GPU Programming in Computer Vision

Summer Semester 2015

Thomas Möllenhoff, Robert Maier, Caner Hazirbas

The structure tensor of an image

Given an input image $u : \Omega \to \mathbb{R}^k$, compute the smoothed version as $S := G_\sigma * u$.

The *structure tensor* T of u is defined at each pixel (x, y) as the smoothing

$$T := G_\rho * M$$

of the matrix

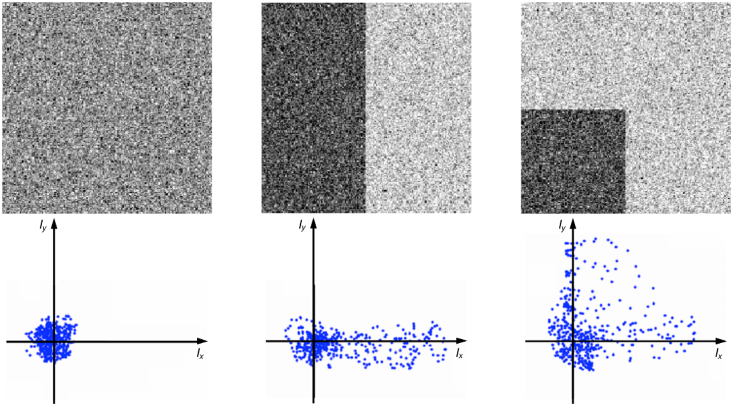
$$M := \nabla S \cdot \nabla S^\top = \begin{pmatrix} (\partial_x S)^2 & (\partial_x S)(\partial_y S) \\ (\partial_x S)(\partial_y S) & (\partial_y S)^2 \end{pmatrix},$$

where $\sigma > 0$ is called the inner scale, $\rho > 0$ the outer scale.

- $T(x, y) \in \mathbb{R}^{2\times 2}$ is symmetric and positive definite. It has two non-negative eigenvalues.
- How do its eigenvalues and eigenvectors look like?

Interpretation of the structure tensor

Consider the local distribution of partial derivatives around edges and corners.



Structure tensor as a Covariance Matrix

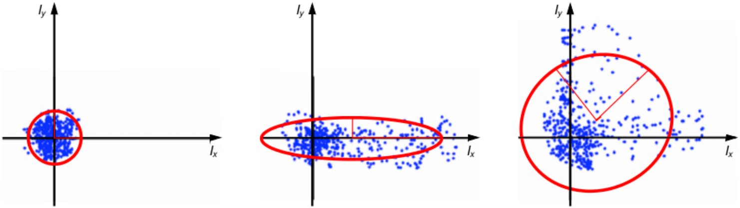
Treat $\partial_x S$ and $\partial_y S$ as random variables and assume $\mu_1 := \mathbb{E}[\partial_x S] = 0$ and $\mu_2 := \mathbb{E}[\partial_y S] = 0$.

Since convolution corresponds to taking the (weighted) expected value we have:

$$\begin{aligned}
\mathrm{Cov}(\partial_x S, \partial_y S) &= \begin{pmatrix} \mathbb{E}[(\partial_x S - \mu_1)^2] & \mathbb{E}[(\partial_x S - \mu_1)(\partial_y S - \mu_2)] \\ \mathbb{E}[(\partial_x S - \mu_1)(\partial_y S - \mu_2)] & \mathbb{E}[(\partial_y S - \mu_2)^2] \end{pmatrix} \\
&= \begin{pmatrix} \mathbb{E}[(\partial_x S)^2] & \mathbb{E}[(\partial_x S)(\partial_y S)] \\ \mathbb{E}[(\partial_x S)(\partial_y S)] & \mathbb{E}[(\partial_y S)^2] \end{pmatrix} \\
&= \begin{pmatrix} G_\rho * (\partial_x S)^2 & G_\rho * (\partial_x S)(\partial_y S) \\ G_\rho * (\partial_x S)(\partial_y S) & G_\rho * (\partial_y S)^2 \end{pmatrix} = T.
\end{aligned}$$

Interpretation of the structure tensor

- The fact that the structure tensor is as a covariance matrix allows for an immediate interpretation.
- The eigenvectors are the directions of principal axes and the eigenvalues the length of the principal axes.
- Allows for a simple edge/corner detector ("Harris corners").



$\lambda_1 \approx \lambda_2$ λ_1 large, λ_2 small λ_1, λ_2 large