

Canny Edges Tutorial: Oct. 4, '03

Canny Edges Tutorial

References:

- `imageTutorial.m`
- `cannyTutorial.m`
- `~jepson/pub/matlab/iseToolbox/tutorials`
- `~jepson/pub/matlab/utvisToolbox/tutorials`
- Canny, J. 'A Computational Approach to Edge Detection', *Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, November, 1986, pp. 679-698

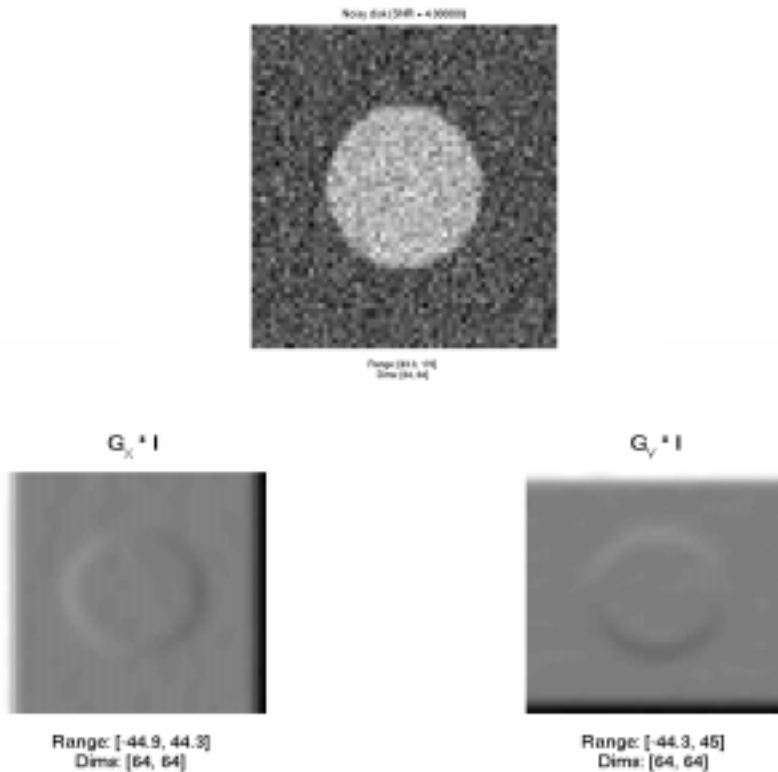
Canny Edge Detection steps

The process of edge extraction is formed by the following (general) steps:

- Convolution of the image with edge enhancing masks in the x and y directions.
- Computation of image gradient (magnitude and direction)
- Thresholding of the gradient image
- Non-maximum edgel suppression

Image Gradient

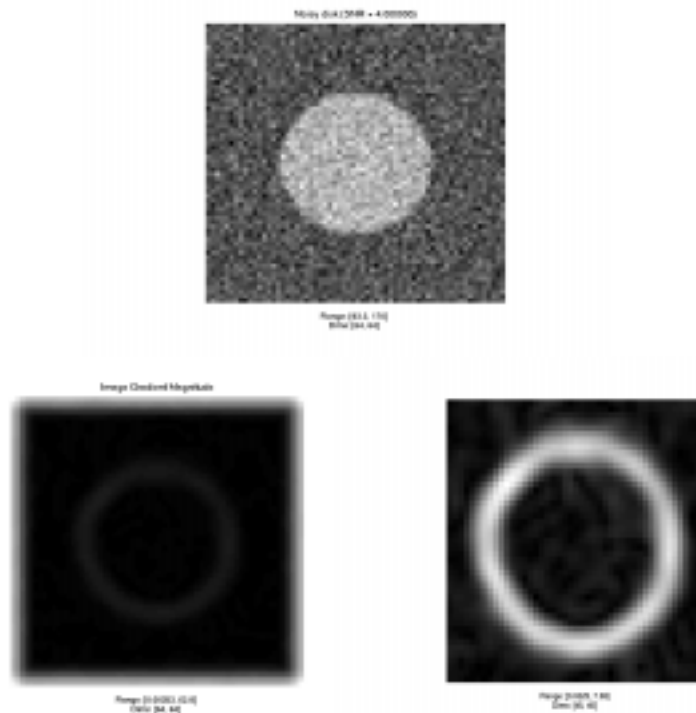
The convolution of the image with derivative filters in the x and y directions yields the x and y components of the image gradient $\nabla I = [I_x I_y] = A(x, y)e^{-i(\theta(x, y))}$.



