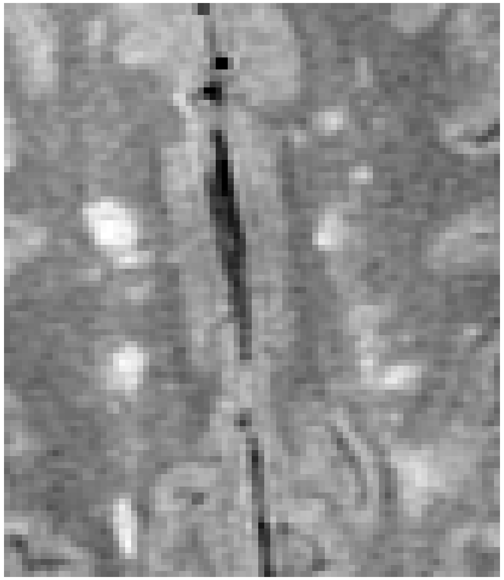
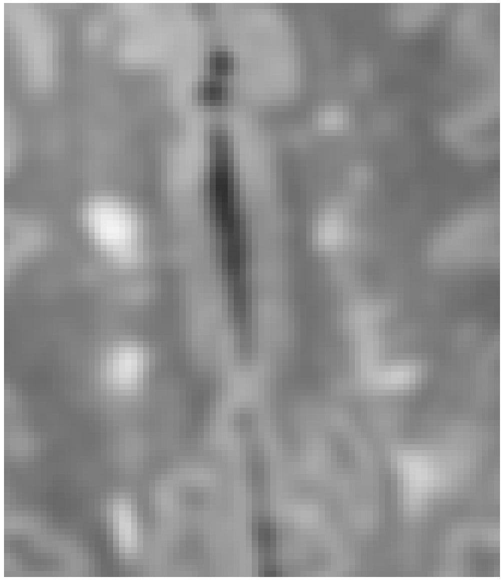
References

- Book: Chapters 5 & 6, Image Processing, Analysis, amd Machine Vision, Sonka et al, latest edition (you may collect a copy of the relevant chapters from my office)
- Papers:
  - Perona and Malik, *IEEE TPAMI*, **12**: 629-639, 1990.
  - Saha and Udupa, *IEEE TMI*, **20**:1140-1156, 2001.
  - Saha, *CVIU*, 99, 384-413, 2005.
  - Canny, *IEEE TPAMI*, **8**:679-698,1986

Filtering

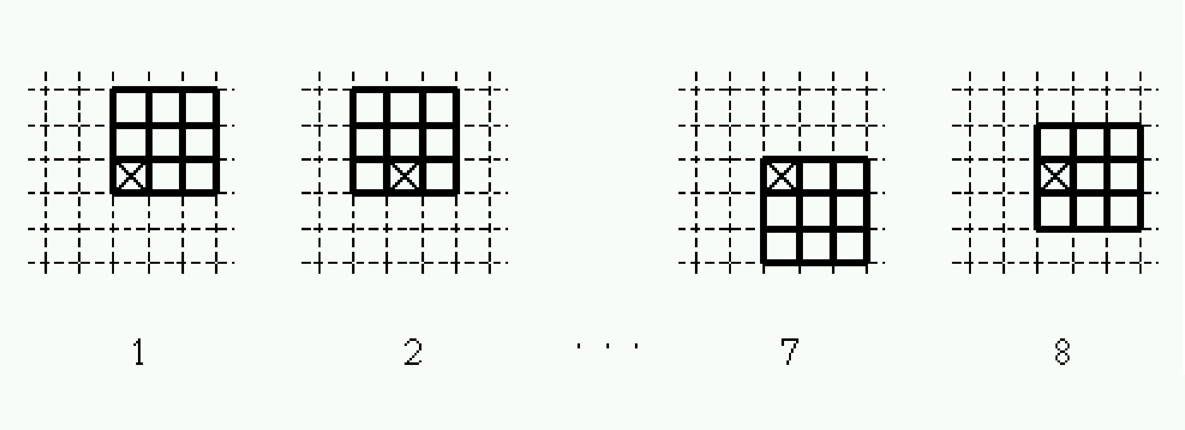
- Objective
  - improve SNR and CNR
- Challenges
  - blurs object boundaries and smears out important structures

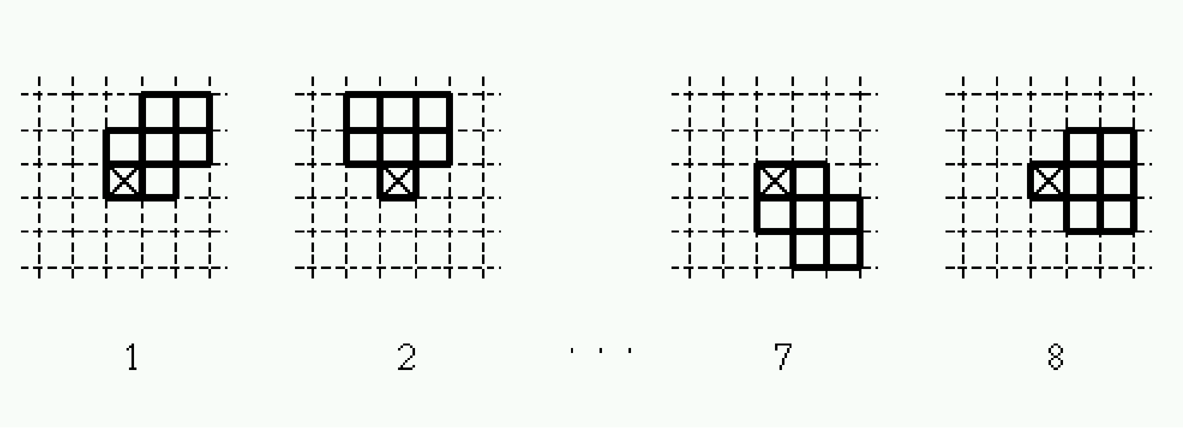




Averaging using rotating mask

- Consider each image pixel (i,j). Calculate dispersion in the mask for all possible mask rotations about (i,j).
- Choose the mask with minimum dispersion.
- Assign to the pixel (i,j) in the output image the average brightness in the chosen mask.

