Image Registration:
Preprocessing Operations

Arthur Goshtasby
Wright State University
Image Registration and Fusion Systems

Preprocessing operations

All operations performed on images that improve the registration performance. These include:

- Noise filtering
- Deblurring
- Region extraction
- Edge detection

Noise smoothing

Given image *f(x,y)* and smoothing filter *h(i,j)*, noise smoothing is defined by:

$$\bar{f}(x,y) = \sum_{i=-k}^{k} \sum_{j=-l}^{l} f(x+i, y+j)h(i,j)$$

The intensity of pixel *(x,y)* in the output is obtained from a weighted sum of intensities of pixels at and around *(x,y)* in the input. *h(i,j)* is the weight of pixel *(x+i,y+j)* in the neighborhood of *(x,y)*, and the sum of the weights over all *i* and *j* is1.

Mean filtering

- When the weights defined by the filter are all the same, the operation is known as mean filtering.
- Intensities of all pixels at and around a point in input have the same effect on the intensity at the same point in output.
- This operation is not rotationally invariant if the filter kernel is not circular.

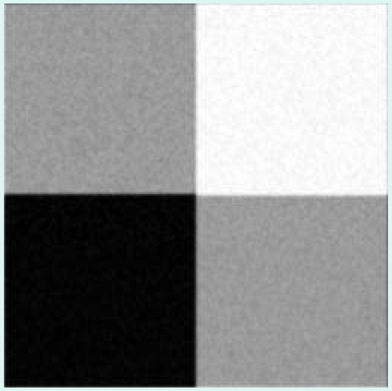
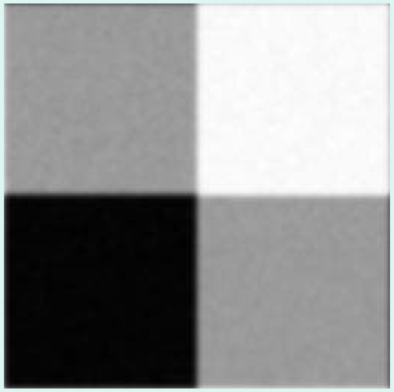
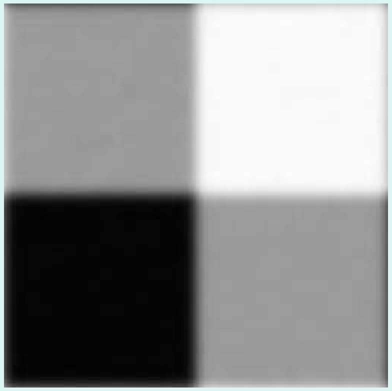


Image containing Zero-mean noise



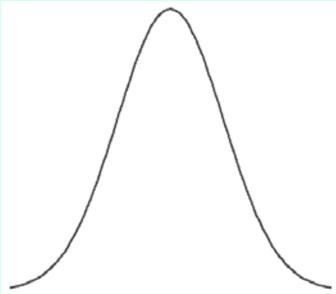
Filter-radius 2 pixels



Filter-radius 4 pixels

Computed using FFT algorithm

Gaussian filtering



- If the weights in filter kernel represent Gaussian coefficients, a Gaussian filter is obtained.
- Gaussian filtering is effective when image noise is zero-mean.
- Gaussian filtering is rotationally invariant.
- A 2-D Gaussian can be decomposed into 2 1-D Gaussians: $G(x,y) = G(x) * G(y)$; therefore, filtering can be carried in 1-D rather than in 2-D

Computation of mean and Gaussian filtering

Although filtering is a convolution operation and can be computed using the FFT algorithm, since FFT considers an image is a periodic signal, if left and right image borders, or top and bottom image borders are not the same, artifacts will appear near the image borders. To avoid this, carry out the computations directly.

