EPL668: Computer Vision

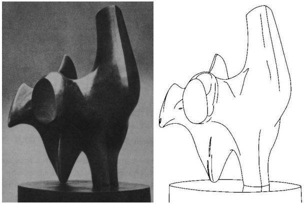
Lecture 3: Edge Detection I

Dr Paris Kaimakis

University of Cyprus
Department of Computer Science
pariskaimakis@cs.ucy.ac.cy

January 2015

It is evident that the human visual system is able to interpret images using a small amount of **edge** and **corner** data. This should be also possible in computer vision.



Edges and corners can be matched in stereo views or tracked over time to deduce the scene's shape. Single images can also be interpreted if adequate assumptions about the scene are made.

With current computer technology, it is necessary to discard most of the data coming from the camera before any attempt is made at real-time image interpretation.

$$\begin{array}{ccccc} \text{images} & & \text{salient features} & & \text{representations} \\ \text{100Mb/s} & \longrightarrow & \text{40kb/s} & \longrightarrow & \text{1-10kb/s} \end{array}$$

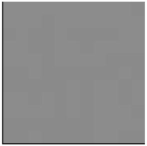
All subsequent interpretation is performed on the generic representation, not the original image.

With data reduction we aim to:

- Discard the redundant information in the images (such as the lighting conditions)
- Preserve the useful information in the images (such as the projected shape of objects in the scene)

In this photo we will examine the pixel data around 3 patches: a featureless region, an edge and a corner.

The featureless region is characterized by a smooth variation of intensities.



140	141	141	140	141	141	140	141	141	141	139
140	141	141	139	140	140	140	140	140	140	140
139	140	139	139	140	139	139	138	139	140	140
140	140	139	139	140	140	138	140	140	140	140
140	140	141	139	141	141	140	140	139	139	138
139	139	141	139	139	139	139	139	139	140	139
139	139	139	139	139	140	139	140	140	139	141
140	140	139	140	139	141	139	140	140	139	140
140	140	140	140	139	140	139	140	131	131	139
139	139	138	139	139	139	139	140	141	140	140
139	139	140	139	139	139	139	140	140	139	139

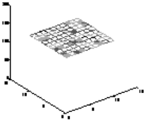
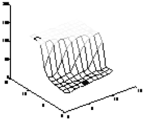


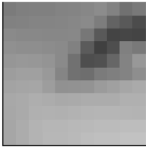
IMAGE STRUCTURE

The patch containing the edge reveals an intensity discontinuity in one direction.

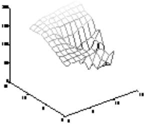
136	136	135	134	135	134	134	135	135	136	135
136	135	134	134	136	135	133	133	135	135	136
136	135	134	134	133	133	134	134	134	134	135
130	133	134	133	132	132	132	133	133	132	133
73	103	127	135	134	133	134	133	132	133	134
54	52	60	88	114	127	133	134	132	133	132
49	48	50	53	56	76	99	117	130	133	135
46	45	45	48	50	53	55	57	77	99	118
44	44	45	46	45	47	50	52	54	49	54
42	43	43	44	46	47	50	51	50	52	55
41	40	44	44	44	46	49	48	50	51	56



The patch containing the corner reveals an intensity discontinuity in two directions.

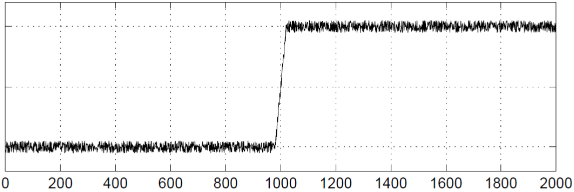


124	127	128	128	131	312	135	140	130	113	121
129	130	132	132	133	135	139	125	99	89	76
138	136	137	135	135	136	130	89	68	71	69
144	142	143	141	139	129	92	64	78	128	121
150	152	151	148	138	113	79	81	102	131	152
158	160	160	154	133	113	111	123	127	131	143
163	165	166	158	145	146	147	155	160	166	171
168	171	174	173	169	171	172	173	173	176	176
167	171	176	177	176	178	180	181	181	180	179
166	173	179	182	182	184	185	187	188	189	190
166	173	179	183	183	185	189	191	191	194	194



