figure 4.1. *Smoothing an image with a series of Gaussian filters. Note how the information is gradually being lost when the variance of the Gaussian filter increases from left to right, top to bottom.*

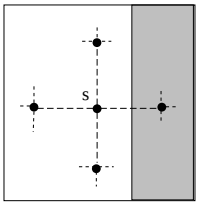
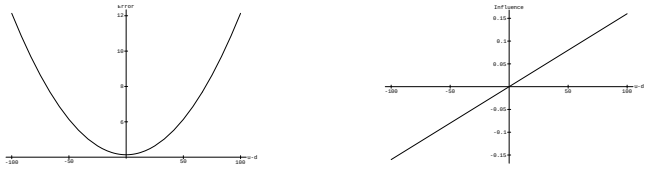
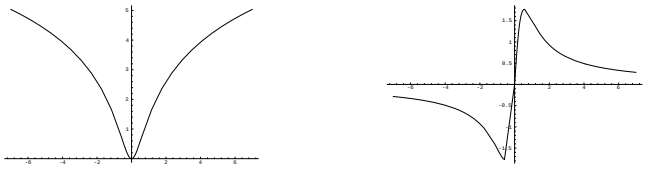


figure 4.2. *Local neighborhood of pixels at a boundary (intensity discontinuity).*



$\rho(x,\sigma)$ $\psi(x,\sigma)$

figure 4.3. *Least-squares (quadratic) error norm.*



$\rho(x,\sigma)$ $\psi(x,\sigma)$

figure 4.4. *Lorentzian error norm.*

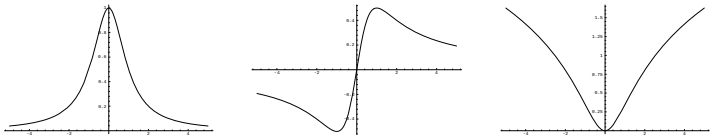


figure 4.5. *Lorentzian error norm and the Perona-Malik g stopping function.*

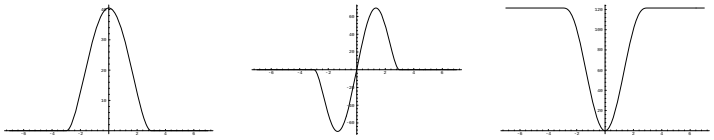


figure 4.6. *Tukey's biweight.*

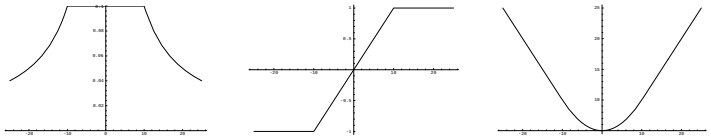


figure 4.7. *Huber's minmax estimator (modification of the L_1 norm).*

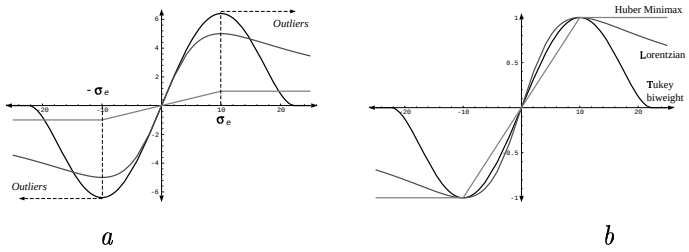


figure 4.8. *Lorentzian, Tukey, and Huber ψ-functions.* (a) *values of σ chosen as a function of σ_e so that outlier "rejection" begins at the same value for each function;* (b) *the functions aligned and scaled.*