The amplitude and orientation of the gradient are computed directly from the image derivatives along x and y .

Gradient magnitude

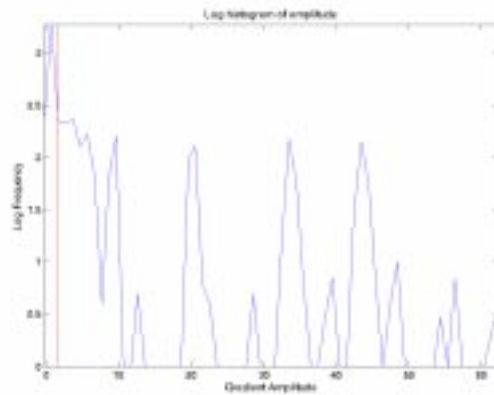
The magnitude of the gradient at pixel $p(x, y)$ is given by $A(x, y) = \sqrt{I_x^2 + I_y^2}$, it gives an indication of the strength of a possible edge at $p(x, y)$.



The figure on the lower-right shows a section of the gradient magnitude image on the lower-left, with pixel brightness rescaled to better show gradient magnitude around the circle.

Gradient thresholding

Once the gradient magnitude has been computed, a threshold is applied to remove all weak responses due to noise.



Norm of Image Gradient



Range: [0.00353, 62.8]
Dims: [64, 64]

Thresholded Region



Range: [0, 1]
Dims: [64, 64]

The resulting binary image is where the actual search for edgels takes place.

Gradient orientation

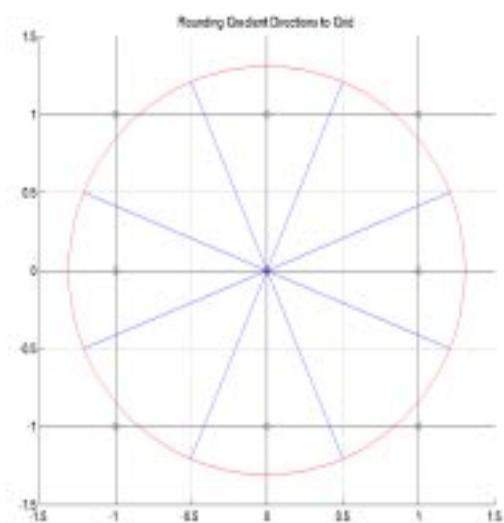
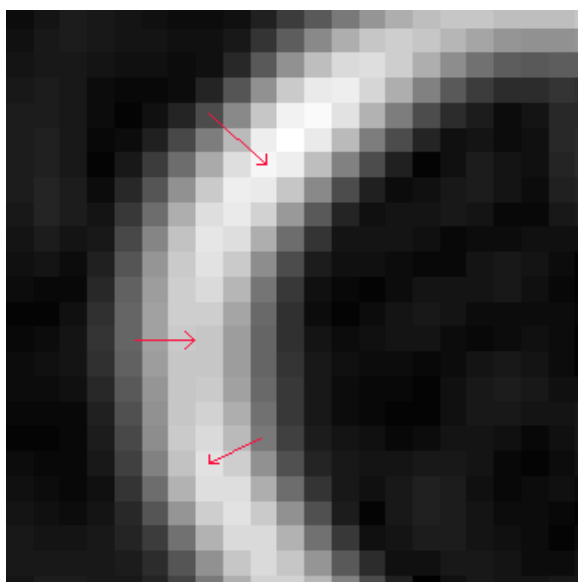
Besides the thresholded gradient magnitude map, we require an estimate of the gradient direction at each pixel, $\theta(x, y) = \arctan(I_y/I_x)$. The gradient direction is always perpendicular to the direction of an edge passing through $p(x, y)$.



The gradient orientation is only computed at image locations that passed the gradient magnitude threshold.