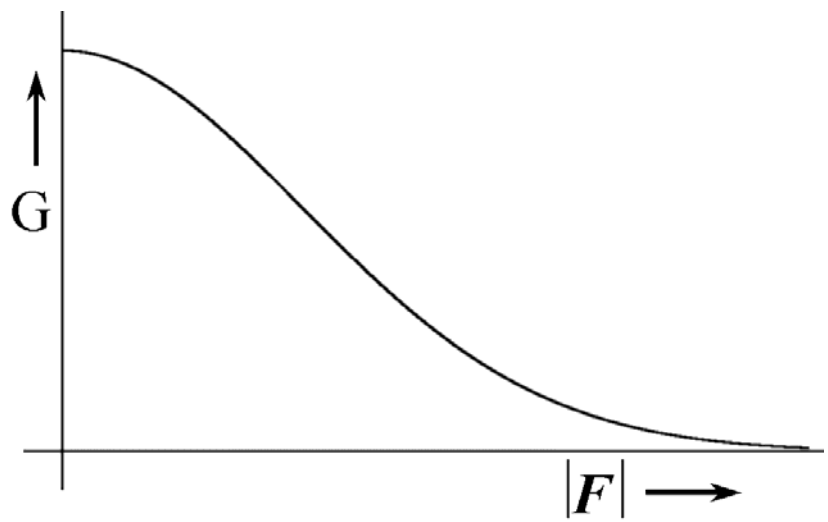


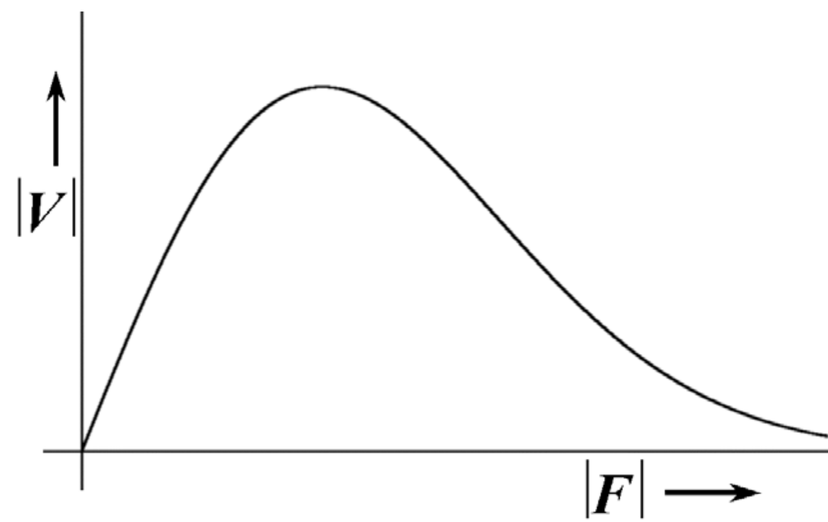


Anisotropic diffusive filtering

- An iterative process in which intensity diffusion V takes place between adjacent pixels in a nonlinear fashion with gradients F as follows:

V = *G*F,   where *G* is diffusion conductance





Diffusion

- A diffusion process can be defined using the divergence operator “div” on a vector field. A mathematical formulation of the diffusion process over a vector field **V** at a point *c* is as follows

$$\frac{\partial f}{\partial t} = \mathrm{div}\,\mathbf{V} = \lim_{\Delta\tau \to 0} \int_s \mathbf{V} \bullet d\mathbf{s}$$

Mathematical formulation

$$\text{Diffusion flow:} \qquad \mathbf{V} = G\mathbf{F}$$

$G:$ Nonlinear diffusion conductance

- $\mathbf{D}(c,d)$: the unit vector for c toward d
- Intensity gradient at l-th iteration:

$$\mathbf{F}_l(c,d) = \frac{f_l(c) - f_l(d)}{\sqrt{\sum_{i=0}^{n} (c_i - d_i)^2 v_i^2}} \mathbf{D}(c,d)$$

- v is the resolution vector, i.e., voxel length along each coordinate direction

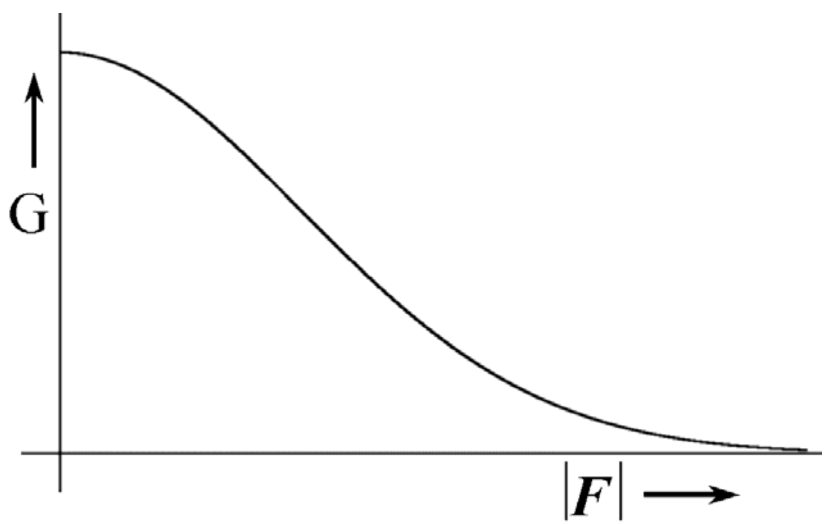
Continued…

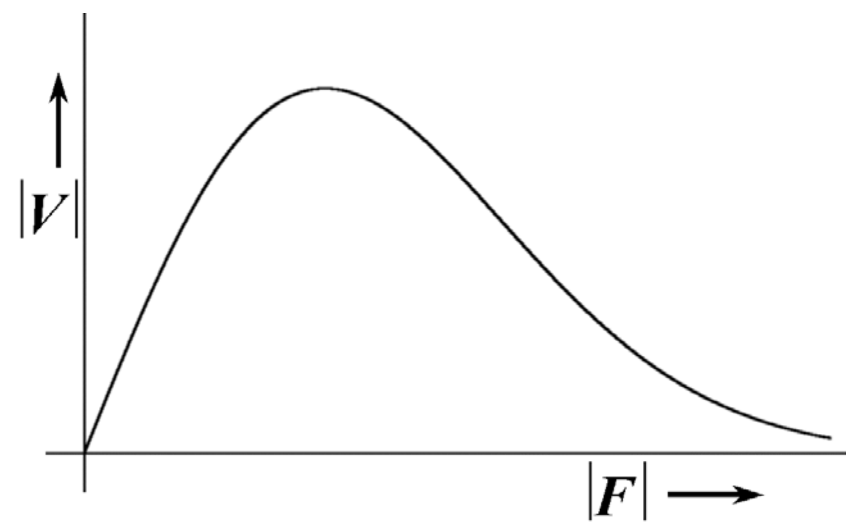
- Diffusion conductance function:

$$G_l(c,d) = e^{-\frac{|\mathbf{F}_l(c,d)|^2}{2\sigma^2}}$$

- Diffusion flow

$$\mathbf{V}_l(c,d) = \mathbf{F}_l(c,d)\, G_l(c,d)$$





Continued ...