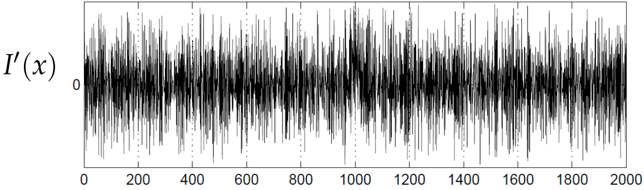
Real-world images always contain **noise**. Consider this signal $I(x)$ with an obvious edge:

$I(x)$



An intuitive approach to detect edges could be to look for maxima and minima in $I'(x)$...

Real-world images always contain **noise**. Consider this signal $I(x)$ with an obvious edge:

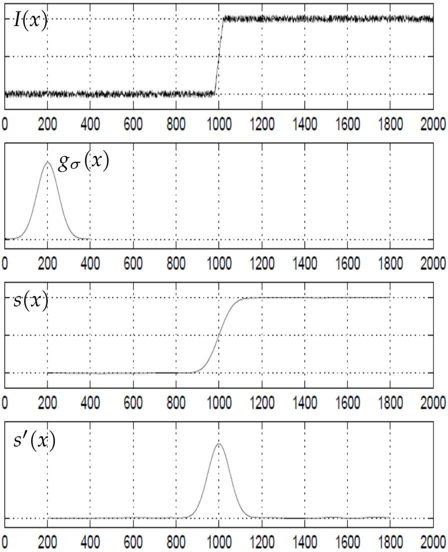


…but this simple strategy is defeated by noise. For this reason, all edge detectors start by **smoothing** the signal to suppress noise. This is done by **convolution** with a Gaussian kernel.

1. Construct a Gaussian kernel:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

2. Convolve $I(x)$ with $g_\sigma(x)$ to produce a smoothed signal $s(x)$.
3. Compute $s'(x)$, the derivative of $s(x)$.
4. Find maxima and minima of $s'(x)$.
5. Use thresholding on the magnitude of the extrema to mark edges.



The smoothing in **step 2** is performed by a 1D convolution:

$$
\begin{aligned}
s(x) = g_\sigma(x) * I(x) &= \int_{-\infty}^{+\infty} g_\sigma(X)\, I(x - X)\, dX \\
&= \int_{-\infty}^{+\infty} g_\sigma(x - X)\, I(X)\, dX
\end{aligned}
$$

The differentiation in **step 3** is also performed by a 1D convolution.

So *at first glance*, edge detection appears to require **two** convolutions, which is quite computationally expensive.

However, it can be shown that

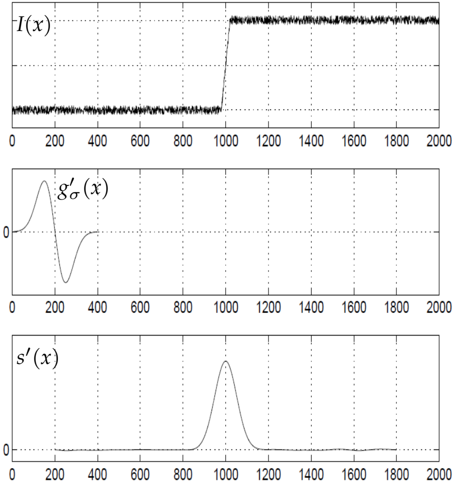
$$\frac{d}{dx}\left[g_\sigma(x) * I(x)\right] = g'_\sigma(x) * I(x)$$

so we can compute $s'(x)$ directly from $I(x)$ by convolving only **once** — a considerable computational saving.