Non-Maximal Suppression

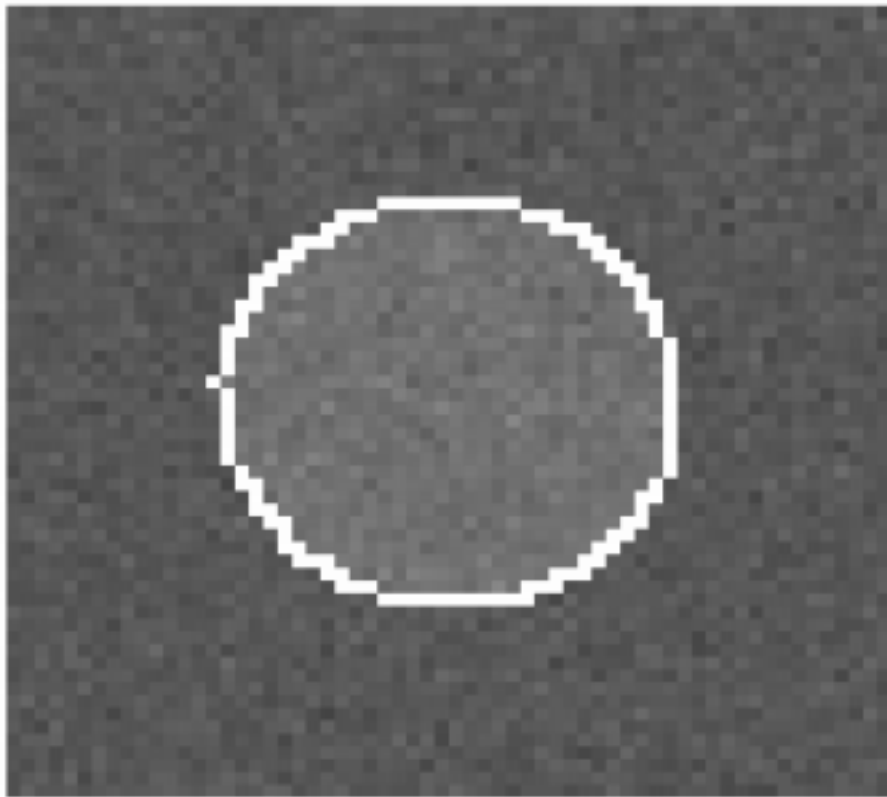
Real edges correspond to places where the gradient magnitude is maximal, other locations along the gradient direction, with non-zero but not maximal responses must be discarded.



Non maximal suppression looks along the gradient direction, suppressing any edgel locations with non-maximal response.

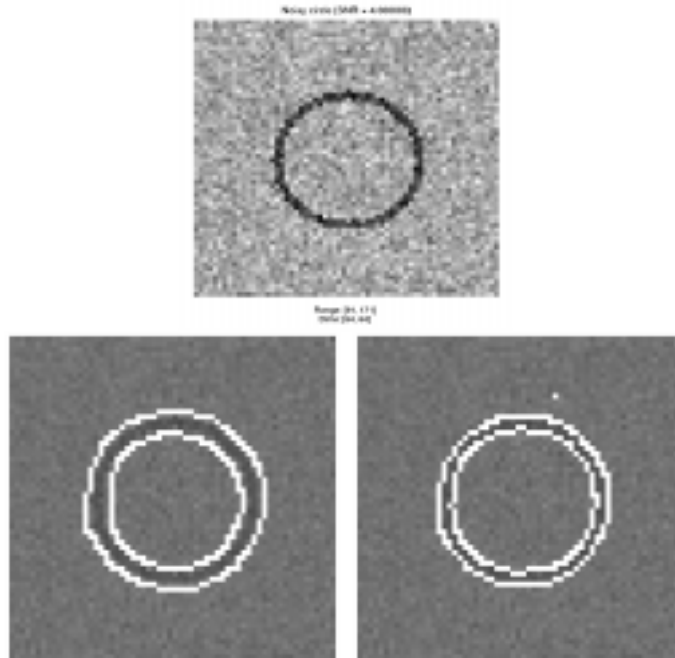
The final edge image

Edgels that remain after non-maximal suppression make up the final edge map.



