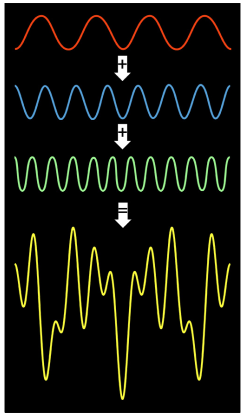
# BIL 717
# Image Processing
Mar. 5, 2014

Erkut Erdem
Dept. of Computer Engineering
Hacettepe University

Nonlinear Filtering

## Review - Signals

- A signal is composed of low and high frequency components



low frequency components: smooth / piecewise smooth

Neighboring pixels have similar brightness values

You're within a region

high frequency components: oscillatory

Neighboring pixels have different brightness values

You're either at the edges or noise points

## Review - Linear Diffusion

- The linear diffusion (heat) equation is the oldest and best investigated PDE method in image processing.
- Let $f(x)$ denote a grayscale (noisy) input image and $u(x, t)$ be initialized with $u(x,0) = u^0(x) = f(x)$.
- The linear diffusion process can be defined by the equation:

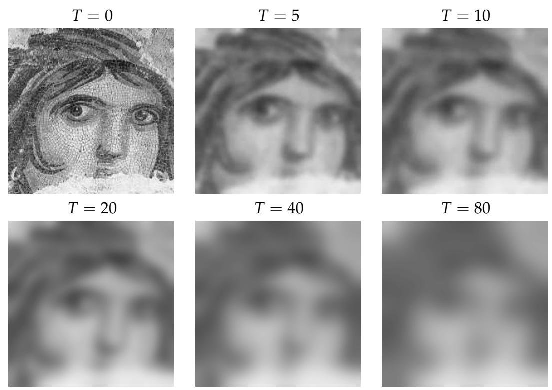
$$\frac{\partial u}{\partial t} = \nabla \cdot (\nabla u) = \nabla^2 u$$

where $\nabla \cdot$ denotes the divergence operator. Thus,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

## Review - Linear Diffusion (cont'd.)

- As we move to coarser scales,
  - the evolving images become more and more simplified since the diffusion process removes the image structures at finer scales.



## Review - Linear Diffusion and Gaussian Filtering

- The solution of the linear diffusion can be explicitly estimated as:

$$u(x,T) = \left(G_{\sqrt{2T}} * f\right)(x)$$

with $G_\sigma(x) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{|x|^2}{2\sigma^2}\right)$

- Solution of the linear diffusion equation is equivalent to a proper convolution of the input image with the Gaussian kernel $G_\sigma(x)$ with standard deviation $\sigma = \sqrt{2T}$
- The higher the value of $T$, the higher the value of $\sigma$, and the more smooth the image becomes.

## Review - Numerical Implementation

- Original model:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

- Space discrete version:

$$\frac{du_{i,j}}{dt} = u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}$$

- Space-time discrete version:

$$\frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} = u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k - 4u_{i,j}^k$$

homogeneous Neumann boundary condition along the image boundary

$\Delta t \leq 0.25$ is required for numerical stability

## Variational interpretation of heat diffusion

- Cost functional:

$$\begin{aligned} E[u] &= \iint_\Omega \|\nabla u\|^2 dxdy \\ &= \iint_\Omega \left(u_x^2 + u_y^2\right) dxdy \end{aligned}$$

- Euler-Lagrange:

$$\begin{aligned} \frac{\delta E}{\delta u} &= \frac{\partial E}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial E}{\partial u_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial E}{\partial u_y}\right) \\ &= -2\frac{\partial u_x}{\partial x} - 2\frac{\partial u_y}{\partial y} \\ &= -2(u_{xx} + u_{yy}) \end{aligned}$$

- Heat diffusion: modifies temperature to decrease E quickly

Slide credit: I. Kokkinos

## Today – Nonlinear Diffusion

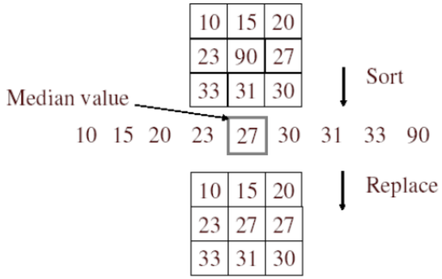
- Median filter
- use nonlinear PDEs to create a scale space representation
  - consists of gradually simplified images
  - some image features such as edges are maintained or even enhanced.
- Perona-Malik Type Nonlinear Diffusion (1990)
- Total Variation (TV) Regularization (1992)
- Weickert's Edge Enhancing Diffusion (1994)

