Image Gradients

Given a discrete image $I(\vec{n})$, consider the smoothed continuous image $B(\vec{x})$ defined by

$$B(\vec{x}) \; = \; G(\vec{x}; \sigma_r^2) * I(\vec{n}) \equiv \sum_{\vec{k}} G(\vec{x} - \vec{k}; \sigma_r^2) I(\vec{k}), \tag{1}$$

where $G(\vec{x}; \sigma_r^2) = \frac{1}{2\pi\sigma_r^2} e^{-\frac{|\vec{x}|^2}{2\sigma_r^2}}$. Here $|\vec{x}|$ is the 2-norm for the vector $\vec{x} = (x, y)^T$. That is, $|\vec{x}| = \sqrt{x^2 + y^2}$.

Note that $G(\vec{x})$ does not satisfy the interpolation conditions $G(\vec{0}) = 1$ and $G(\vec{n}) = 0$ for integer valued $\vec{n} \neq \vec{0}$. Therefore $B(\vec{x})$ does not in general interpolate the original discrete image $I(\vec{n})$ (i.e., generally $B(\vec{n}) \neq I(\vec{n})$ for integer valued image coordinates $\vec{n}$). Instead $B(\vec{x})$ provides a smoothed approximation of the image $I(\vec{n})$ at $\vec{x} = \vec{n}$.

The gradient of a smooth image $B(\vec{x})$ is defined to be the vector of partial derivatives,

$$\vec{\nabla} B(\vec{x}) \; \equiv \; (\frac{\partial B}{\partial x}(\vec{x}), \; \frac{\partial B}{\partial y}(\vec{x}))^T. \tag{2}$$

By differentiating inside the sum in (1) we find

$$\frac{\partial B}{\partial x}(\vec{x}) \quad = \quad \sum_{\vec{k}} \frac{\partial G}{\partial x}(\vec{x} - \vec{k}) I(\vec{k}), \tag{3}$$

$$\frac{\partial B}{\partial y}(\vec{x}) \quad = \quad \sum_{\vec{k}} \frac{\partial G}{\partial y}(\vec{x} - \vec{k}) I(\vec{k}). \tag{4}$$

Note the derivative of a 2D Gaussian is the separable product of a 1D Gaussian times the derivative of a 1D Gaussian, as in

$$\begin{aligned} \frac{\partial G}{\partial x}(\vec{x}; \sigma_r^2) &= -\frac{x}{\sigma_r^2} G(\vec{x}; \sigma_r^2) \\ &= \left[-\frac{x}{\sigma_r^2} G(x; \sigma_r^2) \right] \left[G(y; \sigma_r^2) \right], \end{aligned}$$

where $G(x; \sigma_r^2) = \frac{1}{\sqrt{2\pi}\sigma_r} e^{-x^2/(2\sigma_r^2)}$ is a one-dimensional Gaussian. Therefore separable convolution can be used to compute the image gradient according to equations (2), (3) and (4) above.

Implementation Details. Typically the radius K of the discrete filter support is taken to be $K = 3\sigma_r$. This gives 1D filter kernels of length $2K + 1$. Moreover, in order to avoid strong discretization artifacts in sampling the Gaussian, typically $\sigma_r \geq 1$ is used. The smallest gradient filters of this type are therefore 7×1 and 1×7, which are used for $\sigma_r = 1$ (see cannyTutorial.m).

Properties of Gradients

What does the image gradient $\vec{\nabla} B(\vec{x})$ tell us about the local image brightness? To understand this, consider the directional derivative of the image at $\vec{x}$ in the direction $\vec{u} = (u_1, u_2)^T$, defined by

$$\begin{aligned} D_{\vec{u}} B(\vec{x}) &\equiv \frac{dB}{dt}(\vec{x} + \vec{u}t)|_{t=0}, \\ &= \left[\frac{\partial B}{\partial x}(\vec{x} + \vec{u}t)\frac{d}{dt}(x + u_1 t) + \frac{\partial B}{\partial y}(\vec{x} + \vec{u}t)\frac{d}{dt}(y + u_2 t)\right]\Bigg|_{t=0} \quad \text{by the chain rule,} \\ &= \left(\vec{\nabla} B(\vec{x})\right)^T \vec{u}. \end{aligned} \tag{5}$$

Therefore, given the gradient, we can easily compute the directional derivative in any direction $\vec{u}$.

Note that from equation (5) we see $D_{\vec{u}} B(\vec{x}) = 0$ for directions $\vec{u}$ orthogonal to the gradient $\vec{\nabla} B(\vec{x})$. The image gradient is therefore orthogonal to curves of constant intensity, i.e. contours satisfying $B(\vec{x}(t)) = c$, for any constant c.