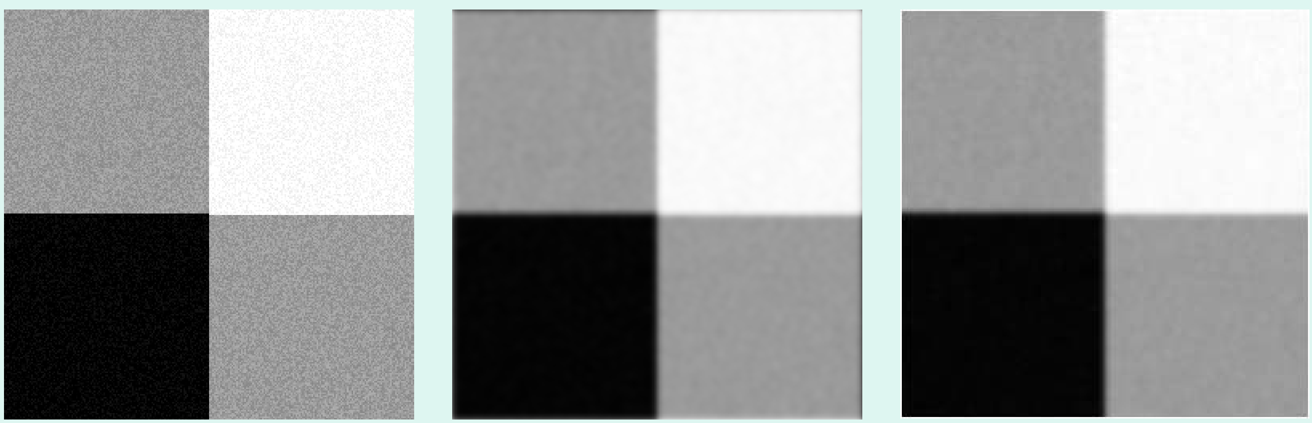


Image containing Zero-mean noise

Computed with FFT

Computed directly

Gaussian filter of $\sigma = 2$ pixels

Image segmentation

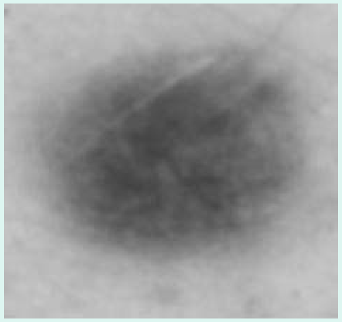
This is the process of partitioning an image into meaningful parts. There are two main approaches to image segmentation.

– Methods that use information within regions
– Methods that use information on the boundary between regions

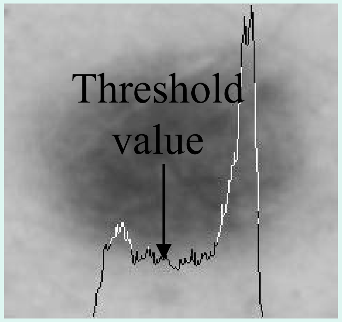
Intensity thresholding

- Assuming an image that contains objects with intensities that represent a Gaussian distribution and a background with intensities that represent a different Gaussian distribution, the objects can be separated from the background using the intensity at the valley between the two histogram modes.
- This method works well when an image contains homogeneous objects and a homogeneous background and the properties of the objects and the background are different.