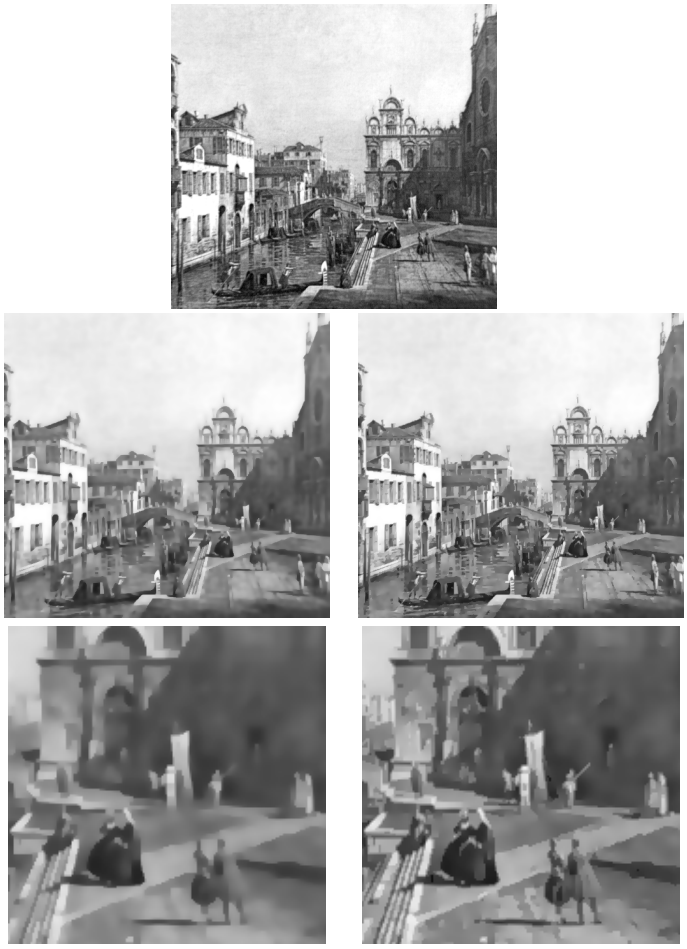


figure 4.9. *Comparison of the Perona-Malik (Lorentzian) function (left) and the Tukey function (right) after 100 iterations. Top: original image. Middle: diffused images. Bottom: magnified regions of diffused images.*

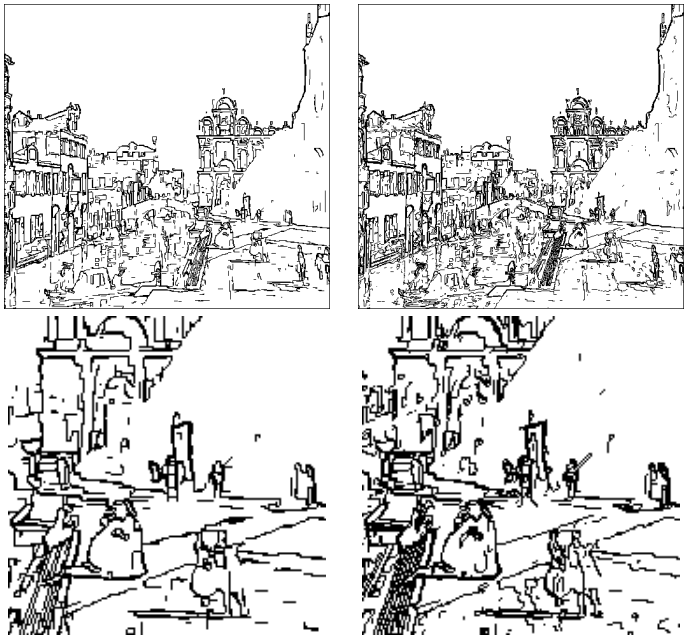


figure 4.10. *Comparison of edges (outliers) for the Perona-Malik (Lorentzian) function (left) and the Tukey function (right) after 100 iterations. Bottom row shows a magnified region.*