- Iterative diffusion process

$$f_{l-1}(c) = \begin{cases} f(c), & \text{if } \mathrm{l} = 0, \\ f_{l-1}(c) - K_d \sum\limits_{d \in N(c)} \mathbf{V}_{l-1}(c,d), & \text{otherwise.} \end{cases}$$

- Diffusion constant

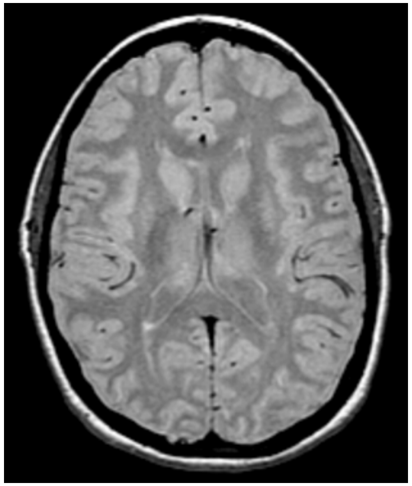
$$K_d \leq \min_{c \in C} \left[\frac{1}{\| N(c) \|} \right]$$

- Diffusion constant used here

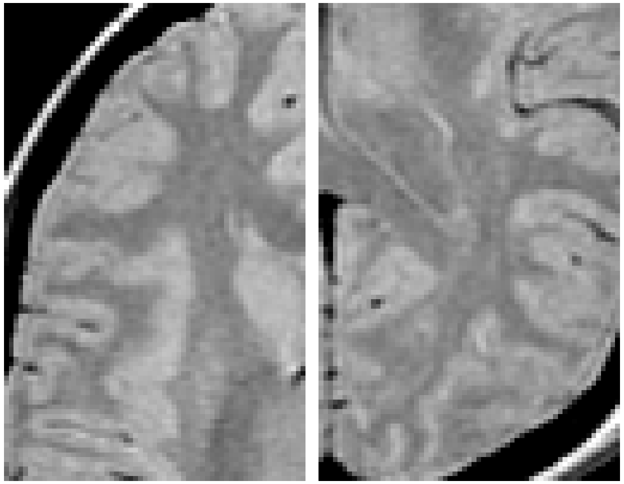
$$K_d = \begin{cases} \frac{1}{5}, & \text{in 2D,} \\ \frac{1}{7}, & \text{in 3D.} \end{cases}$$