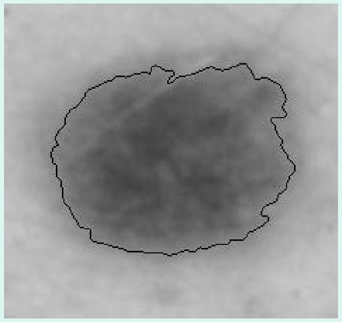
Original



Smoothed

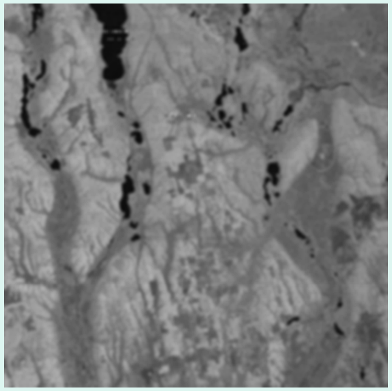


Histogram

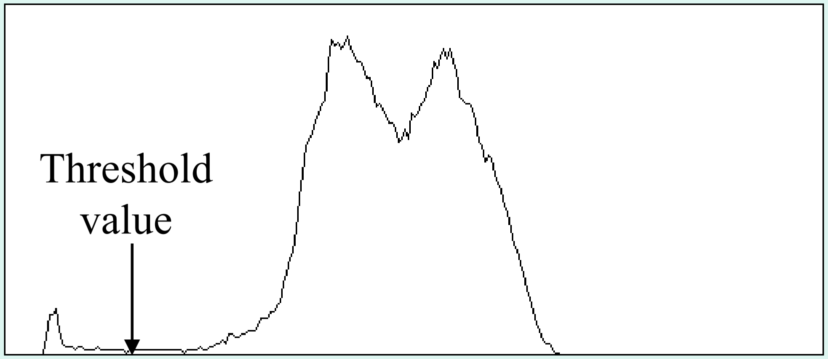


Thresholded image

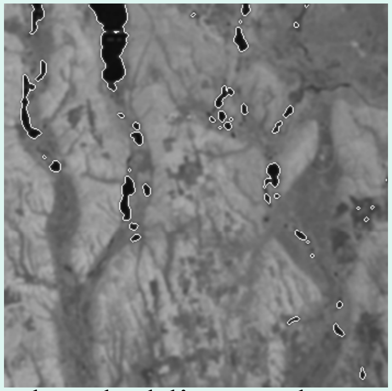
Intensity thresholding



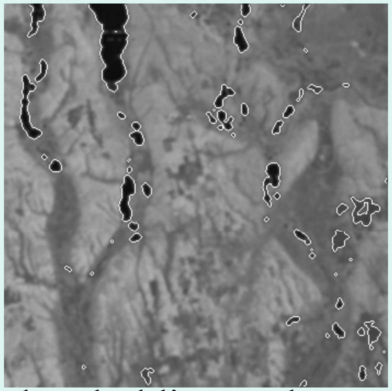
A Landsat image



Histogram of the image



Thresholding at the valley between the first two peaks.



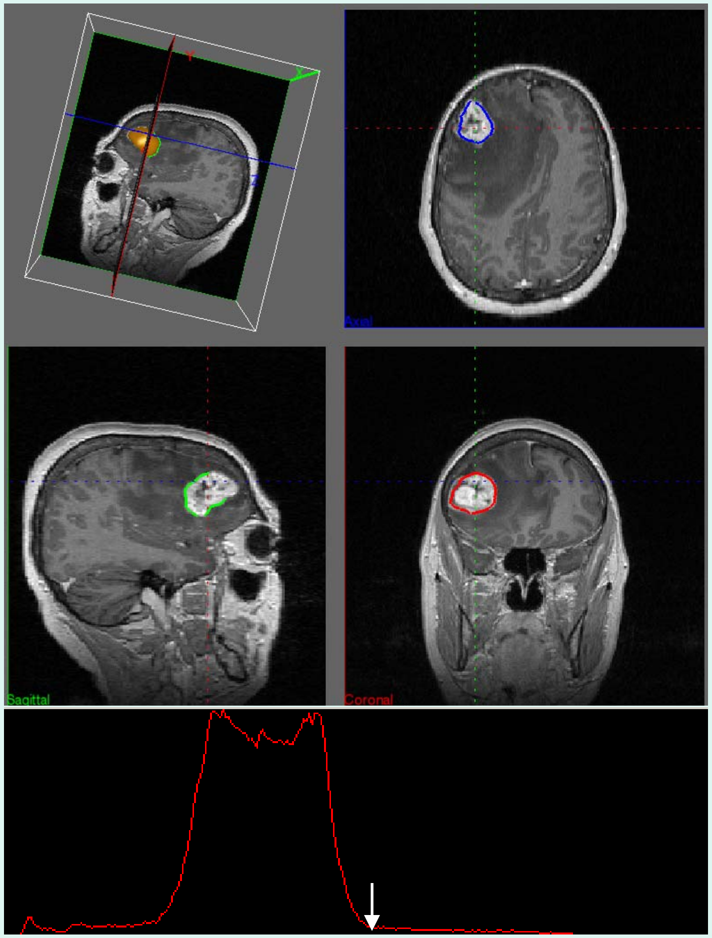
Thresholding at the average intensity of highest-gradient pixels.