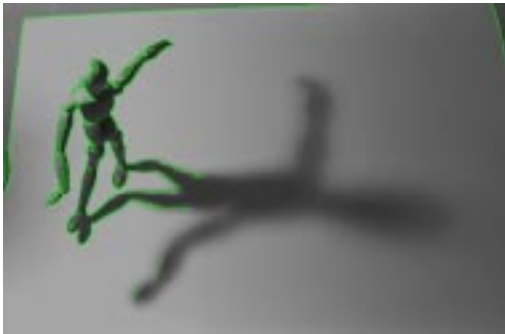
Choice of sigma and level of detail

Smaller sigma values cause the derivative filters to respond to smaller features, but also make the filters more sensitive to noise. Conversely, larger sigma values decrease localization accuracy.

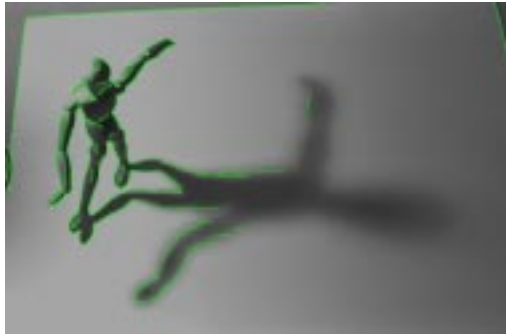


The edges on the lower-left correspond to $\sigma = 2$, the edges on the lower right correspond to $\sigma = 1$. The value of σ determines the scale of the edges that are detected.

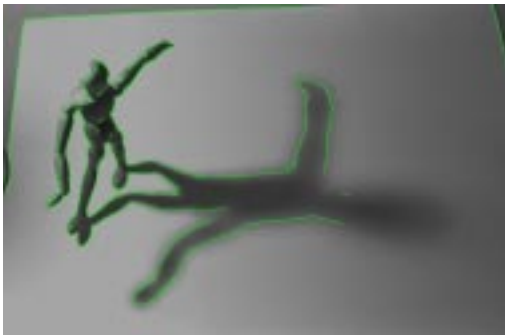
Choice of sigma and level of detail



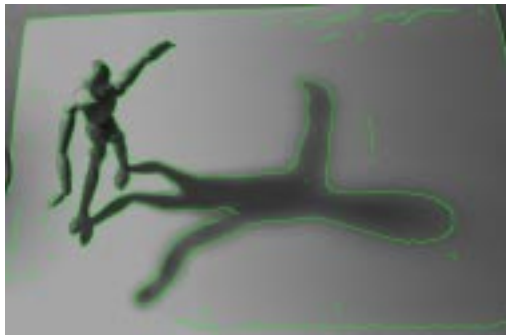
$\sigma=1$



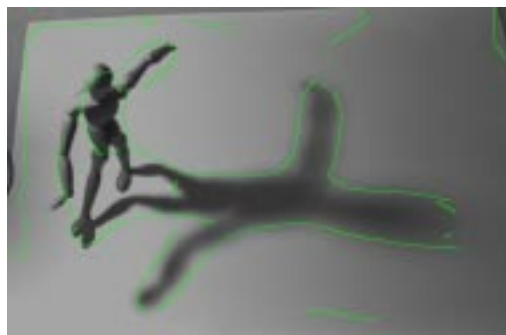
$\sigma=2$



$\sigma=4$



$\sigma=6$

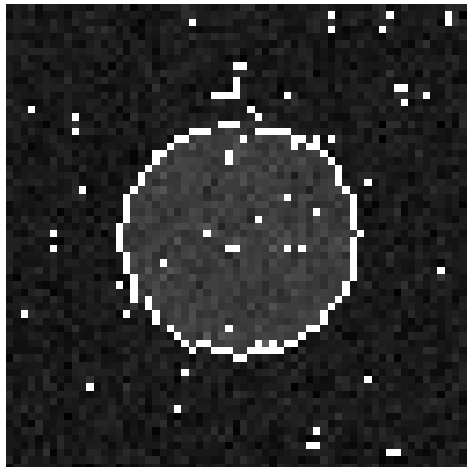
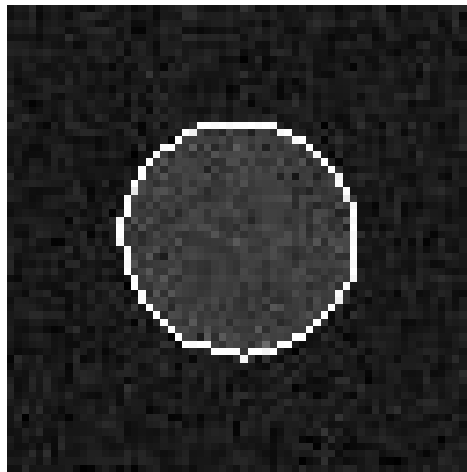