Examples

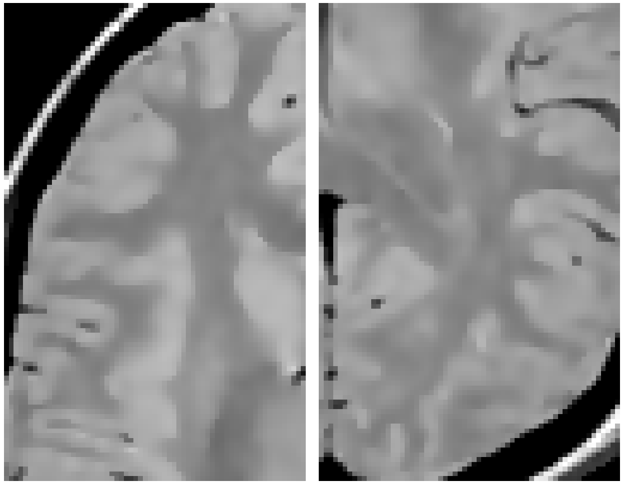
MS 2D



Original MR image



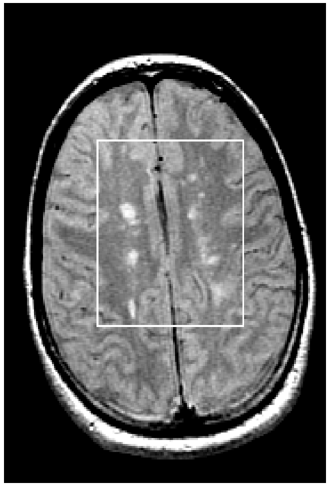
VOI from original image



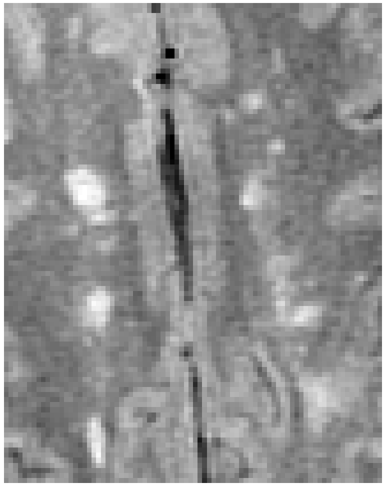
Anisotropic diffusion

Examples

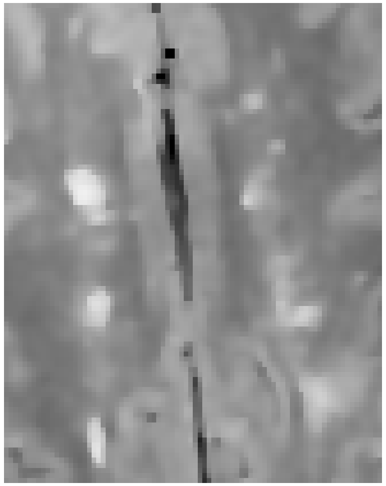
MS 3D



Original MR image

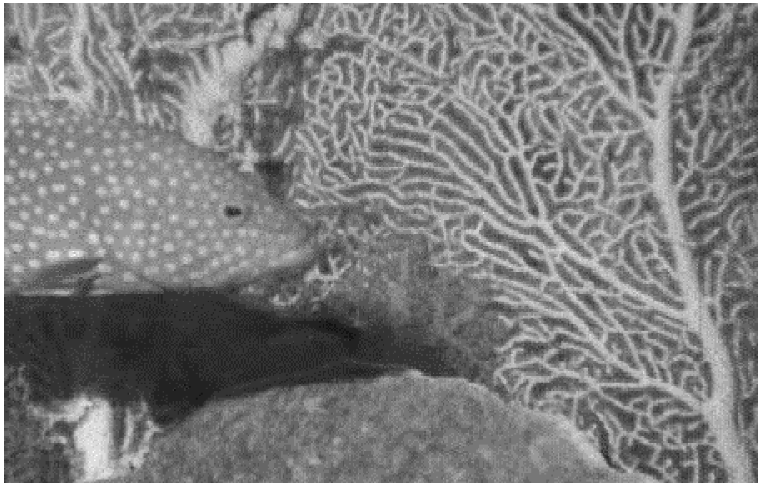


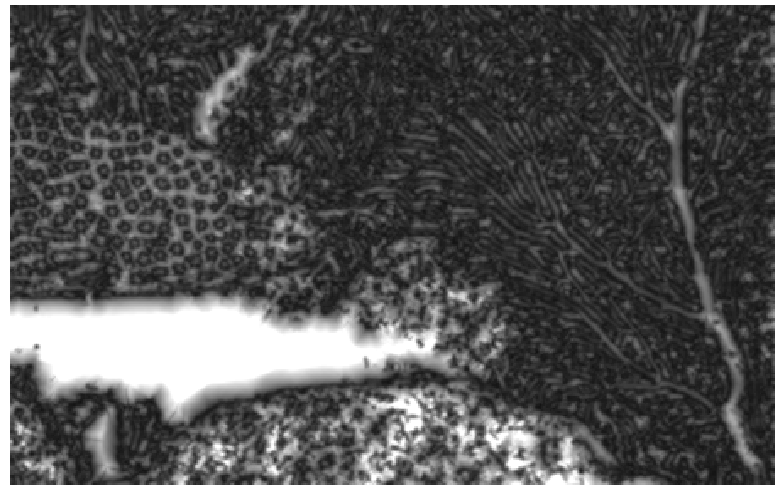
VOI from the original image

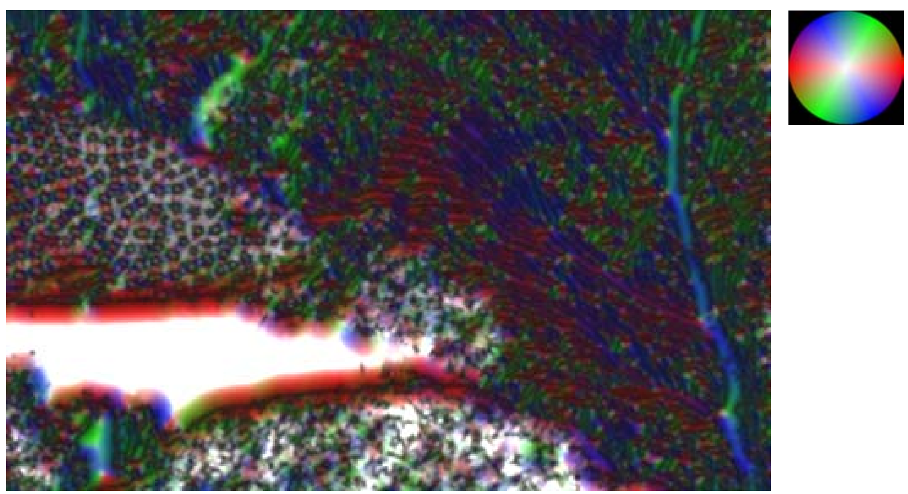


Anisotropic diffusion

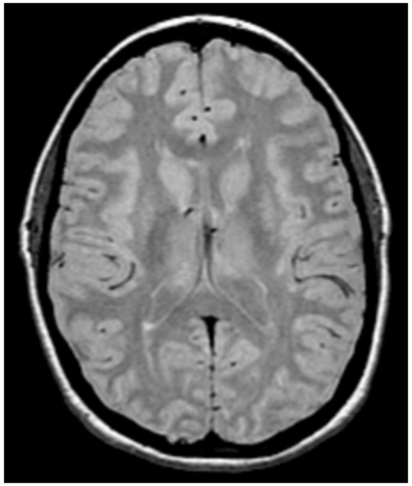
Use of structure scale in diffusive filtering