## Median filters

- A **Median Filter** operates over a window by selecting the median intensity in the window.
- What advantage does a median filter have over a mean filter?
- Is a median filter a kind of convolution?

adapted from: S. Seitz

## Median filter

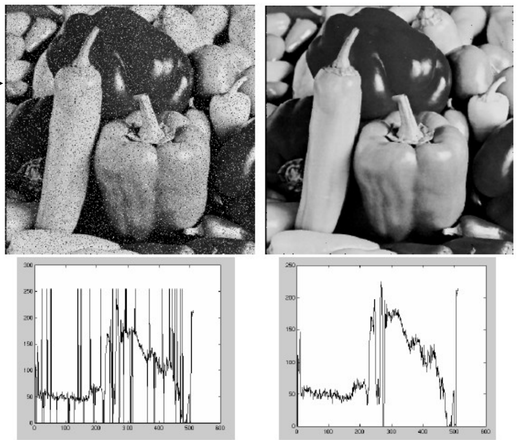



- No new pixel values introduced
- Removes spikes: good for impulse, salt & pepper noise
- Non-linear filter

Slide credit: K. Grauman

## Median filter

Salt and pepper noise



Median filtered

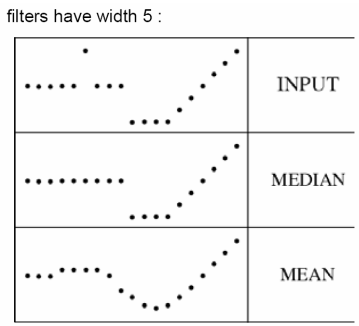
Plots of a row of the image

Matlab: `output im = medfilt2(im, [h w]);`

Slide credit: M. Hebert

## Median filter

- What advantage does median filtering have over Gaussian filtering?
  - Robustness to outliers
  - Median filter is edge preserving




Slide credit: K. Grauman

## Perona-Malik Type Nonlinear Diffusion

- The earliest nonlinear diffusion model proposed in image processing.
- called *anisotropic diffusion* by Perona and Malik.
- It uses a scalar-valued diffusivity.
- In fact, it is an isotropic nonhomogeneous equation.
  - A true example of anisotropic diffusion model: Weickert's Edge-enhancing diffusion (more later on)

## Perona-Malik Type Nonlinear Diffusion

- The Perona-Malik equation is:

$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u|)\nabla u)$$

  with homogeneous Neumann boundary conditions and the initial condition $u0(x) = f(x)$, $f$ denoting the input image.
- Constant diffusion coefficient of linear equation is replaced with a smooth non-increasing diffusivity function $g$ satisfying
  - $g(0) = 1$,
  - $g(s) \geq 0$,
  - $\lim_{s\to\infty} g(s) = 0$
- <u>The diffusivities become variable in both space and time (image dependent)</u>.

## Perona-Malik Type Nonlinear Diffusion

- The Perona-Malik equation: $\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u|)\nabla u)$
- Two different choices for the diffusivity function:

$$(1) \quad g(s) = \frac{1}{1+s^2/\lambda^2}$$

$$(2) \quad g(s) = e^{-\frac{s^2}{\lambda^2}}$$

- $\lambda$ corresponds to a contrast parameter.
- What is the effect of the parameter $\lambda$?

## 1D Analysis of Perona-Malik Diffusion

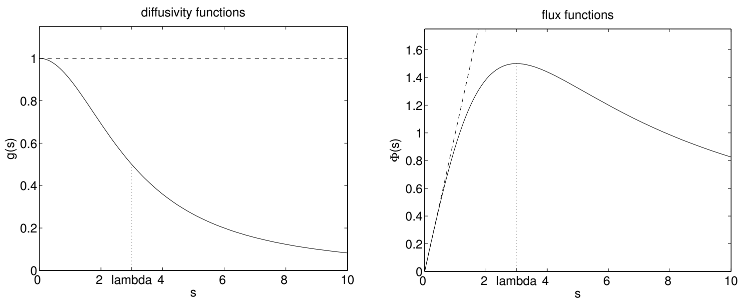
- 1D version to demonstrate the role of the contrast parameter
- For 1D case, the Perona-Malik equation is as follows:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\underbrace{(g(|u_x|)u_x)}_{\Phi(u_x)} = \Phi'(u_x)u_{xx}$$

with $g(|u_x|) = \frac{1}{1+|u_x|^2/\lambda^2}$ or $g(|u_x|) = e^{-\frac{|u_x|^2}{\lambda^2}}$

