
## Corner detection in pictures, images

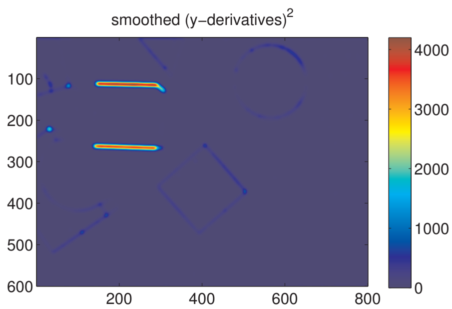


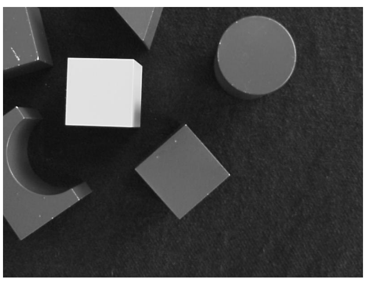





## Corner detection in pictures, (smoothed) derivatives

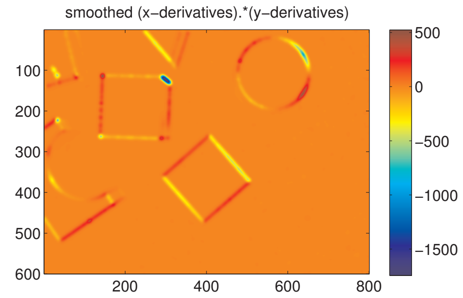


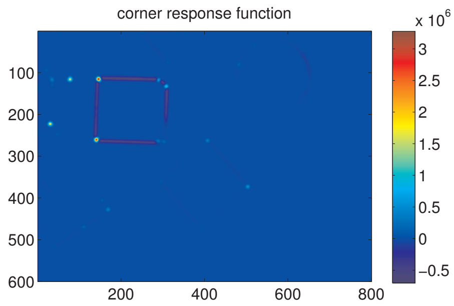



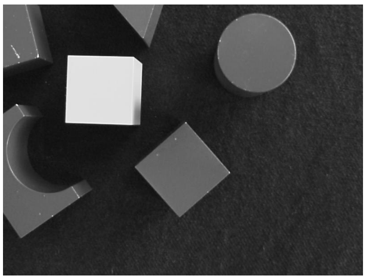



# Corner detection in pictures, mixed derivatives and reponse function



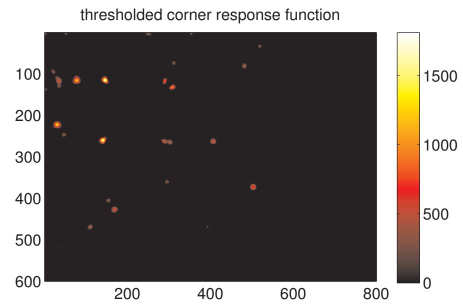


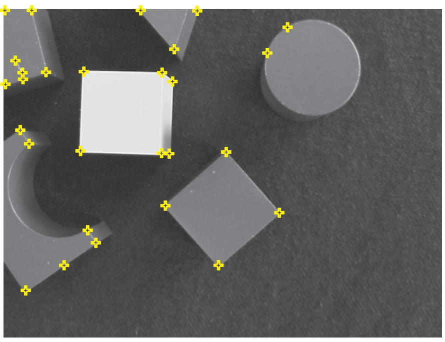



# Corner detection in pictures, thresholding and corners





# References



[1] C. Harris and M. Stephen. A combined corner and edge detection. In M. M. Matthews, editor, Proceedings of the 4th ALVEY vision conference, pages 147–151, University of Manchaster, England, September 1988. on-line copies available on the web.

[2] Tomáš Svoboda, Jan Kybic, and Václav Hlaváč. Image Processing, Analysis and Machine Vision. A MATLAB Companion. Thomson, 2007. Accompanying www site http://visionbook.felk.cvut.cz.

## End