The steepest ascent direction $\vec{u}$ (at a particular value of $\vec{x}$) is defined to be the unit vector which maximizes the directional derivative $D_{\vec{u}} B(\vec{x})$. From equation (5) this steepest ascent direction is given by

$$\vec{u} = \vec{\nabla} B(\vec{x}) / |\vec{\nabla} B(\vec{x})|, \tag{6}$$

where $|\vec{w}| = \sqrt{w_1^2 + w_2^2}$ denotes the Euclidean norm (i.e., 2-norm). Thus the gradient points in the steepest ascent direction.

If the gradient $\vec{\nabla} B(\vec{x}) = \vec{0}$, then $\vec{x}$ is said to be a stationary point of $B(\vec{x})$. Typically this is a local minimum, maximum or saddle point in $B(\vec{x})$. At a stationary point $\vec{x}$, the directional derivative $D_{\vec{u}} B(\vec{x}) = 0$ for any direction $\vec{u}$, and therefore the steepest ascent direction is undefined at such an $\vec{x}$ (i.e., a divide by zero occurs in equation (6)).

2D Edge Detection

We extend our approach to 1D edge detection to 2D images by considering the variation of image brightness in particular directions $\vec{u}$. That is, at a pixel $\vec{x}$, we consider the variation along a 1D slice, $I(\vec{x} + \vec{u}t)$, in the neighbourhood of $t = 0$.

The direction of this slice is chosen to be the steepest ascent direction at each pixel, as given by the direction of the image gradient $\vec{R}(\vec{x})$:

$$\vec{\mathbf{u}}(\vec{\mathbf{x}}) = \frac{\vec{\mathbf{R}}(\vec{\mathbf{x}})}{|\vec{\mathbf{R}}(\vec{\mathbf{x}})|}.$$

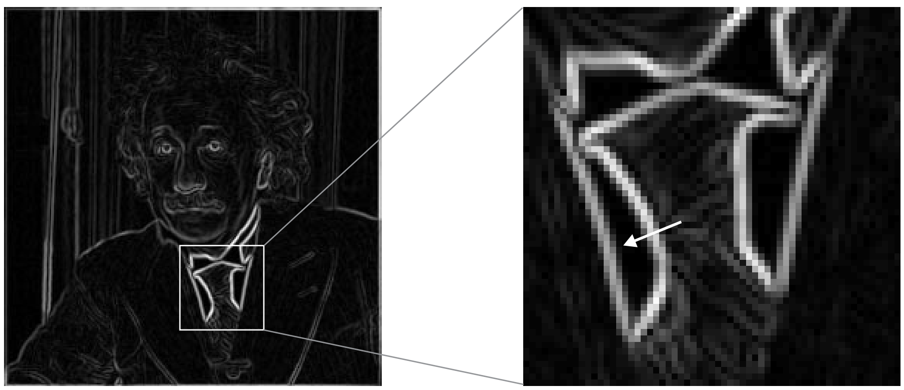
As described in the previous notes, this gradient can be estimated by differentiating a Gaussian blurred and interpolated approximation of the image,

$$\vec{\mathbf{R}}(\vec{\mathbf{x}}) = \vec{\nabla} G(\vec{\mathbf{x}}; \sigma_r^2) * I(\vec{\mathbf{x}})$$

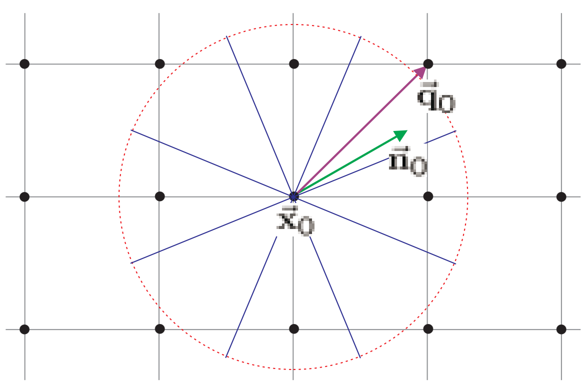
Image Edgel Detection: Recall that in 1D we detected edges by identifying local maxima in the absolute value of the response of a derivative of Gaussian filter applied to the signal. The analogous operation in 2D is to search for maxima in the directional image derivative taken in the gradient direction $\vec{\mathbf{u}}(\vec{\mathbf{x}})$. Since the gradient direction $\vec{\mathbf{u}}(\vec{\mathbf{x}})$ is perpendicular to curves of constant brightness, we take any detected edgel to have normal $\vec{\mathbf{n}}$ given by the gradient direction, that is, $\vec{\mathbf{n}}(\vec{\mathbf{x}}) = \vec{\mathbf{u}}(\vec{\mathbf{x}})$.