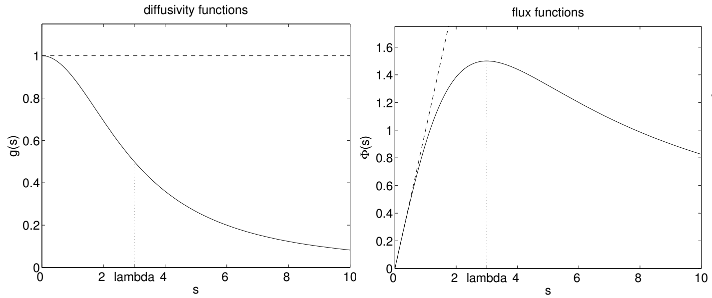
# 1D Analysis of Perona-Malik Diffusion

- Diffusivities and the corresponding flux functions for the linear diffusion (*plotted in dashed line*) and the Perona-Malik type nonlinear diffusion (*plotted in solid line*).




$$g(s) = \frac{1}{1+s^2/\lambda^2}$$
$$\lambda = 3$$

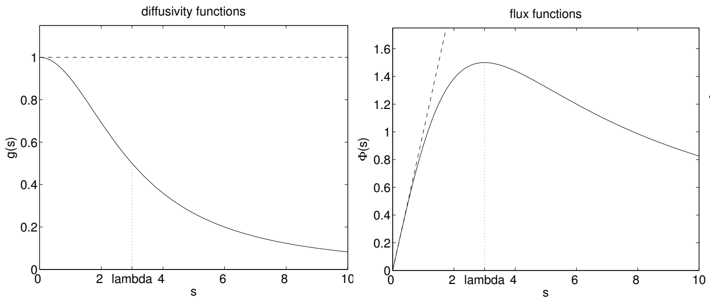
# 1D Analysis of Perona-Malik Diffusion




$$g(s) = \frac{1}{1+s^2/\lambda^2}$$
$$\lambda = 3$$

- For linear diffusion the diffusivity is constant ($g(s) = 1$), which results in a linearly increasing flux function.
- For linear diffusion all points, including the discontinuities, are smoothed equally.

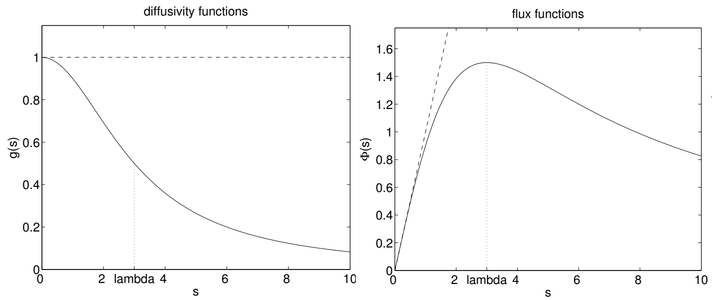
# 1D Analysis of Perona-Malik Diffusion




$$g(s) = \frac{1}{1+s^2/\lambda^2}$$
$$\lambda = 3$$

- For Perona-Malik, the diffusivity is variable and decreases as $|u_x|$ increases.
- The decay in diffusivity is particularly rapid after the contrast parameter $\lambda$.
- This leads to two different behaviors in the diffusion process.

# 1D Analysis of Perona-Malik Diffusion

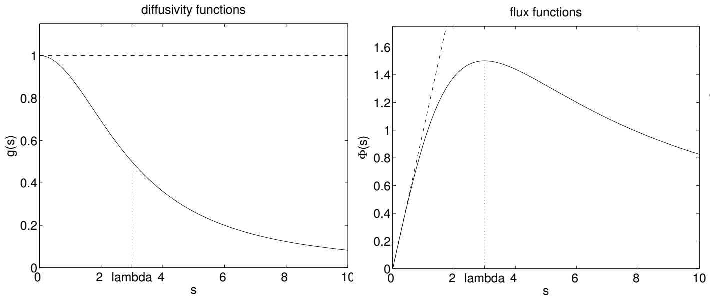



$$g(s) = \frac{1}{1+s^2/\lambda^2}$$
$$\lambda = 3$$

$$\frac{\partial u}{\partial t} = \Phi'(u_x)u_{xx}$$

- For the points where $|u_x| < \lambda$, $\Phi'(u_x) > 0$ we have lost in the material.
- For the points where $|u_x| > \lambda$, on the contrary, $\Phi'(u_x) < 0$ which generates an enhancement in the material.