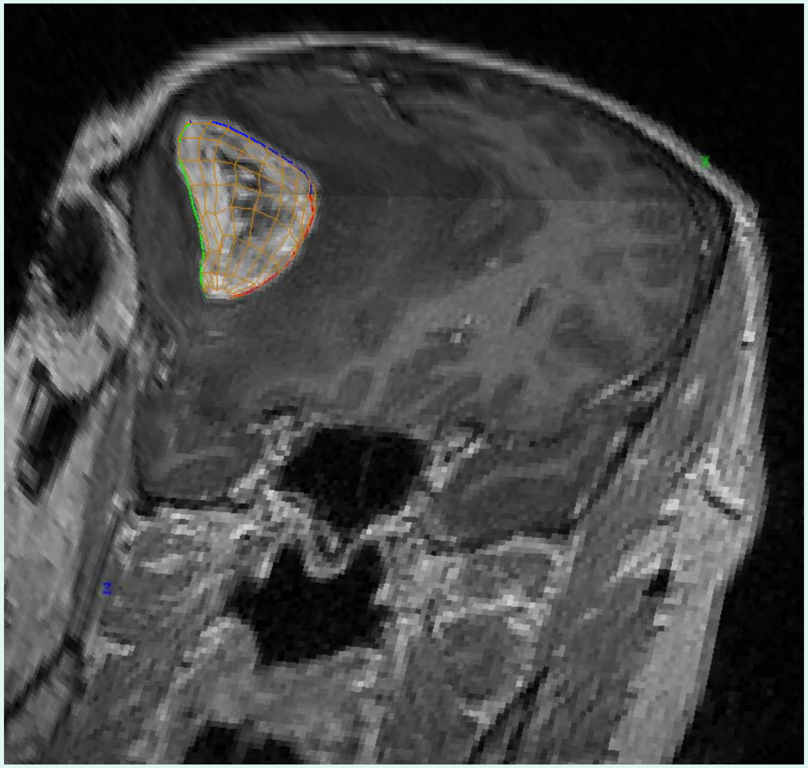
A. Goshtasby

Threshold selection

- The threshold value is computed by:
  - Finding the valley between the modes of the histogram of the image.
  - Finding the intensity that represents the average of intensities of high-gradient pixels.
  - Finding the intensity at which a change in the intensity will minimally change the segmentation result.

Interactive thresholding





Delineated tumor

Data courtesy of Kettering Medical Center

Software courtesy of Marcel Jackowski, Yale University

Edge detection

- Edge detection methods can be categorized into those that search for locally maximum image gradient magnitudes and those that search for zero-crossings of the Laplacian of an image.
- Methods that search for gradient peaks do not pick false edges but the ones picked could be disconnected.
- Methods that search of the zero-crossings of the Laplacian image find closed boundaries, but parts of the boundaries could be false.

LoG edge detector

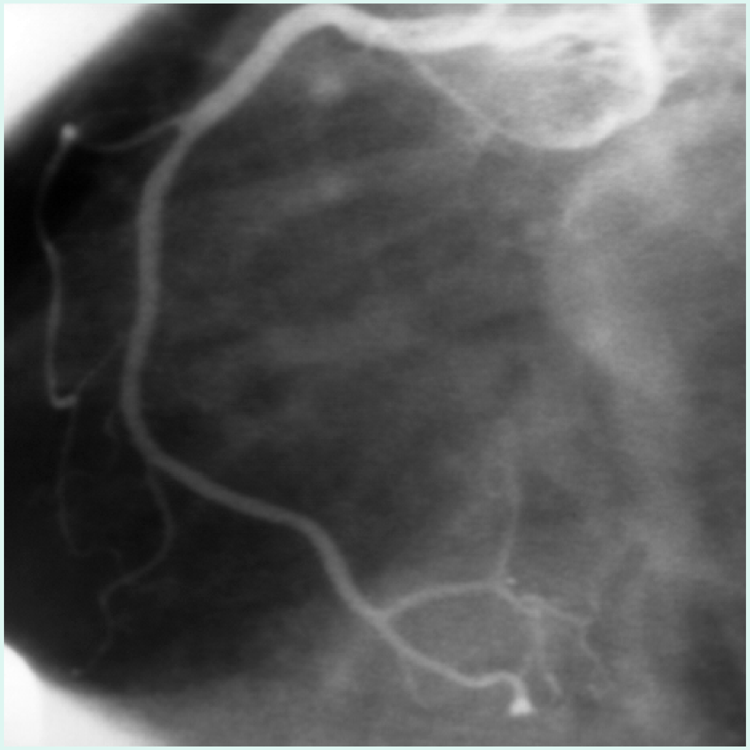
Determination of the LoG of an image involves computation of:

$$
\begin{aligned}
LoG[f(x,y)] &= \frac{\partial^2[f(x,y) \star G(x,y)]}{\partial x^2} + \frac{\partial^2[f(x,y) \star G(x,y)]}{\partial y^2} \\
&= f(x,y) \star \frac{\partial^2 G(x,y)}{\partial x^2} + f(x,y) \star \frac{\partial^2 G(x,y)}{\partial y^2}
\end{aligned}
$$