2D Edge Detection (cont.)

Search for local maxima of gradient magnitude $S(\vec{\mathbf{x}}) = |\vec{\mathbf{R}}(\vec{\mathbf{x}})|$, in the direction normal to local edge, $\vec{\mathbf{n}}(\vec{\mathbf{x}})$, suppressing all responses except for local maxima (called non-maximum suppression).



In practice, the search for local maxima of $S(\vec{\mathbf{x}})$ takes place on the discrete sampling grid. Given $\vec{\mathbf{x}}_0$, with normal $\vec{\mathbf{n}}_0$, compare $S(\vec{\mathbf{x}}_0)$ to nearby pixels closest to the direction of $\pm\vec{\mathbf{n}}_0$, e.g., pixels at $\vec{\mathbf{x}}_0 \pm \vec{\mathbf{q}}_0$, where $\vec{\mathbf{q}}_0$ is $\frac{1}{2\sin(\pi/8)}\vec{\mathbf{n}}_0$ with each of its coefficients rounded to the nearest integer.



The red circle depicts points $\vec{\mathbf{x}}_0 \pm \frac{1}{2\sin(\pi/8)}\vec{\mathbf{n}}_0$. Normal directions between (blue) radial lines all map to the same neighbour of $\vec{\mathbf{x}}_0$.

Canny Edge Detection

Algorithm:

1. Convolve with gradient filters (possibly at multiple scales σ_r)

$$\vec{\mathbf{R}}(\vec{\mathbf{x}}) \;\equiv\; (R_1(\vec{\mathbf{x}}),\, R_2(\vec{\mathbf{x}})\,)^T \;=\; \vec{\nabla} G(\vec{\mathbf{x}};\, \sigma_r^2) \,*\, I(\vec{\mathbf{x}})\;.$$

2. Compute response magnitude, $S(\vec{\mathbf{x}}) = \sqrt{R_1^2(\vec{\mathbf{x}}) + R_2^2(\vec{\mathbf{x}})}$.

3. Compute local edge orientation (represented by unit normal):

$$\vec{\mathbf{n}}(\vec{\mathbf{x}}) \;=\; \begin{cases} (R_1(\vec{\mathbf{x}}),\, R_2(\vec{\mathbf{x}}))/S(\vec{\mathbf{x}}) & \text{if } \; S(\vec{\mathbf{x}}) > threshold \\ \vec{0} & \text{otherwise} \end{cases}$$

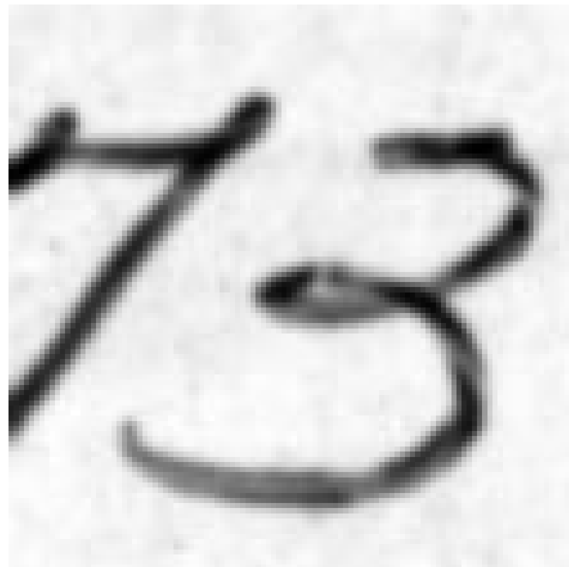
4. Peak detection (non-maximum suppression along edge normal)

Extensions: In order to select an appropriate scale σ_r for an edgel, non-maximum suppression can also be done across neighbouring scales. Also the simple thresholding described above can be replaced by hysteresis thresholding along edges (see Canny (1986) for details). These are beyond the scope of this course.

