Gaussian filters

- Remove high-frequency components from the image (low-pass filter)
- Convolution with self is another Gaussian
  - So can smooth with small-$\sigma$ kernel, repeat, and get same result as larger-$\sigma$ kernel would have
  - Convolving two times with Gaussian kernel with std. dev. $\sigma$ is same as convolving once with kernel with std. dev. $\sigma\sqrt{2}$
- **Separable** kernel
  - Factors into product of two 1D Gaussians
  - Discrete example:

\[
\begin{bmatrix}
1 & 2 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1 \\
\end{bmatrix}
= 2 \begin{bmatrix}
1 \\
2 \\
1 \\
\end{bmatrix}
\]

Source: K. Grauman

Why is separability useful?

- Separability means that a 2D convolution can be reduced to two 1D convolutions (one among rows and one among columns)
- What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?
  - $O(n^2 m^2)$
- What if the kernel is separable?
  - $O(n^2 m)$

Separability of the Gaussian filter

\[
G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right)
= \left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{x^2}{2\sigma^2} \right) \right) \left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{y^2}{2\sigma^2} \right) \right)
\]

The 2D Gaussian can be expressed as the product of two functions, one a function of $x$ and the other a function of $y$

In this case, the two functions are the (identical) 1D Gaussian

Source: D. Lowe
The Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin wide “ridges” down to single pixel width
4. Linking and thresholding (hysteresis):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them

MATLAB:  \texttt{edge(image, ‘canny’)};

The Canny edge detector

How to turn these thick regions of the gradient into curves?

Non-maximum suppression

Check if pixel is local maximum along gradient direction, select single max across width of the edge
- requires checking interpolated pixels p and r

The Canny edge detector

Problem: pixels along this edge didn’t survive the thresholding

thinning (non-maximum suppression)

Hysteresis thresholding

Use a high threshold to start edge curves, and a low threshold to continue them.
Hysteresis thresholding

1. Compute x and y gradient images
2. Find magnitude and orientation of gradient
3. **Non-maximum suppression:**
   - Thin wide “ridges” down to single pixel width
4. **Linking and thresholding (hysteresis):**
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them

MATLAB: `edge(image, 'canny');`

Recap: Canny edge detector


Modern edge detection

- Boundary prediction as a machine learning problem
- Lot of early work from the Berkeley vision group

Modern techniques

A very fast version

- Random forest edge detector (Piotr Dollar et al., ICCV 13)

Further thoughts and readings …

- Hybrid images project
  - [http://cvcl.mit.edu/hybridimage.htm](http://cvcl.mit.edu/hybridimage.htm)
- Canny edge detector
- Bilateral filtering for image denoising (and other application)
- If all else fails [www.xkcd.com](http://www.xkcd.com)