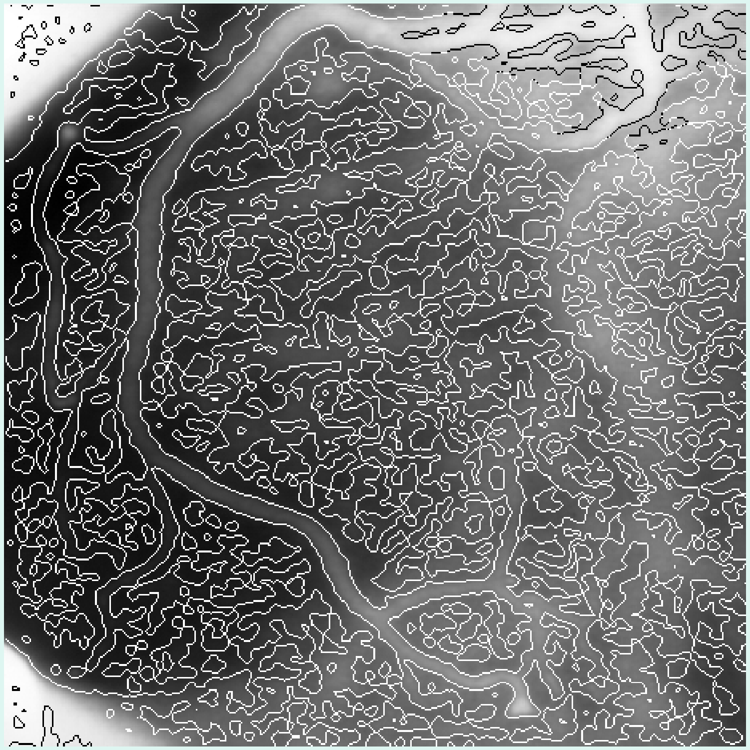
or,

$$
LoG[f(x,y]) = \frac{\partial^2 G(x)}{\partial x^2} \star G(y) \star f(x,y) + G(x) \star \frac{\partial^2 G(y)}{\partial y^2} \star f(x,y)
$$