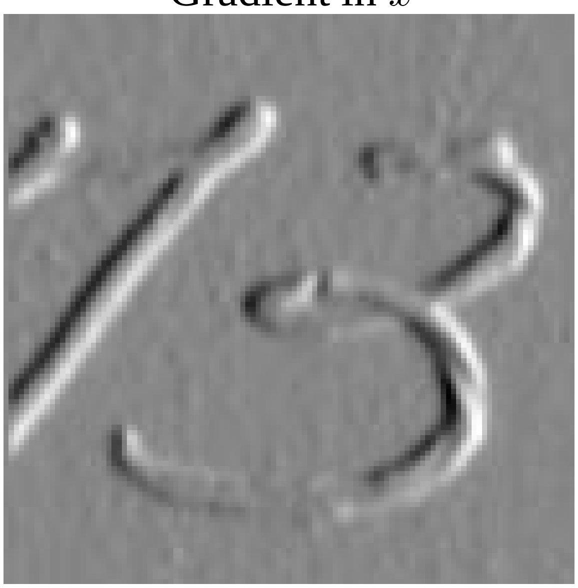
Filtering with Derivatives of Gaussians

Image `three.pgm`

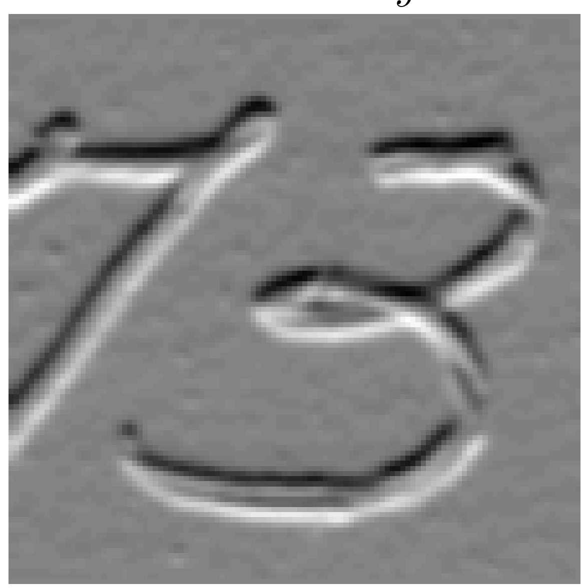
Gaussian Blur $\sigma = 1.0$



Gradient in x



Gradient in y



Canny Edgel Measurement

Gradient Strength

Gradient Orientations

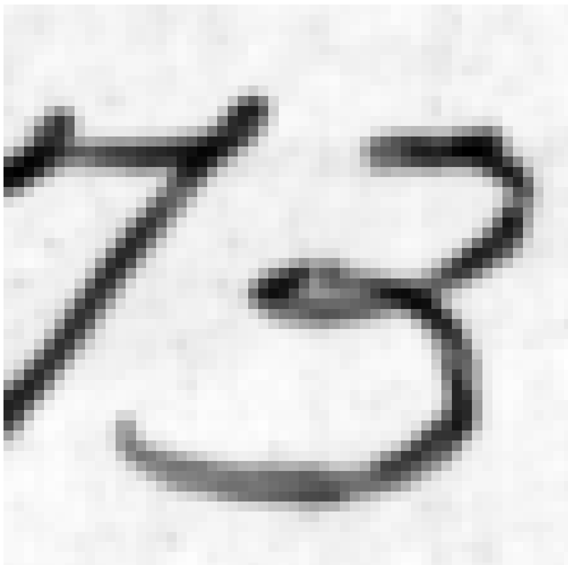
Canny Edgels

Edgel Overlay

Colour gives gradient direction (red – $0°$; blue – $90°$; green – $270°$)

Gaussian Pyramid Filtering (Subsample $\times$ 2)

Blurred and Down-Sampled ($\times 2$)



Gaussian Blur $\sigma = 1.0$

Gradient Magnitude (dec $\times 2$)

Gradient Orientations

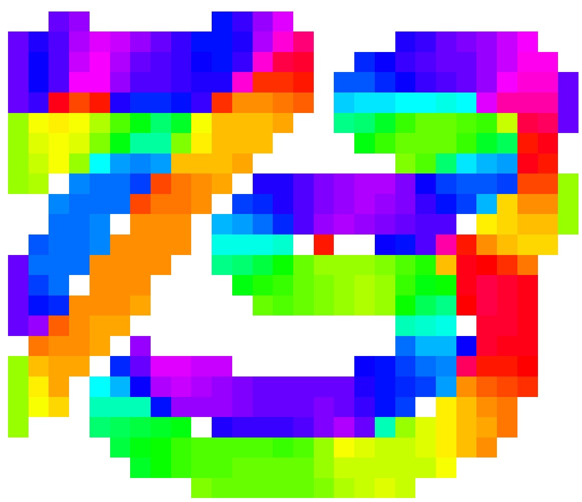
Gaussian Pyramid Filtering (Subsample $\times$ 4)