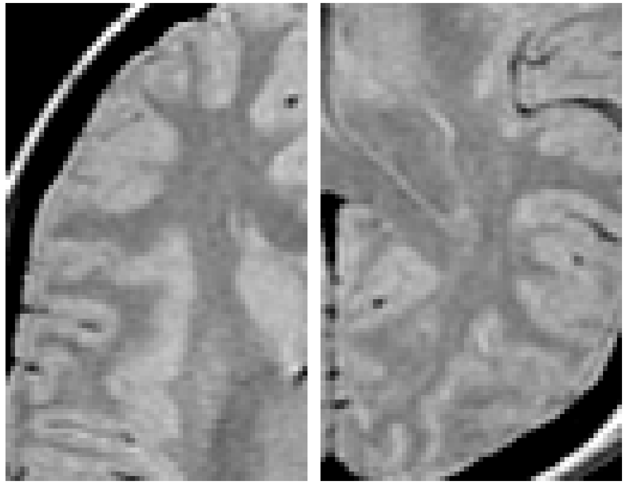




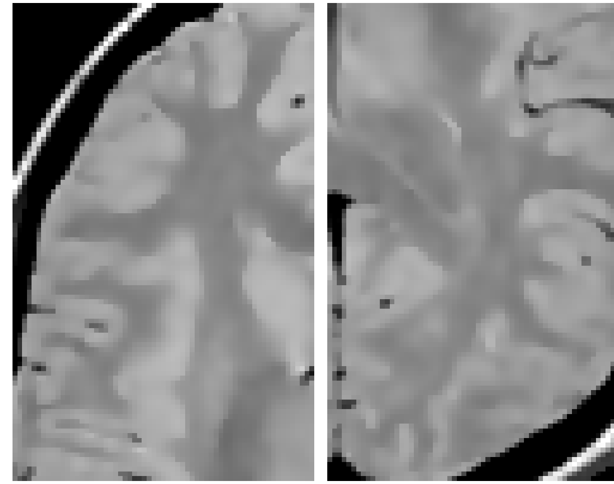
Examples



Original MR image



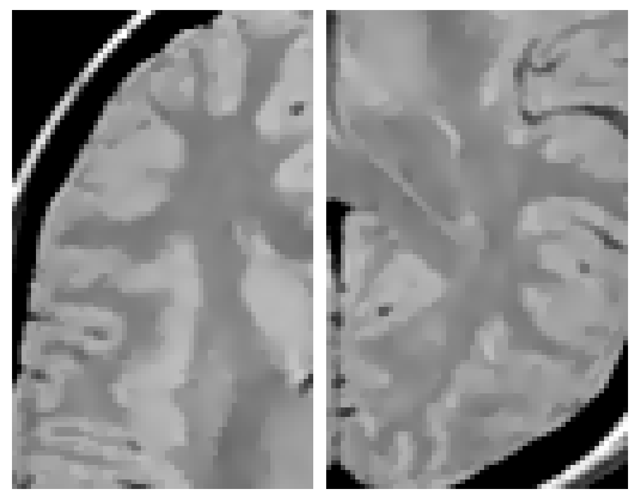
VOI from original image



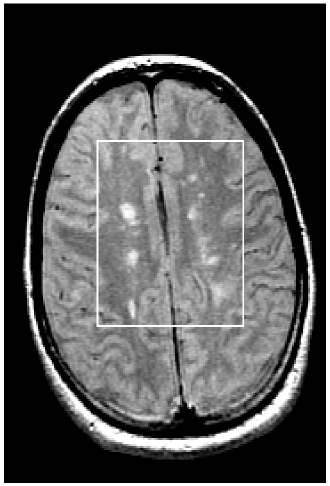
Anisotropic diffusion

MS 2D

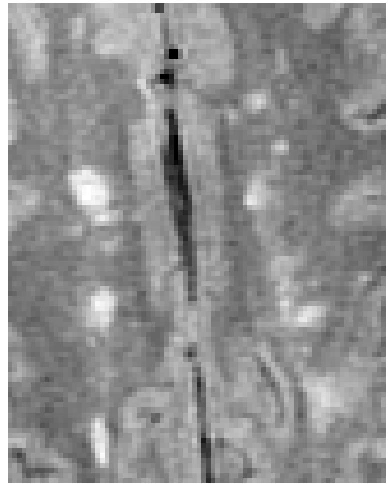
Scale-based anisotropic diffusion



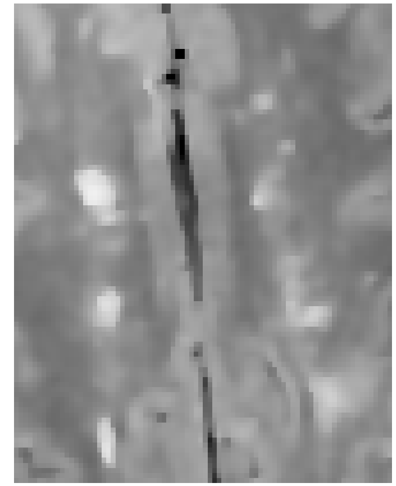
Examples



Original MR image



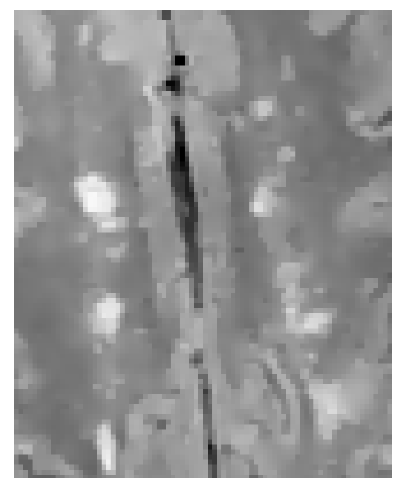
VOI from the original image



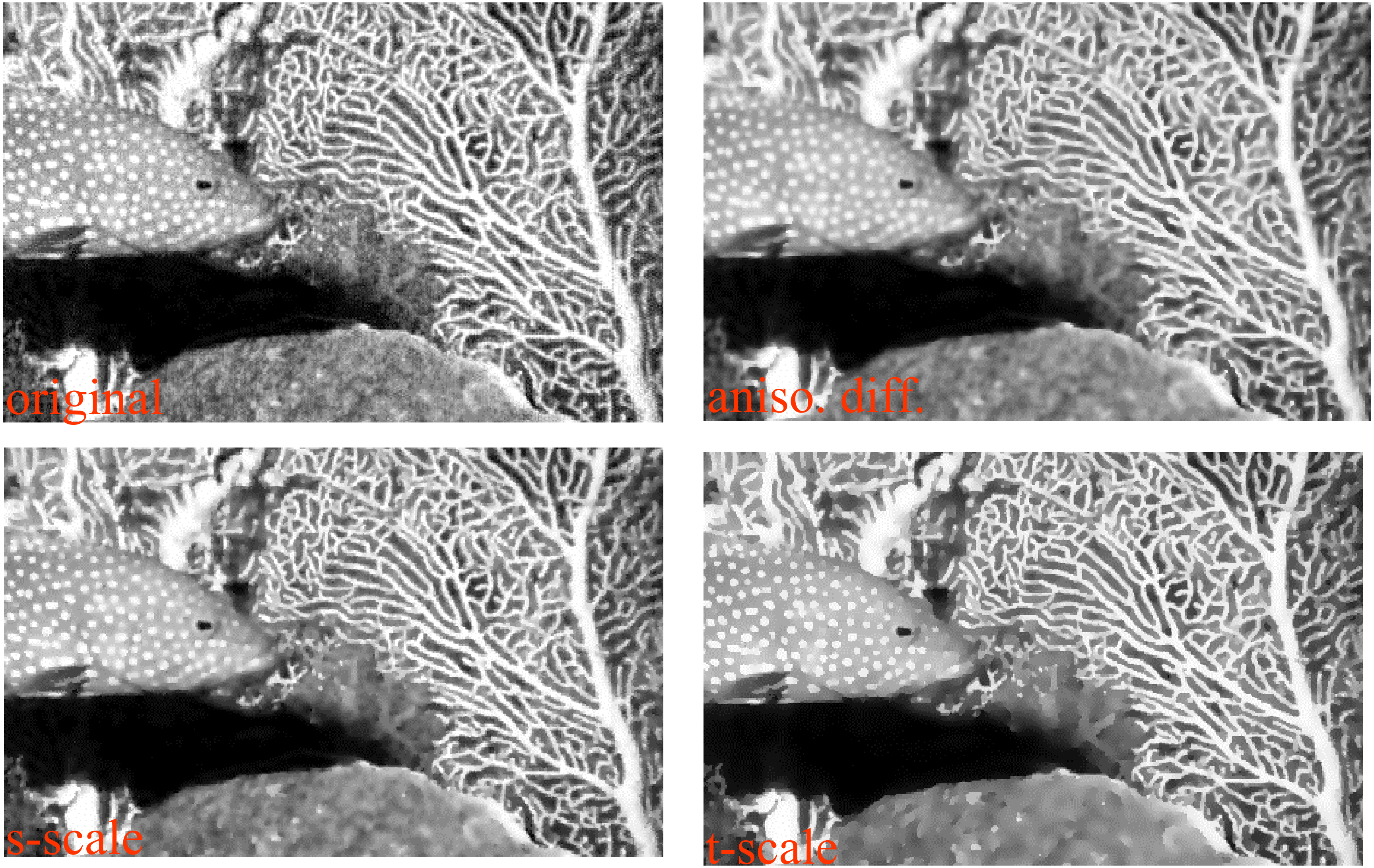
Anisotropic diffusion

MS 3D

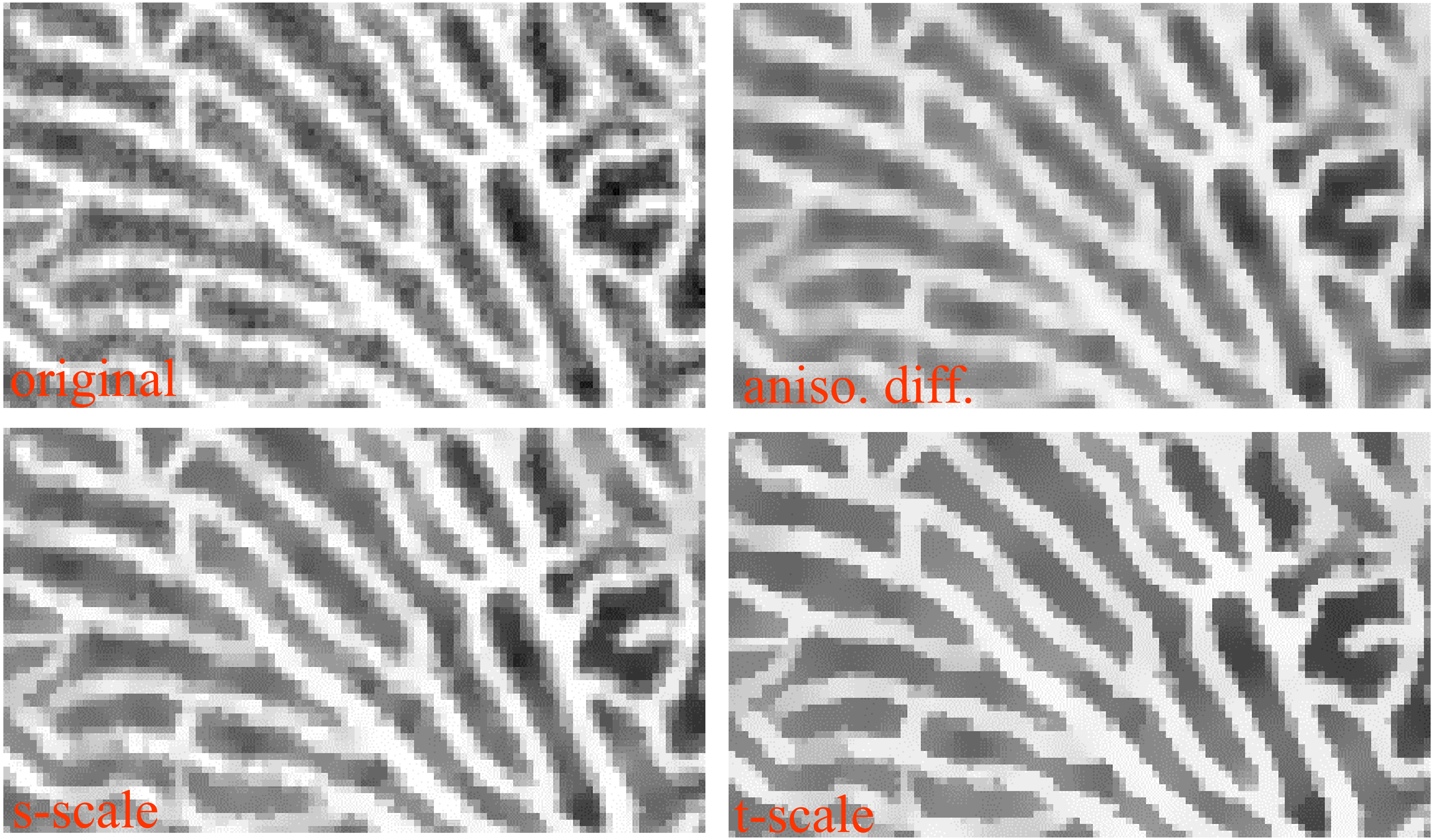
Scale-based anisotropic diffusion



Results



Zoomed Display

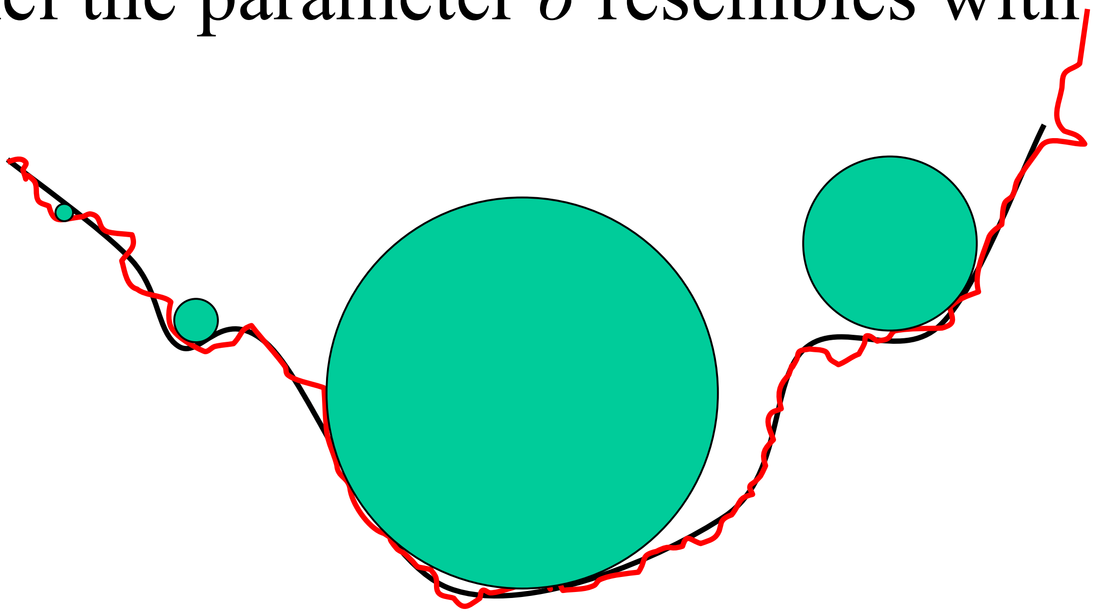


Canny’s edge detection

- The **detection** criterion expresses the fact that important edges should not be missed and that there should be no spurious response
- The **localization** criterion minimizes the distance between the actual and the located edge position