LoG edge detection 2-D

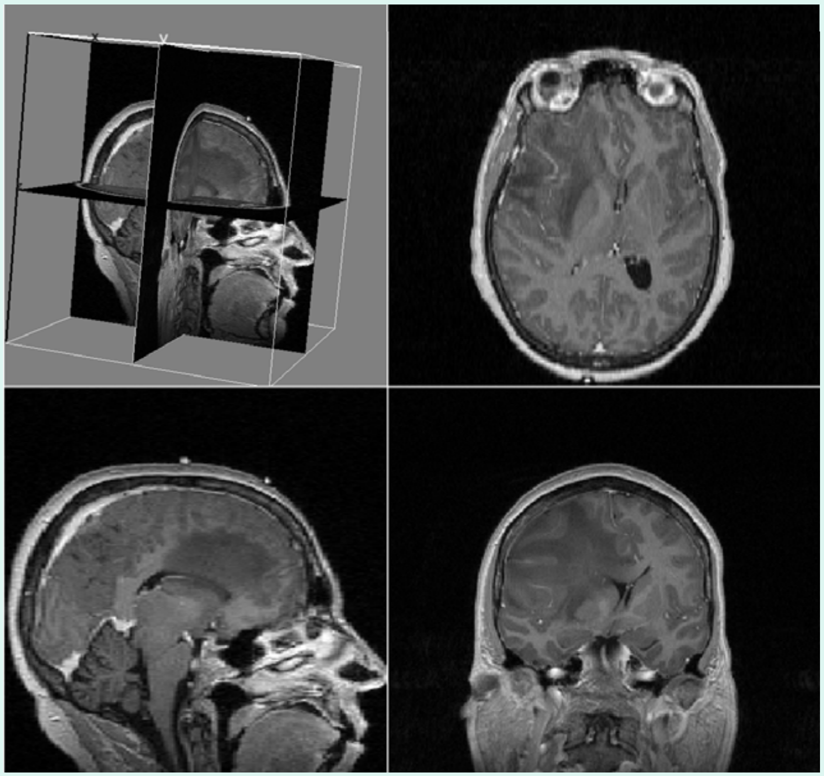


An *x*-ray angiogram

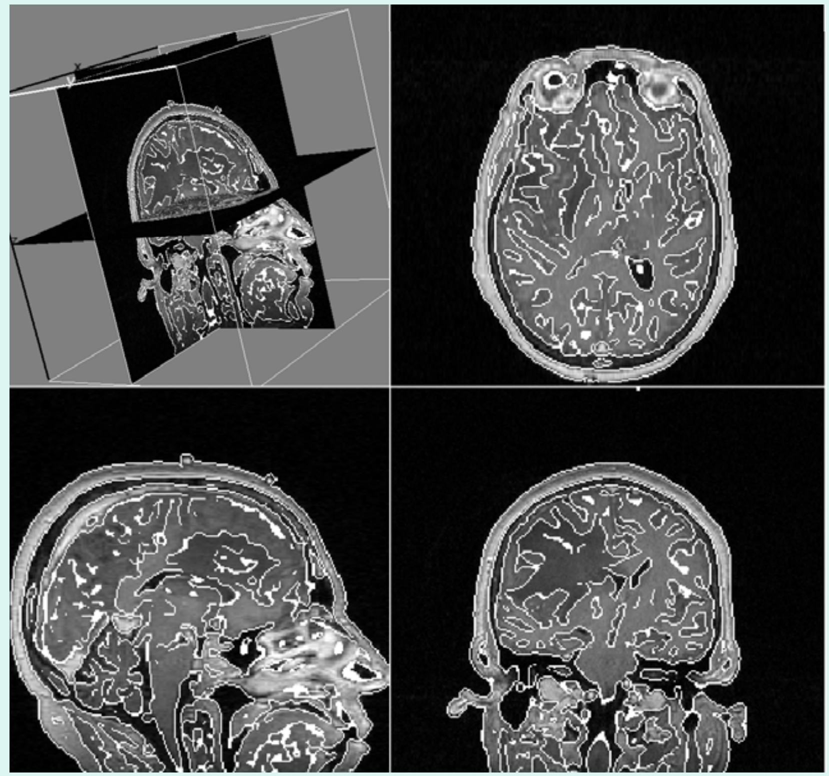


LoG zero-crossing edges

LoG edge detection: 3-D



A volumetric MR brain image

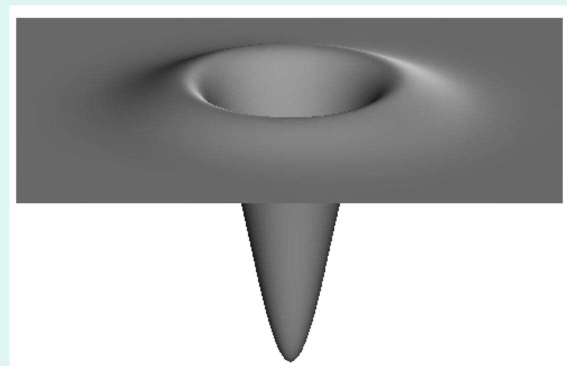


LoG edges

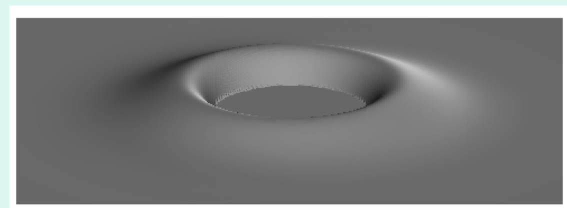
Data courtesy of Kettering Medical Center

Edge detection by intensity ratio

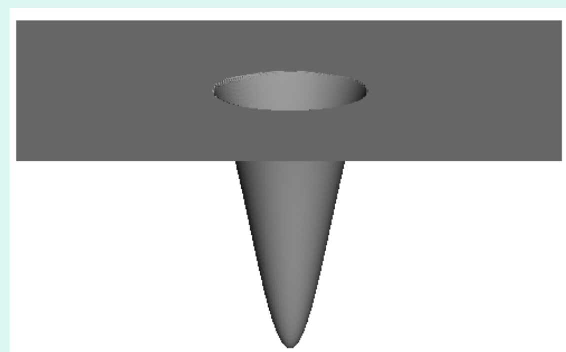
- Zero-crossing edges can be determined by convolving the negative and positive parts of the LoG with an image separately and subtracting the convolved images and locating the zero-crossings.
- If instead of subtracting corresponding values in the convolved images, we divide the values and locate the one-crossings, intensity-ratio edges will be obtained.