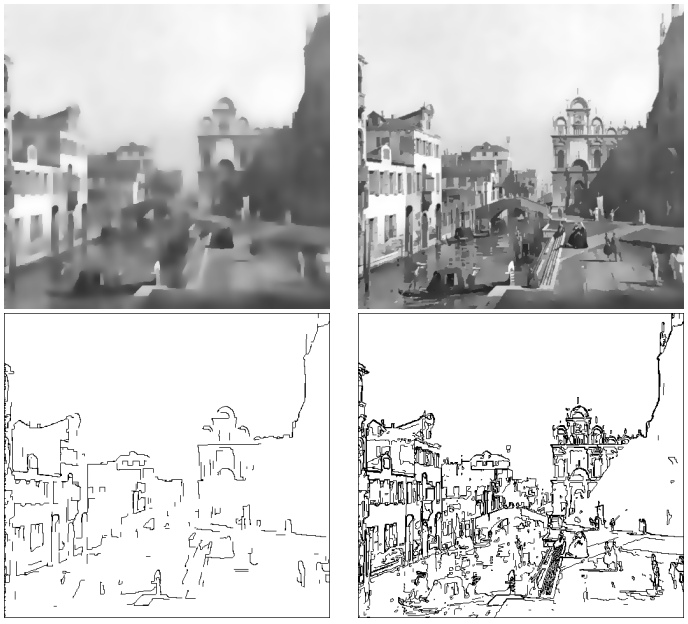


figure 4.11. *Comparison of the Perona-Malik (Lorentzian) function (left) and the Tukey function (right) after 500 iterations.*

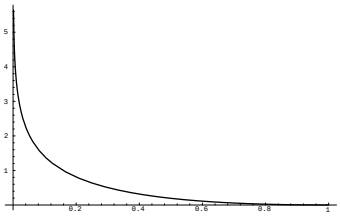


figure 4.12. *Lorentzian (Perona-Malik) penalty function,* $P(l)$, $0 \leq l \leq 1$.

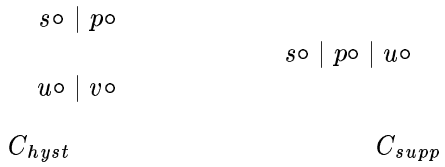


figure 4.13. *Cliques for spatial interaction constraints (up to rotation) at a site s. The circles indicate pixel locations and the bars indicate discontinuities between pixels. The C_{hyst} cliques are used for hysteresis and the C_{supp} cliques are used for non-maxima suppression.*

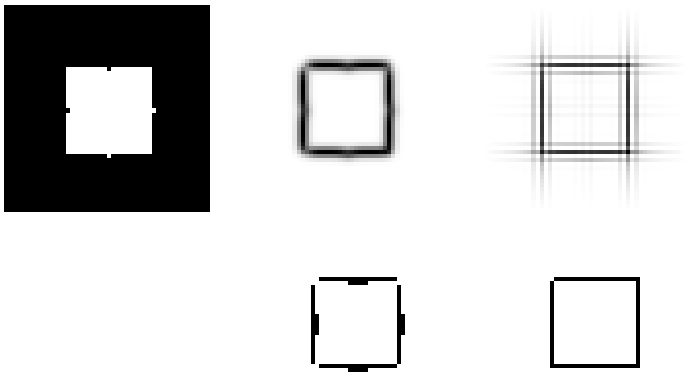


figure 4.14. *Anisotropic diffusion with spatial organization of outliers. Left: input image. Middle: line process for Perona-Malik (bottom - thresholded). Right: Perona-Malik line process with spatial coherence (bottom - thresholded).*

figure 4.15. *Anisotropic diffusion with spatially coherent outliers. Left: smoothed image. Right: the value of the line process at each point taken to be the product, $l_{s,h}l_{s,v}$, of the horizontal and vertical line process at s.*