$\sigma=20$

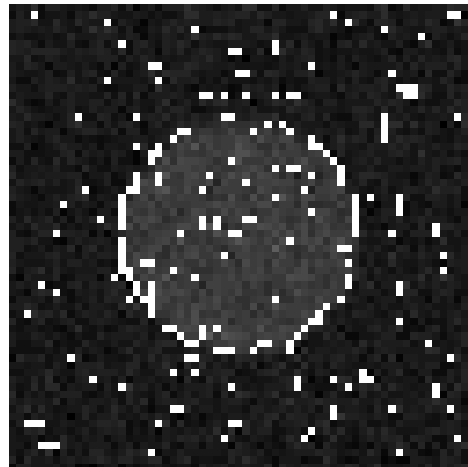
Other edge extraction operators

Instead of using a Gaussian and its derivatives, we could use other standard filter masks such as those defined by Sobel or Roberts.

Canny



Sobel, $[1\ 2\ 1; 0\ 0\ 0; -1\ -2\ -1]$ and its transpose



Roberts, $[1\ 0; 0\ -1]$ and its transpose

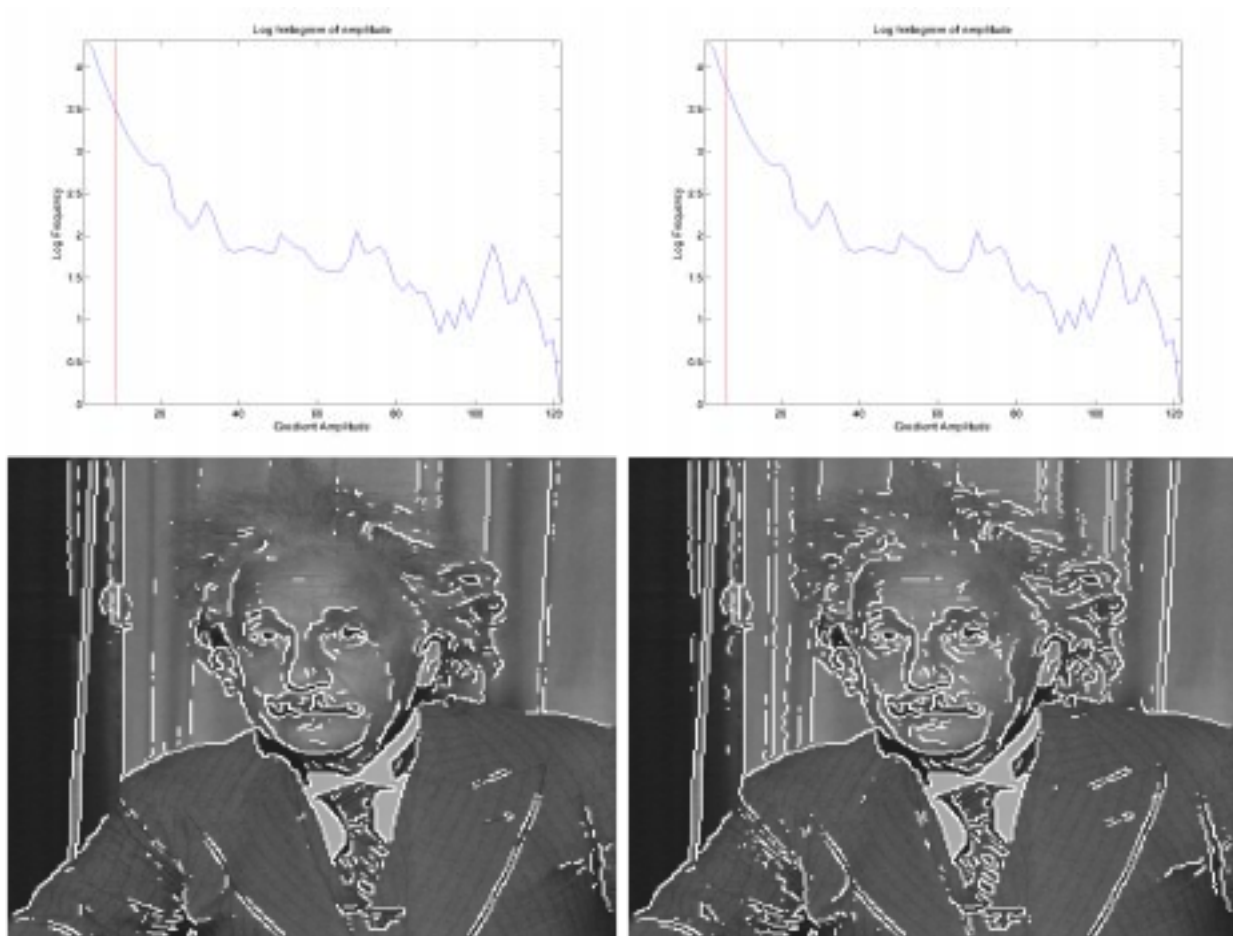
Common issues with Canny edges

- Threshold selection
- False positives
- False negatives
- Double edges
- How to determine edge saliency?



Threshold selection

A small change in threshold can make a large difference in the resulting edgel map.