1. Construct the derivative of a Gaussian:

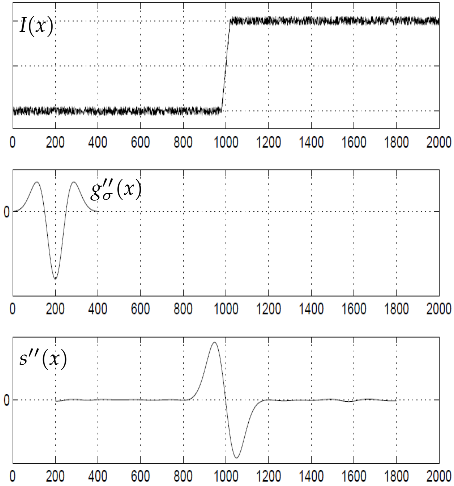
$$g'_\sigma(x) = -\frac{x}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

2. Convolve $I(x)$ with $g'_\sigma(x)$ to directly produce $s'(x)$.
3. Find maxima and minima of $s'(x)$.
4. Use thresholding on the magnitude of the extrema to mark edges.

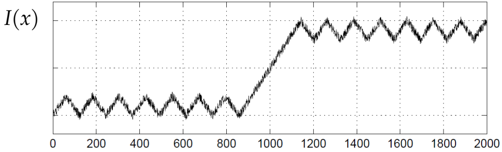


Looking for maxima and minima of $s'(x)$ is the same as looking for zero-crossings of $s''(x)$. In some variants of the edge detection algorithm, the signal is convolved with the **Laplacian** of a Gaussian, $g''_\sigma(x)$:

$$s''(x) = g''_\sigma(x) * I(x)$$

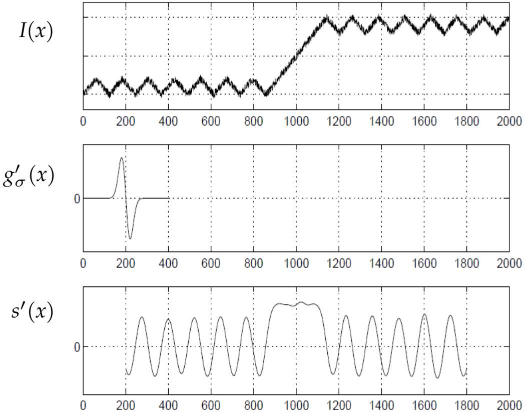


We have not yet addressed the issue of what value of σ to use. Consider this signal:

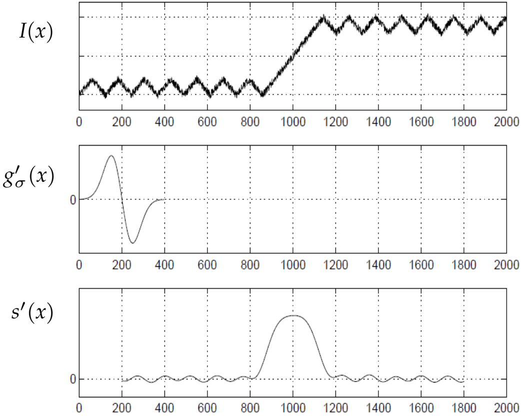


Does the signal have one positive edge, or a number of positive and negative edges?

Using a small σ brings out all the edges.



As σ increases, the signal is smoothed more and more, and only the central edge survives.



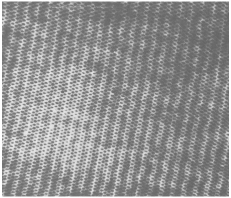
The amount of smoothing controls the **scale** at which we analyze the image. There is no right or wrong size for the Gaussian kernel, it all depends on the scale we are interested in.

Modest smoothing (a Gaussian kernel with small σ) brings out edges at a fine scale. More smoothing (larger σ) identifies edges at larger scales, suppressing the finer detail.

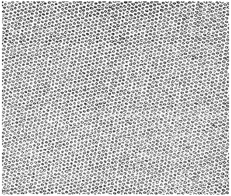
1D Edge Detection

Multiscale Edge Detection: 2D Example

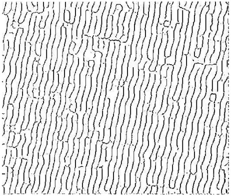
On the left is an image of a dish cloth. After edge detection, we see different features at different scales σ.



raw



$\sigma = 1$



$\sigma = 5$