- The **response** criterion minimizes multiple response to a single edge

Scale in edge detection

- Scale is a resolution or a range of resolution needed to provide a sufficient yet compact representation of the object or a target information
- In a Gaussian smoothing or edge detection kernel the parameter σ resembles with **scale**



Canny's edge localization

- It seeks out zero-crossings of

$$\partial^2 (G * f)/\partial \mathbf{n}^2 = \partial G'_{\mathbf{n}}/\partial \mathbf{n} = \partial((\mathbf{G}' * f)\cdot \mathbf{n})/\partial \mathbf{n}$$

- In one-dimension a closed form solution may be found using calculus of variation.
- In two or higher dimension, the best solution is obtained by a numerical optimization, called **non-maximal suppression**, that essentially seeks for the best solution for

$$\partial^2 (G * f)/\partial \mathbf{n}^2 = 0$$

Algorithm: Non-maximal suppression

1. Quantize edge directions in the eight ways according to 8-connectivity
2. For each pixel with non zero edge magnitude, inspect the two adjacent pixels indicated by the direction of its edge
3. If the magnitude of either of these two exceeds that of the pixel under inspection, mark it for deletion
4. When all pixels have been inspected, re-scan the image and erase to zero all edge data marked for deletion

Examples

Edge strengths using DoG

Edges located using the non-maximal suppression algorithm

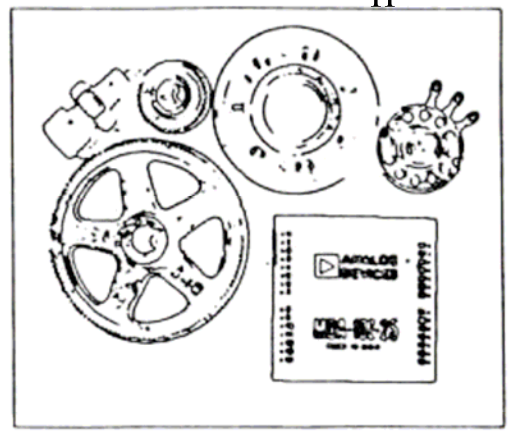
Algorithm: Hysteresis to filter output of an edge operator

1. Mark all edges with magnitude greater than t_H as correct
2. Scan all pixels with edge magnitude in the range $[t_L, t_H]$
3. If such a pixel is adjacent to another already marked as an edge, then mark it too.
4. Repeat from step 2 until stability

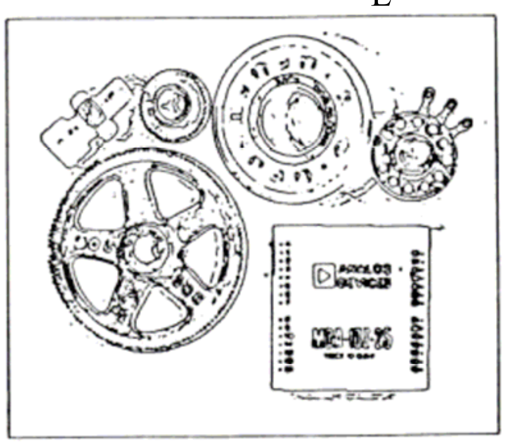
Examples

original

threshold at t_H

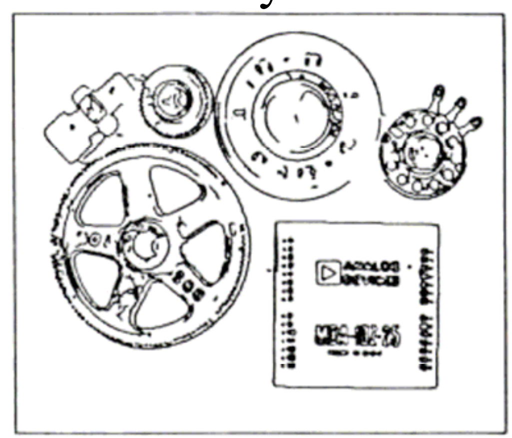


threshold at t_L



(b)