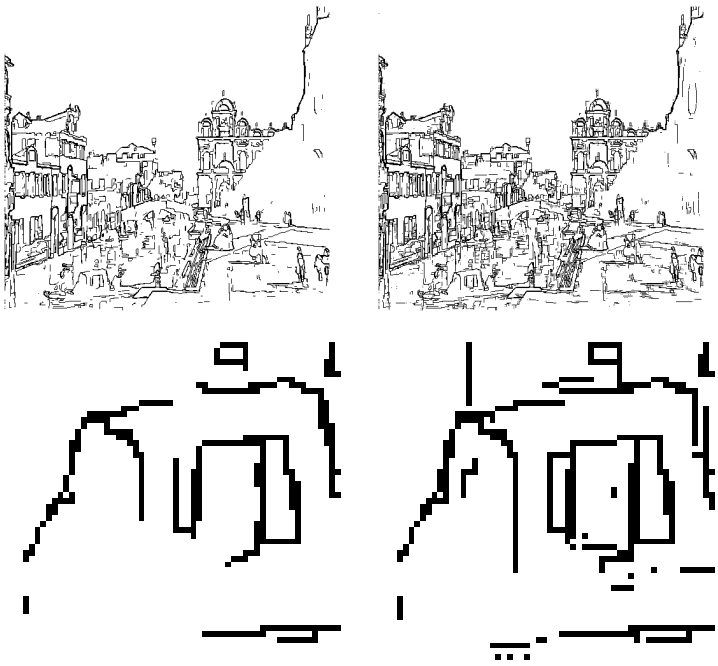


figure 4.16. *Edges obtained with Perona-Malik (left); and Perona-Malik with additional spatial coherence in the line processes (right). Lower images show details on the gondola.*

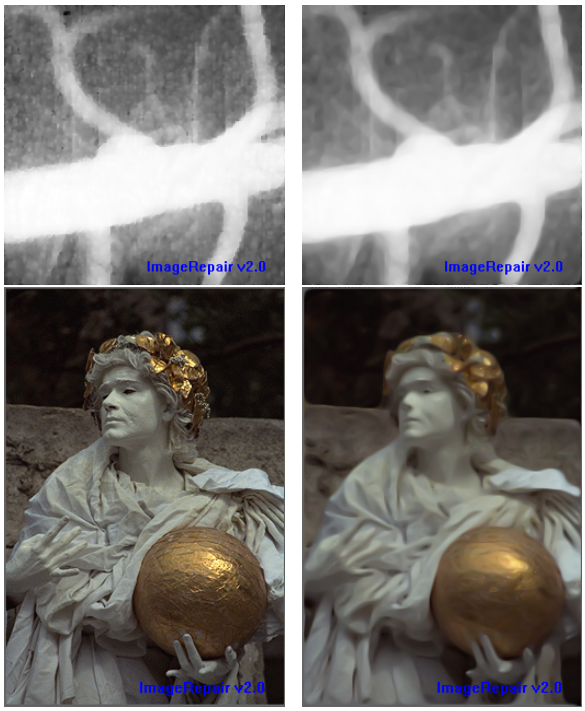


figure 4.17. *Example of anisotropic diffusion (original on the left and processed on the right).*