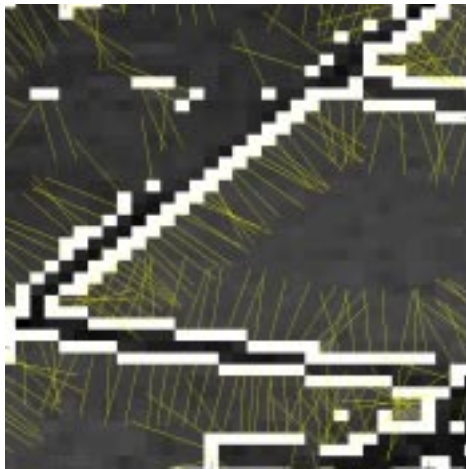
Determining edge saliency

A very difficult open problem. Edge saliency is related to the likelihood that a given edge belongs to an object boundary.

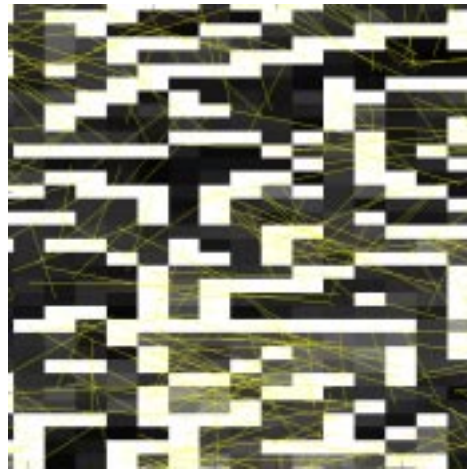
- Can this be evaluated locally?
- How about edges from shadows, illumination changes, and texture?
- Canny enhancement: Hysteresis threshold
- We can do a bit better with some clever filtering!

The Orientation Tensor

The gradient orientation behaves in different ways depending on whether an edgel is located within a textured region or not.