Blurred and Down-Sampled ($\times 4$)

Gaussian Blur σ = 1.0

Gradient Magnitude (dec $\times 4$)

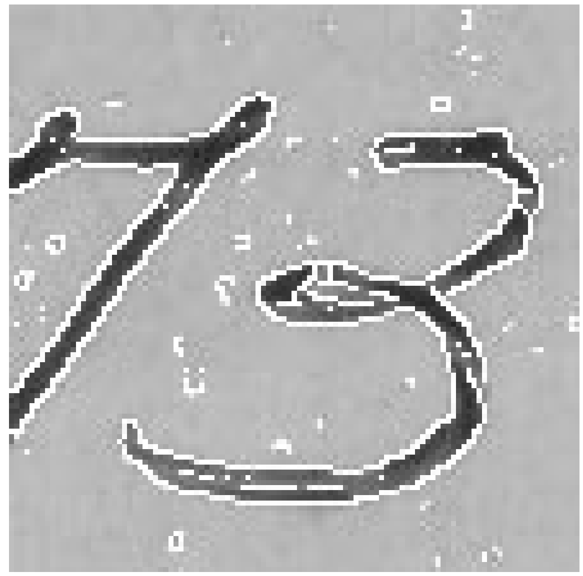
Gradient Orientations



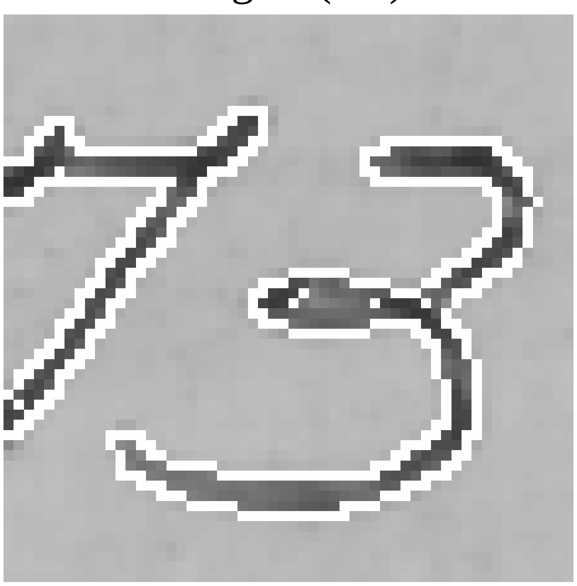
Multiscale Canny Edgels

Image `three.pgm`

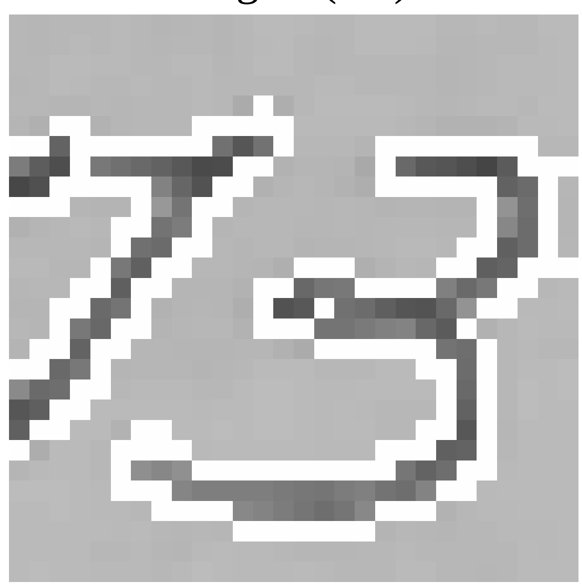
Edgels (x 1)



Edgels (x 2)



Edgels (x 4)



Edge-Based Image Editing

Existing edge detectors are sufficient for a wide variety of applications, such as image editing, tracking, and simple recognition.

[from Elder and Goldberg (2001)]

Approach:

1. Edgels represented by location, orientation, blur scale (min reliable scale for detection), and brightness on each side.
2. Edgels are grouped into curves (i.e., maximum likelihood curves joining two edge segments specified by a user.)
3. Curves are then manipulated (i.e., deleted, moved, clipped etc).
4. The image is reconstructed from edgel positions and the image brightnesses each side.

Further Readings

Castleman, K.R., **Digital Image Processing**, Prentice Hall, 1995

John Canny, ”A computational approach to edge detection.” *IEEE Transactions on PAMI*, 8(6):679–698, 1986.

James Elder and Richard Goldberg, ”Image editing in the contour domain.” *IEEE Transactions on PAMI*, 23(3):291–296, 2001.