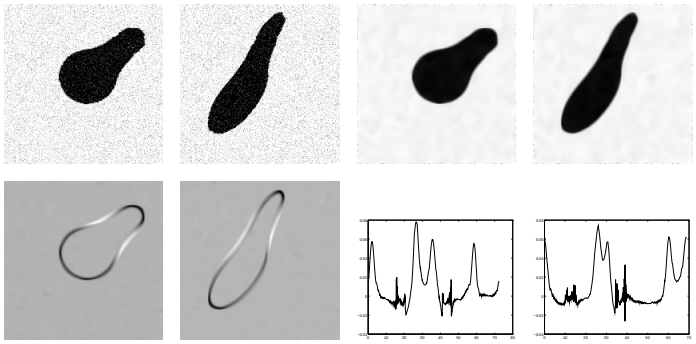


figure 4.18. *Affine denoising, edge maps, and curvature computation for two affine related images. The first row shows the two original noisy images (two left figures). Gaussian random noise was added independently to each one of the images after the affine transformation was performed. The subsequent images on the same row show the results of the affine denoising algorithm. The second row shows, on the left, affine invariant edges for the images after affine denoising. The last two figures on the second row show plots of the affine curvature vs. affine arc-length of the mid-range level-set for these images. The affine curvature was computed using implicit functions. Although this is not the best possible way to compute the affine curvature in discrete curves, it is sufficient to show the qualitative behavior of the algorithm. The affine arc-length was computed using the relation between affine and Euclidean arc-lengths described in the text. The curve was smoothed with the affine geometric heat flow for a small number of steps to avoid large noise in the discrete computations. Note that different starting points were used for both images, and therefore the corresponding plots are shifted.*

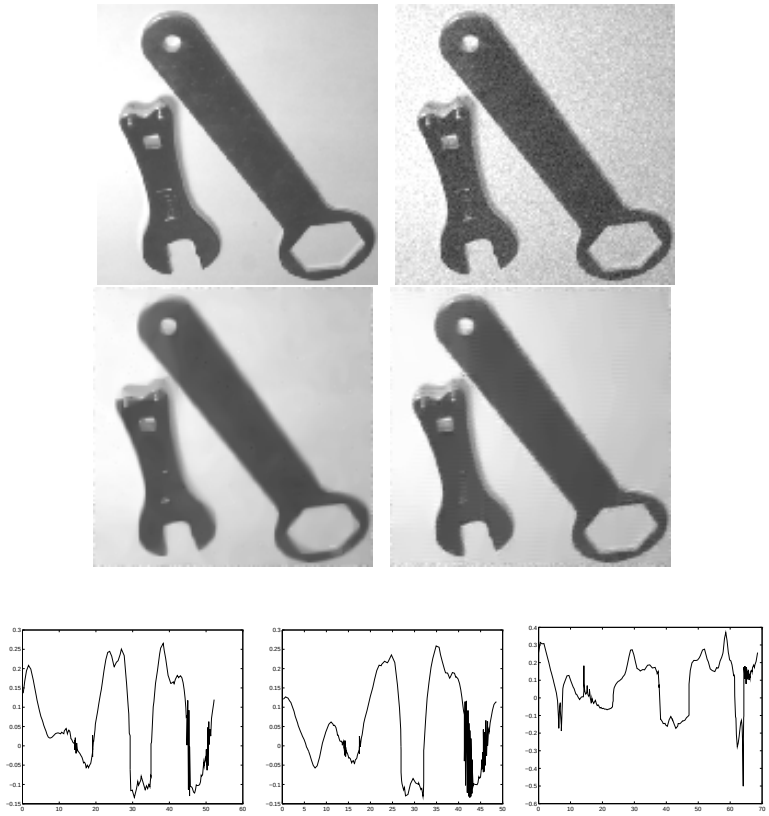
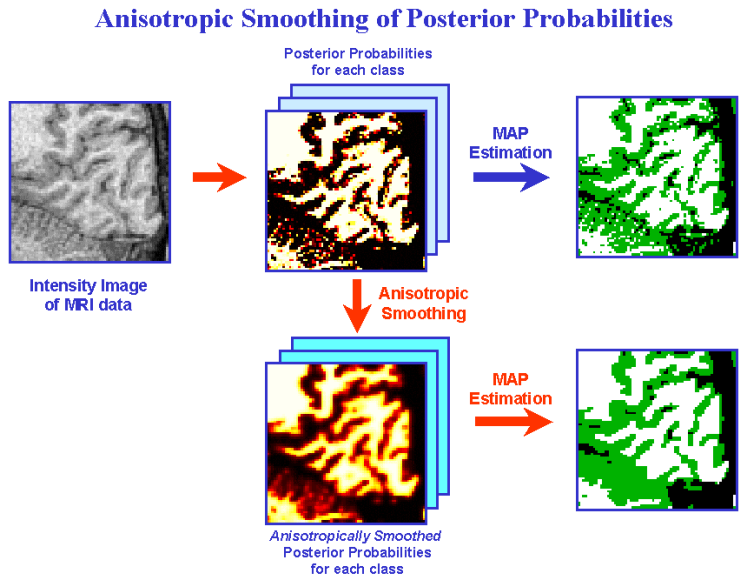
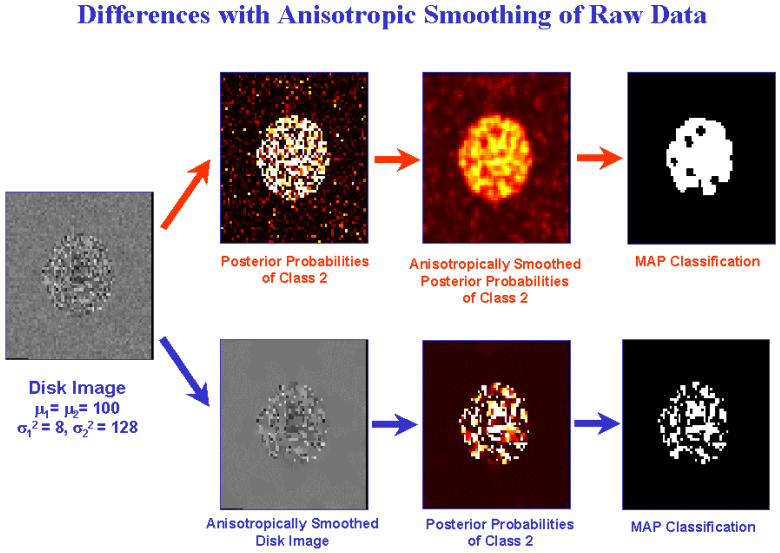