# 1D Analysis of Perona-Malik Diffusion




$g(s) = \frac{1}{1+s^2/\lambda^2}$

$\lambda = 3$

$$\frac{\partial u}{\partial t} = \Phi'(u_x)u_{xx}$$

- Although the diffusivity is always nonnegative, one can observe both *forward* and *backward* diffusions during the smoothing.
- The contrast parameter $\lambda$ separates the regions of forward diffusion from the regions of backward diffusion.

# Perona-Malik Type Nonlinear Diffusion

- In 2D case, the diffusivities are reduced at the image locations where $|\nabla u|^2$ is large.
- As $|\nabla u|^2$ can be interpreted as a measure of edge likelihood, this means that the amount of smoothing is low along image edges.
- The contrast parameter $\lambda$ specifies a measure that determines which edge points are to be preserved or blurred during the diffusion process.
- Even edges can be sharpened due to the local backward diffusion behavior as discussed for the 1D case.
- Since the backward diffusion is a well-known ill-posed process, this may cause an instability, the so-called *staircasing effect.*

# Staircasing Effect

- Due to backward diffusion, a piece-wise smooth region in the original image evolves into many unintuitive piecewise constant regions.

Original noisy image

Perona-Malik Diffusion

- A possible solution to this drawback is to use regularized gradients in diffusivity computations.

# Regularized Perona-Malik Model

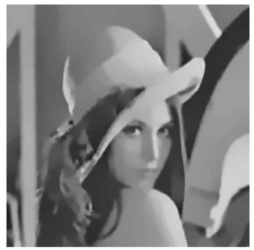
- Replacing the diffusivities $g(|\nabla u|)$ with the regularized ones $g(|\nabla u_\sigma|)$ leads to the following equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u_\sigma|)\nabla u)$$

where $u_\sigma = G_\sigma * u$ represents a Gaussian-smoothed version of the image.

Original noisy image

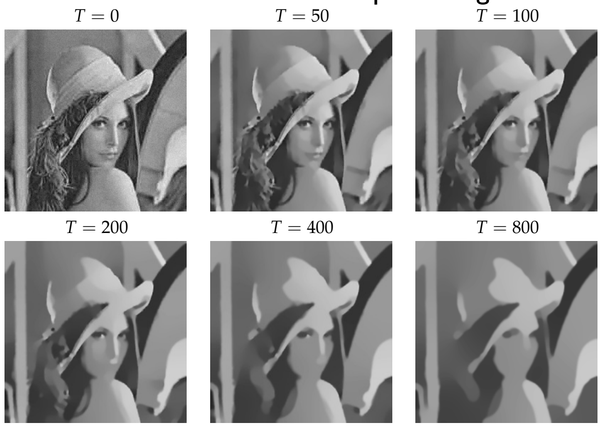
Perona-Malik Diffusion



Regularized Perona-Malik Diffusion

## Regularized Perona-Malik Model

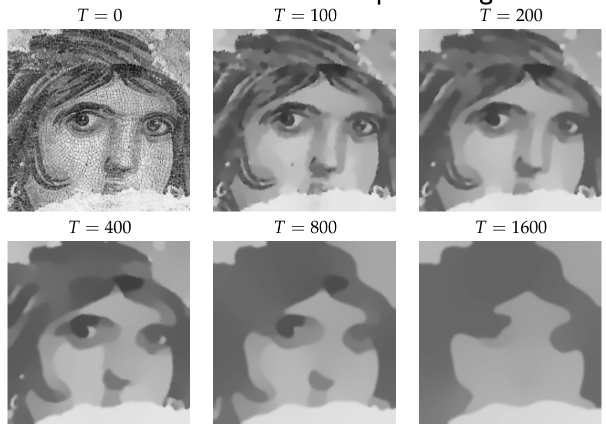
- Smoothing process diminishes noise while retaining or enhancing edges since it considers a kind of a priori edge knowledge




$(\lambda = 1, \sigma = 1)$

## Regularized Perona-Malik Model

- Smoothing process diminishes noise while retaining or enhancing edges since it considers a kind of a priori edge knowledge




$(\lambda = 1, \sigma = 1)$

## Numerical Implementation

- Central differences is used to approximate the gradient magnitude at a pixel (*i*, *j*) in the diffusivity estimation, $g_{i,j} = g(|\nabla u_{i,j}|)$

$$|\nabla u_{i,j}| = \sqrt{\left(\frac{du_{i,j}}{dx}\right)^2 + \left(\frac{du_{i,j}}{dy}\right)^2}$$

$$\approx \sqrt{\left(\frac{u_{i+1,j} - u_{i-1,j}}{2}\right)^2 + \left(\frac{u_{i,j+1} - u_{i,j-1}}{2}\right)^2}$$