Gradient directions for a region containing lines

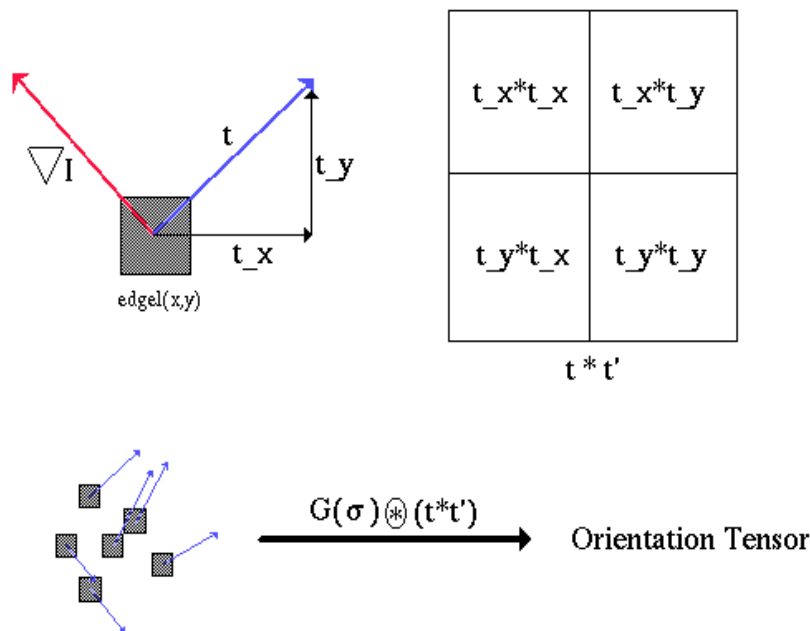


Gradient directions for a region containing texture

Notice that for edgels that lie along a line, the gradient orientation is very similar. Conversely, for edgels that arise from texture or noise, gradient directions look random.

The Orientation Tensor

The Orientation Tensor at each edgel is given by $G(\sigma) * (t \times t')$, where t is a column vector with the tangent direction at each edgel (the direction perpendicular to the gradient at that edgel).



The Orientation Tensor at edgel (x, y) , then, is a combination of the terms $t \times t'$ for edgels around (x, y) , the contribution of each surrounding edgel is weighted

by a Gaussian with standard deviation σ , centered at (x, y) .

The Orientation Tensor

The Orientation Tensor at each edgel is represented with a 2×2 matrix, it encodes the combination of edgel directions in the vicinity of edgel (x, y) . The **trace** (sum of the diagonal elements) of the Orientation Tensor is indicative of the density of edgels around a local neighborhood.



Bench edgels

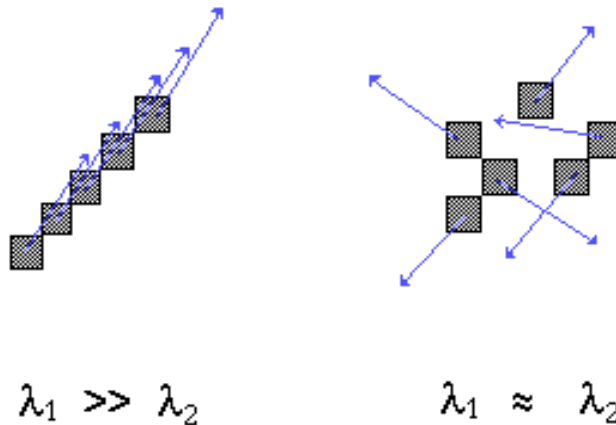


Trace of the Orientation Tensor

Notice the trace of the Orientation Tensor is large where there is a high density of edgels.

The Orientation Tensor

The 2 eigenvalues of the Orientation Tensor at each edgel are indicative of the degree of correlation between edgel directions in the local neighborhood. If there is a dominant direction for edgels in the neighborhood of (x, y) , the Orientation Tensor at (x, y) will have one large eigenvalue, and one small eigenvalue. Conversely, when both eigenvalues have roughly the same magnitude, there is no preferred orientation.



The above provides an indication of whether an edgel is likely to have originated from noise or image texture.

The Orientation Tensor

The normalized difference between the eigenvalues indicates how uniformly directed a region is (the region's directedness), the normalized average of the eigenvalues indicates how uniformly scattered the directions in a region are (the region's randomness).



Directedness

Randomness

The Orientation Tensor

We can combine both images into a colored plot that shows regions with a dominant direction in red, and regions with large randomness in green.



The Orientation Tensor

- We can use the orientation tensor as a rough indicator of whether a pixel is within a textured region or not
- Have we solved the saliency problem?
- How about texture that has a dominant orientation?
- Still local!
- Still an open problem, in general, we do the best job we can at edge extraction, and then turn the job over to more complex feature extraction algorithms (e.g. line and curve fitting), and perceptual grouping.