LoG operator in 2-D

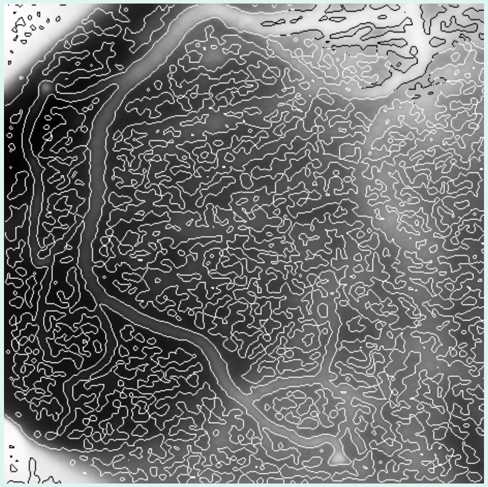


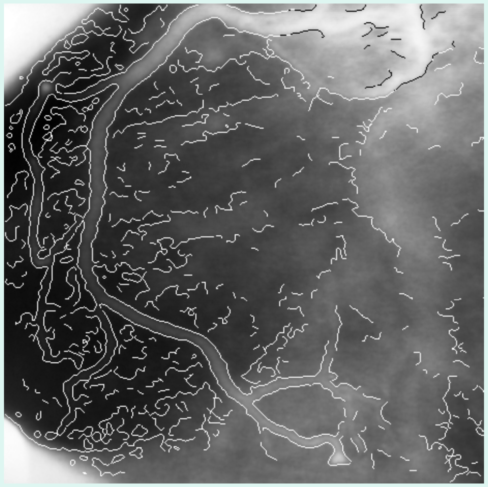
Positive part

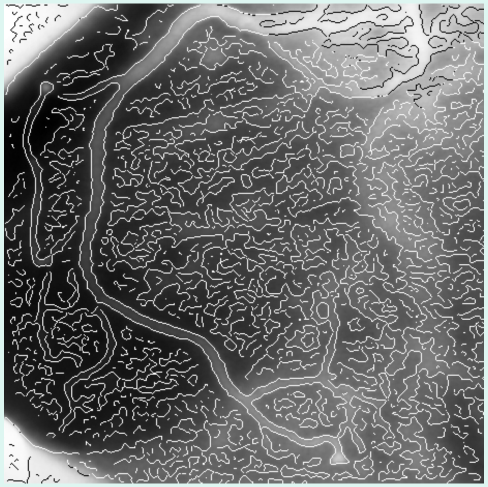


Negative part

Intensity ratio edges

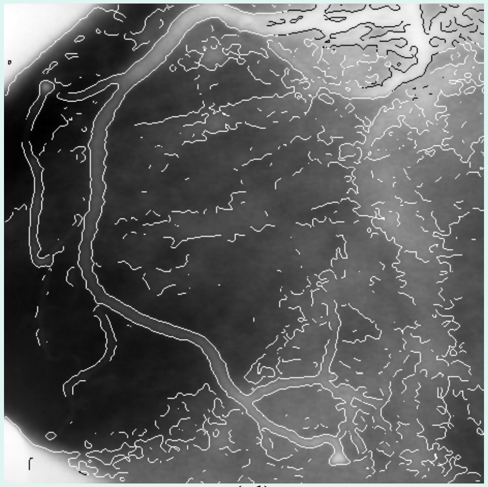

(a)


(b)


(c)

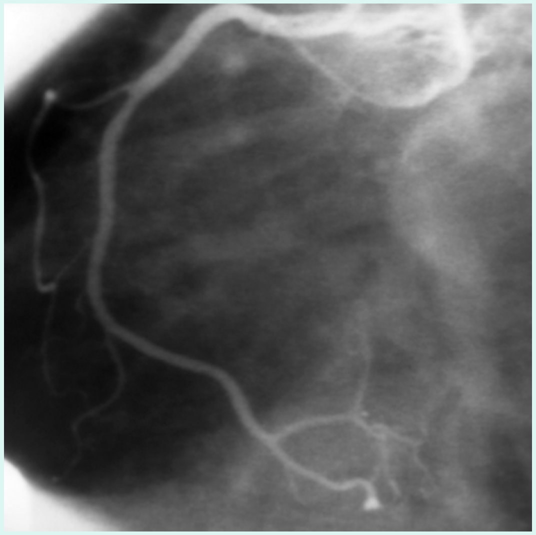
(a) Intensity ratio edges. (b) 30% highest-gradient ratio edges. (c) Intensity difference edges. (d) 30% highest-gradient difference edges.