## Numerical Implementation

- Original model:

$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u|)\nabla u)$$

- Space discrete version:

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}(g(|\nabla u|)u_x) + \frac{\partial}{\partial y}(g(|\nabla u|)u_y) \\
\frac{du_{i,j}}{dt} &= g_{i+\frac{1}{2},j} \cdot (u_{i+1,j} - u_{i,j}) - g_{i-\frac{1}{2},j} \cdot (u_{i,j} - u_{i-1,j}) \\
&+ g_{i,j+\frac{1}{2}} \cdot (u_{i,j+1} - u_{i,j}) - g_{i,j-\frac{1}{2}} \cdot (u_{i,j} - u_{i,j-1})
\end{aligned}$$

## Numerical Implementation

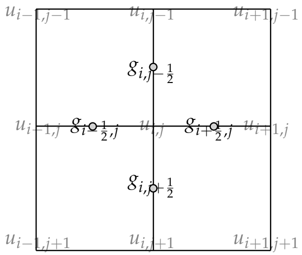
- Space discrete version:

$$\frac{du_{i,j}}{dt} = g_{i+\frac{1}{2},j}\cdot\left(u_{i+1,j}-u_{i,j}\right) - g_{i-\frac{1}{2},j}\cdot\left(u_{i,j}-u_{i-1,j}\right) + g_{i,j+\frac{1}{2}}\cdot\left(u_{i,j+1}-u_{i,j}\right) - g_{i,j-\frac{1}{2}}\cdot\left(u_{i,j}-u_{i,j-1}\right)$$

- This discretization scheme requires the diffusivities to be estimated at mid-pixel points.
- They are computed by taking averages of the diffusivities over neighboring pixels:

$$g_{i\pm\frac{1}{2},j} = \frac{g_{i\pm 1,j}+g_{i,j}}{2}$$

$$g_{i,j\pm\frac{1}{2}} = \frac{g_{i,j\pm 1}+g_{i,j}}{2}$$




## Numerical Implementation

- Space discrete version:

$$\frac{du_{i,j}}{dt} = g_{i+\frac{1}{2},j}\cdot\left(u_{i+1,j}-u_{i,j}\right) - g_{i-\frac{1}{2},j}\cdot\left(u_{i,j}-u_{i-1,j}\right) + g_{i,j+\frac{1}{2}}\cdot\left(u_{i,j+1}-u_{i,j}\right) - g_{i,j-\frac{1}{2}}\cdot\left(u_{i,j}-u_{i,j-1}\right)$$

- Space-time discrete version:

$$\frac{u_{i,j}^{k+1}-u_{i,j}^{k}}{\Delta t} = g^k_{i+\frac{1}{2},j}\cdot u^k_{i+1,j} + g^k_{i-\frac{1}{2},j}\cdot u^k_{i-1,j} + g^k_{i,j+\frac{1}{2}}\cdot u^k_{i,j+1} + g^k_{i,j-\frac{1}{2}}\cdot u^k_{i,j-1} - \left(g^k_{i+\frac{1}{2},j} + g^k_{i-\frac{1}{2},j} + g^k_{i,j+\frac{1}{2}} + g^k_{i,j-\frac{1}{2}}\right)\cdot u^k_{i,j}$$

homogeneous Neumann boundary condition along the image boundary

$\Delta t \leq 0.25$ is required for numerical stability

## Extension to vectorial images

- Extension of nonlinear diffusion to vectorial images:

$$\boldsymbol{u} = (u_1, u_2, \ldots, u_N)$$

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(g(\|\nabla u\|)\nabla u\right)$$

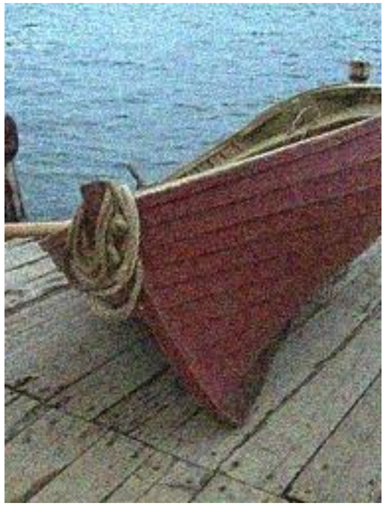
*generalization*

$$\boxed{\frac{\partial u_i}{\partial t} = \operatorname{div}\left(g(\|\nabla \boldsymbol{u}\|)\nabla u_i\right),\ i = 1,..,N}$$