results of hysteresis

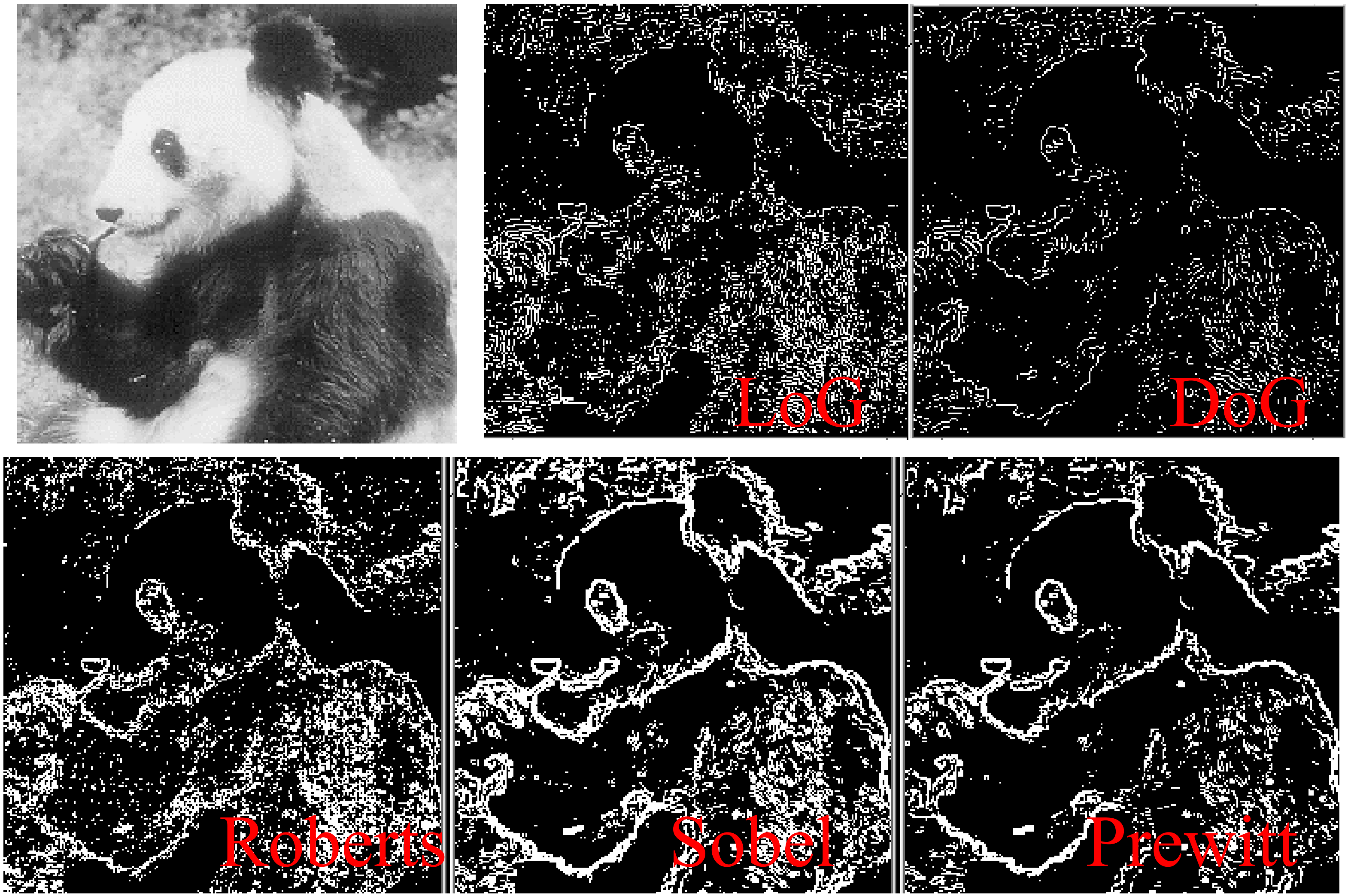


(d)

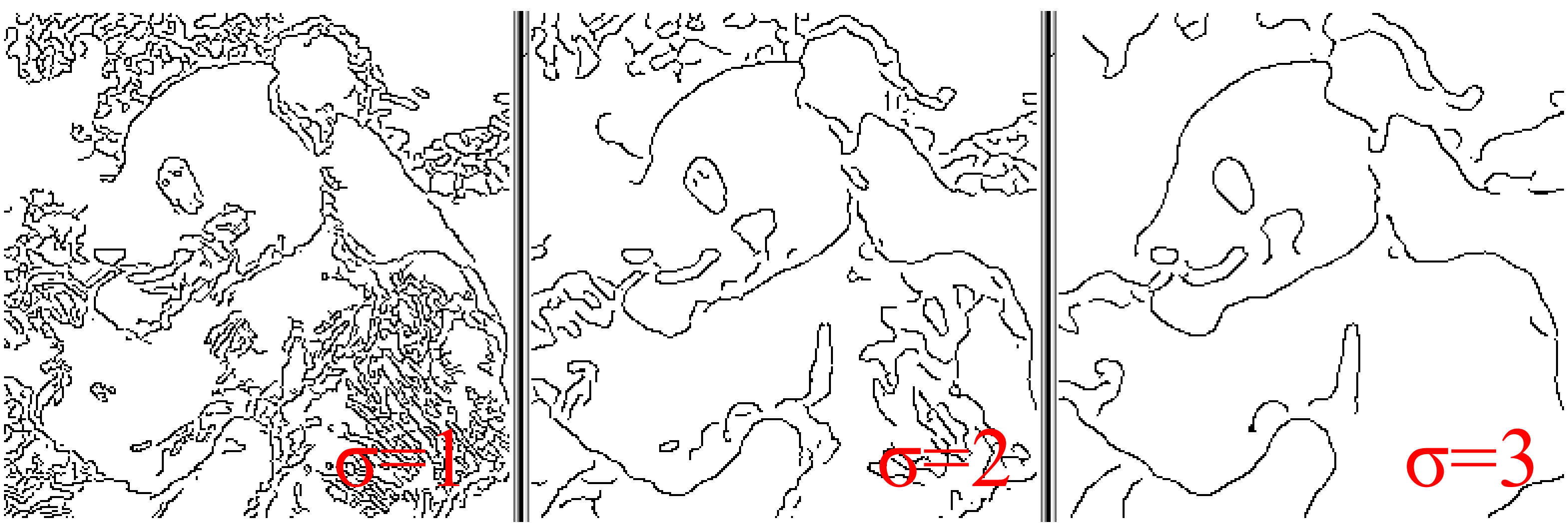
Canny's edge detector

1. Convolve an image *f* with a Gaussian of scale σ
2. Localize edge points using the **Non-Maximal Suppression** algorithm
3. Compute edge magnitude of the edge at each locations
4. Apply the **Hysteresis** algorithm to filter edge locations eliminating spurious responses
5. Repeat steps (1) through (5) for ascending values of scales σ of a range $[\sigma_{min}, \sigma_{max}]$
6. Aggregate the edge information at different scale using **feature synthesis**

Examples



Examples



Examples

Parametric edge detection

- **Facet model**: a piecewise continuous function representing the intensity in the neighborhood of a pixel, e.g., a bi-cubic faced model

$$
\begin{aligned}
f(i,j) = {} & c_1 + c_2 x + c_3 y + c_4 x^2 + c_5 xy \\
& + c_6 y^2 + c_7 x^3 + c_8 x^2 y + c_9 xy^2 + c_{10} y^3
\end{aligned}
$$

Parametric edge detection

- Model parameters may be computed using a least-squares method with singular value decomposition
- Facet model is computationally expensive but gives more accurate localization of edges with sub-pixel accuracy
- Haralick and Shairo have shown that, for a 5x5 facet model, the parameters may be directly computed using ten 5x5 kernels