Intensity ratios detect more edges in dark areas while intensity differences detect more edges in bright areas.

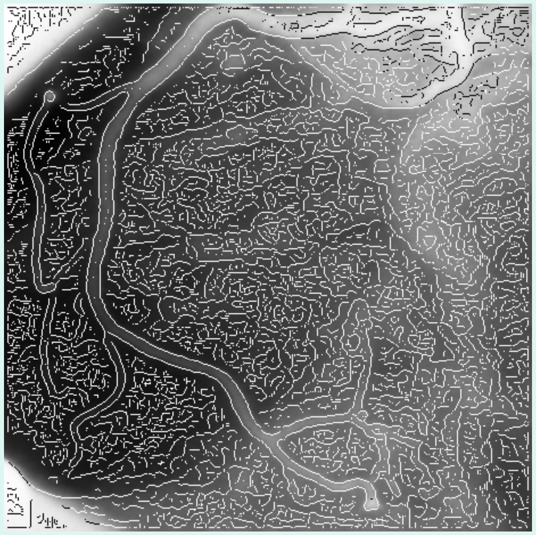

(d)

Canny edge detector

- This method smoothes the image with a Gaussian to reduce noise and then locates locally maximum gradient magnitudes in the gradient direction and uses them as the edges.



An *x*-ray angiogram



Canny edges with $\sigma = 2.5$ pixels

Edges in color images

- Edge detection in a color image can be considered edge detection in a 2-D vector field.
- If ***u****(x,y)* and ***v****(x,y)* represent color gradients in *x* and *y* directions, edges can be considered points where color gradients are locally maximum in the gradient direction.
- If *R(x,y), G(x,y),* and *B(x,y)* represent red, green, and blue color components at *(x,y)*, respectively, color gradients are:

$$\begin{aligned}\mathbf{u}(x,y) &= \frac{\partial R(x,y)}{\partial x}\mathbf{r} + \frac{\partial G(x,y)}{\partial x}\mathbf{g} + \frac{\partial B(x,y)}{\partial x}\mathbf{b} \\ \mathbf{v}(x,y) &= \frac{\partial R(x,y)}{\partial y}\mathbf{r} + \frac{\partial G(x,y)}{\partial y}\mathbf{g} + \frac{\partial B(x,y)}{\partial y}\mathbf{b}\end{aligned}$$

r, **g**, and **b** are unit vectors along red, green, and blue axes, respectively, in the color space.

Color edges

- Gradient direction at *(x,y)* is the direction maximizing

$$F(x,y) = [\mathbf{u}(x,y)\cos\theta(x,y) + \mathbf{v}(x,y)\sin\theta(x,y)]^2$$

and is obtained from

$$\theta(x,y) = 0.5\tan^{-1}\left(\frac{2\mathbf{u}(x,y)\cdot\mathbf{v}(x,y)}{\mathbf{u}(x,y)\cdot\mathbf{u}(x,y) - \mathbf{v}(x,y)\cdot\mathbf{v}(x,y)}\right)$$

Edge detection in color images

A color image

Edges of the color image

Preprocessing references

1. D. Marr and E. Hildreth, Theory of edge detection, *Proc. R. Soc. Lond.*, **207**:187–217 (1980).
2. J. J. Clark, Authenticating edges produced by zero-crossing algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **11**(1):43–57 (1989).
3. J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8**:679–714 (1986).
4. L. Ding and A. Goshtasby, On the Canny edge detector, *Pattern Recognition,* **34**:721–725 (2001).
5. M. Bomans, K.-H. Hohne, U. Tiede, and M. Riemer, 3-D segmentation of MR images of the head for 3-D display, *IEEE Transactions on Medical Imaging*, **9**(2):177–183 (1990).
6. A. Goshtasby and H-L Shu, Edge detection by curve fitting, *Image and Vision Computing*, **13**(3): 169–177, 1995.
7. A. Goshtasby, On edge focusing, *Image and Vision Computing*, **12**(4): 247–256, 1994.
8. L. Zagorchev, A. Goshtasby, and M. Satter, R-snakes, *Image and Vision Computing*, **25:** 945–959, 2007.