where: $\|\nabla \boldsymbol{u}\| = \sqrt{\sum_{i=1}^{N} \|\nabla u_i\|^2}$

Slide credit: I. Kokkinos

## Perona-Malik results for color images



Slide credit: I. Kokkinos

## Total Variation (TV) Regularization

- Rudin et al. (1992) formulated image restoration as minimization of the total variation (TV) of a given image.
- The Total Variation (TV) regularization model is generally defined as:

$$E_{TV}(u) = \int_{\Omega} \left( \frac{1}{2}(u-f)^2 + \alpha |\nabla u| \right) dx$$

  - $\Omega \subset \mathbf{R}^2$ is connected, bounded, open subset representing the image domain,
  - $f$ is an image defined on $\Omega$,
  - $u$ is the smooth approximation of $f$,
  - $\alpha > 0$ is a scalar.

## Total Variation (TV) Regularization

- The Total Variation (TV) regularization model:

$$E_{TV}(u) = \int_{\Omega} \left( \frac{1}{2}(u-f)^2 + \alpha |\nabla u| \right) dx$$

- The gradient descent equation for Equation (10) is defined by:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) - \frac{1}{\alpha}(u-f); \quad \left.\frac{\partial u}{\partial n}\right|_{\partial\Omega} = 0$$

- The value of $\alpha$ specifies the relative importance of the fidelity term.
- It can be interpreted as a scale parameter that determines the level of smoothing.

## Sample TV Restoration results

$\alpha = 50$ $\alpha = 100$ $\alpha = 200$

- The value of $\alpha$ specifies the relative importance of the fidelity term and thus the level of smoothing.

## TV Regularization

- In the original formulation, the observed image $f$ was assumed to be degraded by additive Gaussian noise with zero mean and known variance $\sigma^2$.
- In order to restore a given image, Rudin et al. proposed to solve the following constrained optimization problem:

$$\min_u \int_{\Omega} |\nabla u| dx$$

subject to

$$\int_{\Omega} (u-f)^2 dx = \sigma^2$$

- $\frac{1}{\alpha}$ can be considered as a Lagrange multiplier.

# TV Regularization and TV Flow

- TV regularization can be associated with a nonlinear diffusion filter, the so-called *TV flow*
- Ignoring the fidelity term in the TV regularization model leads to the PDE:

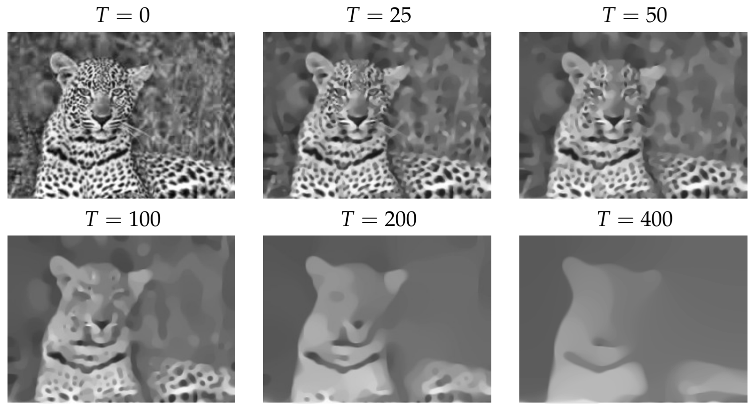
$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u|)\nabla u)$$

with $u^0 = f$ and the diffusivity function $g(|\nabla u|) = \frac{1}{|\nabla u|}$

- Notice that this diffusivity function has no additional contrast parameter as compared with the Perona-Malik diffusivities.

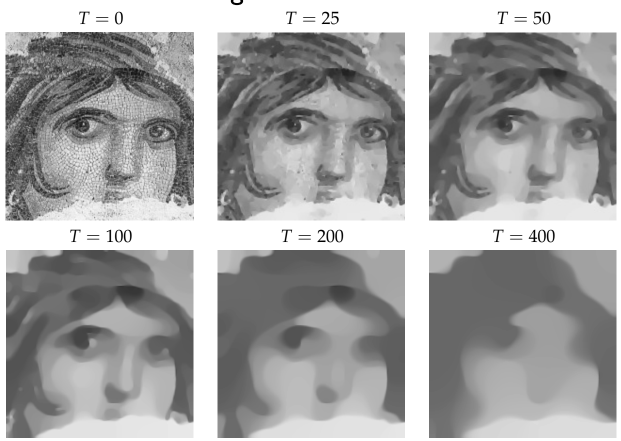
# Sample TV Flow results

- Corresponding smoothing process yields segmentation-like, piecewise constant images.