figure 4.19. *Test of the importance of affine invariant denoising. The first row shows the original image followed by the noisy one. The second row shows the denoising results for the affine and Euclidean flows. The third row shows plots of the affine curvature vs. affine arc-length of the mid-range level-set for the original image and those obtained from the affine and Euclidean denoising algorithms (second row images). The affine curvature was computed using implicit functions. The affine arc-length was computed using the relation between affine and Euclidean arc-lengths described in the text. In both cases, the curve was smoothed with the corresponding affine and Euclidean geometric heat flows for a small number of steps to avoid large noise in the discrete computations.*



a



b

figure 4.20. *(a) Schematic description of the posterior diffusion algorithm. (b) Toy example of the posterior diffusion algorithm. Two classes of the same average and different standard deviation are present in the image. The first row show the result of the proposed algorithm (posterior, diffusion, MAP), while the second row shows the result of classical techniques (diffusion, posterior, MAP).*

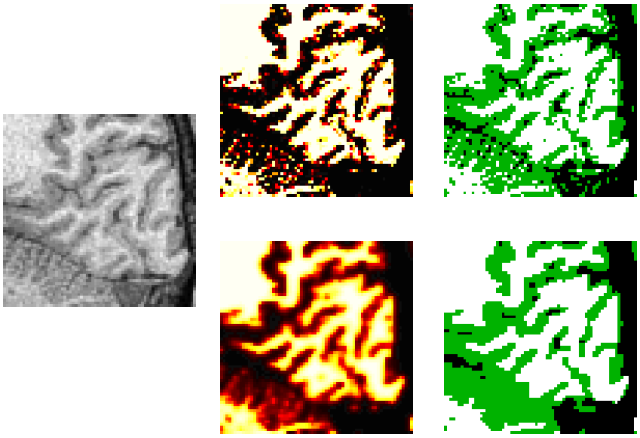


figure 4.21. *(Top row) Left: Intensity image of MRI data. Middle: Image of posterior probabilities corresponding to white matter class. Right: Image of corresponding MAP classification. Brighter regions in the posterior image correspond to areas with higher probability. White regions in the classification image correspond to areas classified as white matter; black regions correspond to areas classified as CSF. (Bottom row) Left: Image of white matter posterior probabilities after being anisotropically smoothed. Right: Image of MAP classification computed using smoothed posteriors.*