# Sample TV Flow results

- Corresponding smoothing process yields segmentation-like, piecewise constant images.




# Numerical Implementation

- The evolution equation can be discretized by using standard finite differences.
- The solution of TV regularization or equivalently TV flow leads to singular diffusivities.
- In numerical implementations based on standard discretization, this leads to stability problems as the image gradient tends to zero.
- A common solution to this problem is to add a small positive constant $\varepsilon$ to image gradients.
- More accurate numerical implementations are suggested.

## Numerical Implementation

- Space discrete version:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{|\nabla u|^2+\epsilon^2}}\right) + \frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{|\nabla u|^2+\epsilon^2}}\right) - \frac{1}{\alpha}(u-f)$$

$$= \frac{u_{xx}\left(u_y^2+\epsilon^2\right) - 2u_x u_y u_{xy} + u_{yy}\left(u_x^2+\epsilon^2\right)}{\left(u_x^2+u_y^2+\epsilon^2\right)^{\frac{3}{2}}} - \frac{1}{\alpha}(u-f)\,,$$

## Numerical Implementation

- Space discrete version:

$$\frac{du_{i,j}}{dt} = \frac{\frac{d^2u_{i,j}}{dx^2}\left(\left(\frac{du_{i,j}}{dy}\right)^2+\epsilon^2\right) - 2\left(\frac{du_{i,j}}{dx}\right)\left(\frac{du_{i,j}}{dy}\right)\left(\frac{d^2u_{i,j}}{dxdy}\right) + \frac{d^2u_{i,j}}{dy^2}\left(\left(\frac{du_{i,j}}{dx}\right)^2+\epsilon^2\right)}{\left(\left(\frac{du_{i,j}}{dx}\right)^2+\left(\frac{du_{i,j}}{dy}\right)^2+\epsilon^2\right)^{\frac{3}{2}}}$$

$$- \frac{1}{\alpha}\left(u_{i,j}-f_{i,j}\right)$$

with $\frac{d^2u_{i,j}}{dxdy} \approx \frac{u_{i+1,j+1}-u_{i+1,j-1}-u_{i-1,j+1}+u_{i-1,j-1}}{4}$

## Numerical Implementation

- Space-time discrete version:

$$\frac{u_{i,j}^{k+1}-u_{i,j}^k}{\Delta t} = \left(\left(\frac{u_{i+1,j}^k-u_{i-1,j}^k}{2}\right)^2 + \left(\frac{u_{i,j+1}^k-u_{i,j-1}^k}{2}\right)^2+\epsilon^2\right)^{-\frac{3}{2}}$$

$$\cdot\Bigg[\left(u_{i+1,j}^k-2u_{i,j}^k+u_{i-1,j}^k\right)\left(\left(\frac{u_{i,j+1}^k-u_{i,j-1}^k}{2}\right)^2+\epsilon^2\right)$$

$$-\frac{1}{8}\left(u_{i+1,j}^k-u_{i-1,j}^k\right)\left(u_{i,j+1}^k-u_{i,j-1}^k\right)$$

$$\left(u_{i+1,j+1}^k-u_{i+1,j-1}^k-u_{i-1,j+1}^k+u_{i-1,j-1}^k\right)$$

$$+\left(u_{i,j+1}^k-2u_{i,j}^k+u_{i,j-1}^k\right)\left(\left(\frac{u_{i+1,j}^k-u_{i-1,j}^k}{2}\right)^2+\epsilon^2\right)\Bigg]$$

$$-\frac{1}{\alpha}\left(u_{i,j}^k-f_{i,j}\right)$$

homogeneous Neumann boundary condition along the image boundary

$\Delta t \leq 0.25\,\varepsilon$ is required for numerical stability

## Structure Tensor

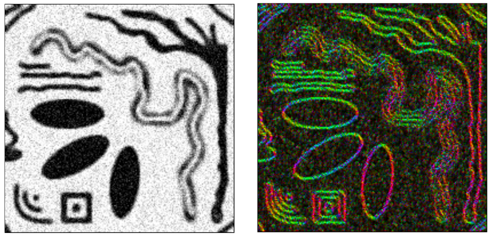
- The structure tensor $J(\nabla u)$ is described by:

$$J(\nabla u) = \nabla u \nabla u^T = \begin{bmatrix} u_x^2 & u_x u_y \\ u_x u_y & u_y^2 \end{bmatrix}$$

- The structure tensor $J(\nabla u)$ can be interpreted as an image feature describing the local orientation information.
- It has
  - an orthonormal basis of eigenvectors $v_1$ and $v_2$ with $v_1 \parallel \nabla u$ and $v_2 \perp \nabla u$, and
  - the corresponding eigenvalues $\lambda_1 = |\nabla u|^2$ and $\lambda_2 = 0$.