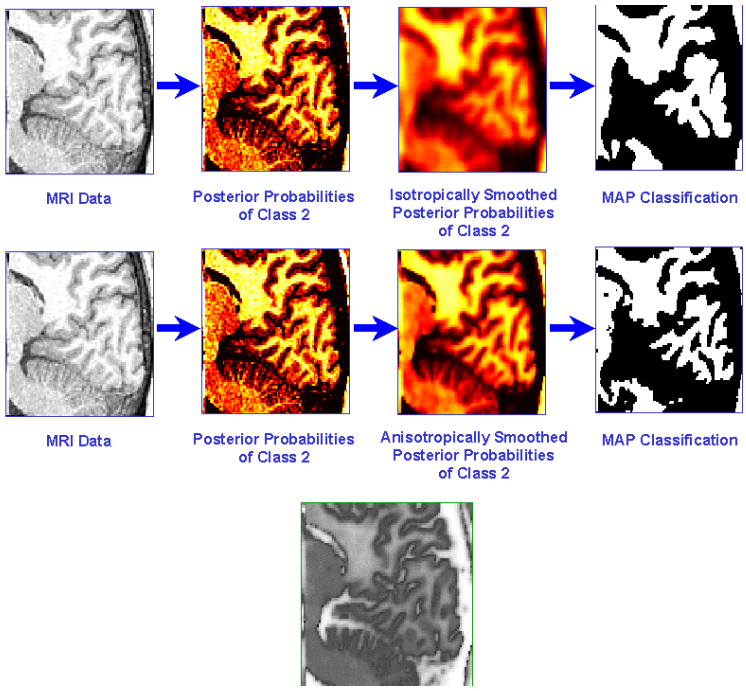


figure 4.22. *Comparison between isotropic and anisotropic diffusion of the posterior, and the variable weight $w_{i,j}$ (last row) for one of the classes in the anisotropic case.*

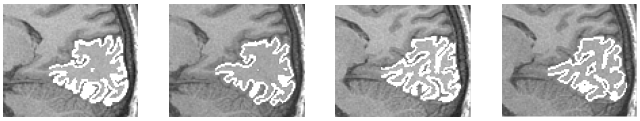


figure 4.23. *The two left images show manual gray matter segmentation results; the two right images show the automatically computed gray matter segmentation (same slices shown).*

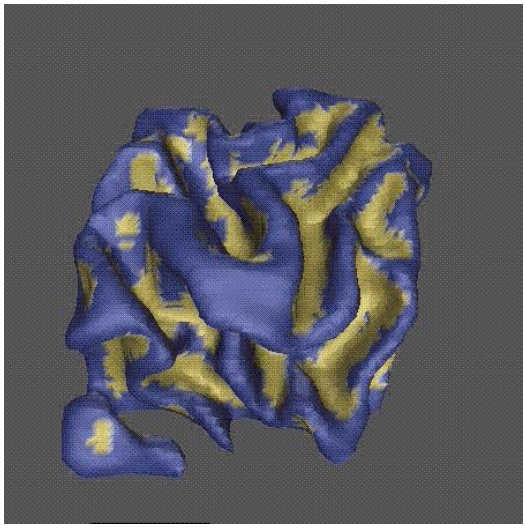


figure 4.24. *3D reconstruction of the human white matter from MRI, based on posterior diffusion techniques.*

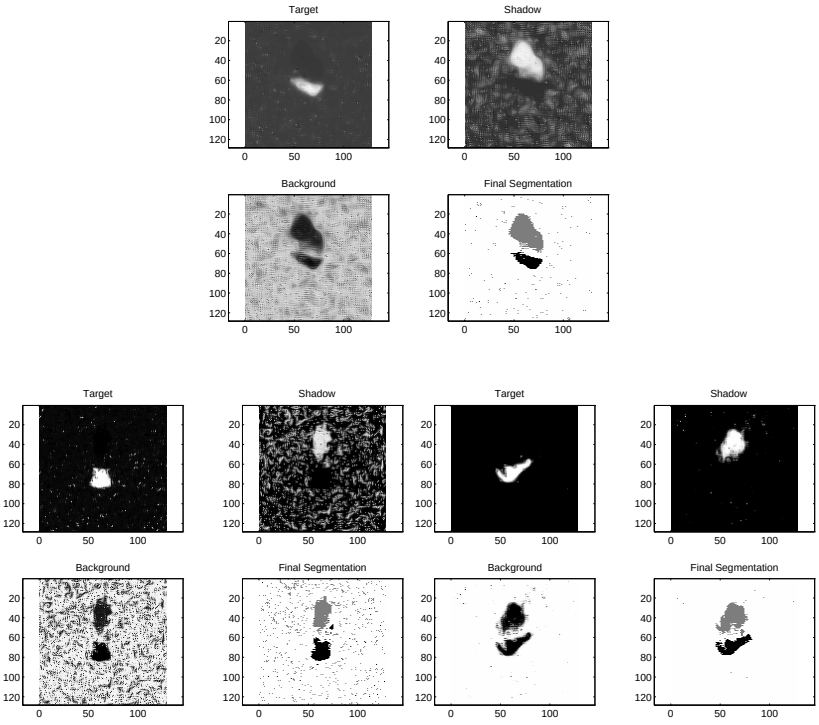


figure 4.25. *Segmentation of SAR data (scalar and video).*

figure 4.26. *Gray-level self-snakes. Note how the image is simplified while preserving the main structures.*

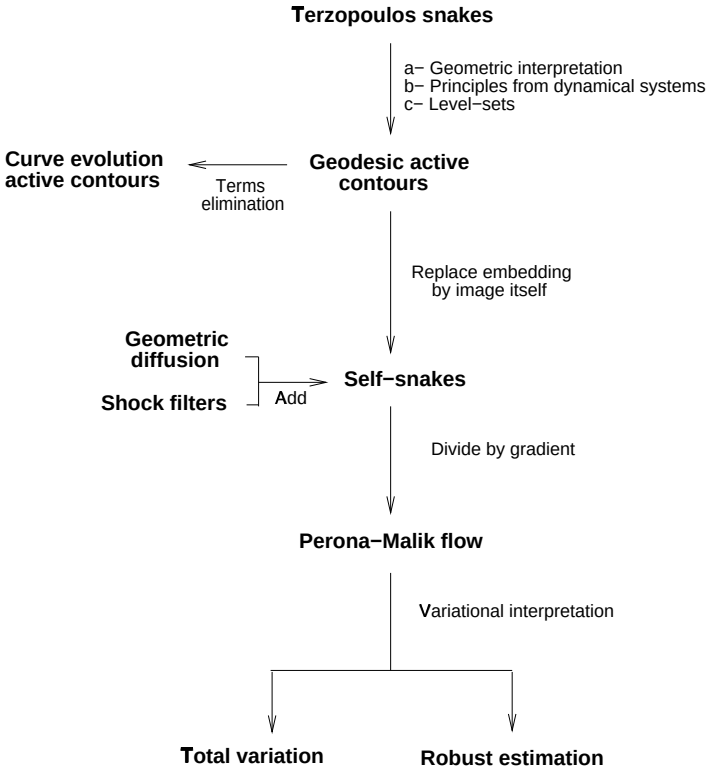


figure 4.27. *Connections between some of the basic PDE's used in image processing.*