# Structure Tensor

$$J(\nabla u) = \nabla u \nabla u^T = \begin{bmatrix} u_x^2 & u_x u_y \\ u_x u_y & u_y^2 \end{bmatrix}$$



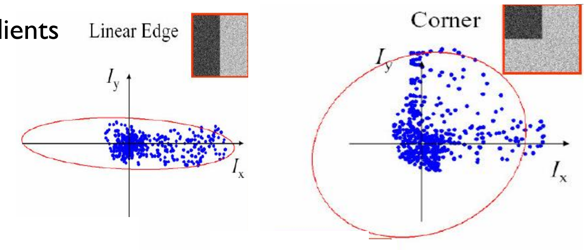
$J(\nabla u) = \nabla u \nabla u^T$

Images are taken from Brox et al., 2004

# Structure Tensor

- aka second moment matrix

Distribution of gradients



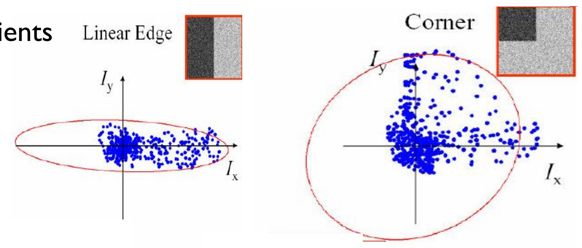

$$J = \begin{bmatrix} \sum_{x',y'} I_x^2 & \sum_{x',y'} I_x I_y \\ \sum_{x',y'} I_x I_y & \sum_{x',y'} I_y^2 \end{bmatrix}$$

Slide credit: I. Kokkinos

# Structure Tensor

- aka second moment matrix

Distribution of gradients

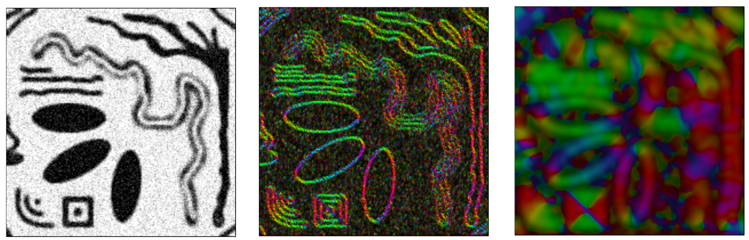



$$J = \sum_{x',y'} (I_x, I_y)^T (I_x, I_y)$$

Slide credit: I. Kokkinos

# Linear Structure Tensor

- Noise significantly affects the tensor estimation.
- The given image $u$ is usually convolved with a Gaussian kernel $G_\sigma$ with a relatively small standard deviation $\sigma$
- The (linear) structure tensor is computed accordingly by using $\nabla u_\sigma = \nabla(G_\sigma * u)$ instead of $\nabla u$.

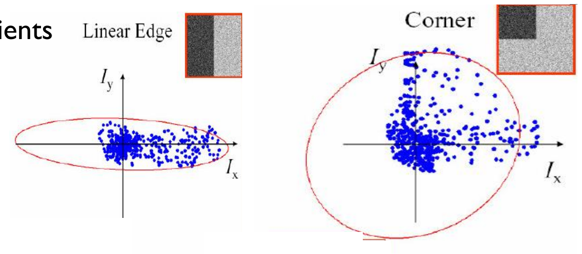


$J(\nabla u) = \nabla u \nabla u^T$ $\quad$ $J_\rho$ with $\rho = 3$

Images are taken from Brox et al., 2004

## Structure Tensor

Distribution of gradients



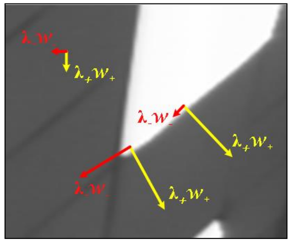

$$J = \sum_{x',y'} (\nabla G_\sigma * u)^T (\nabla G_\sigma * u)$$

Slide credit: I. Kokkinos

## Structure Tensor

$$J = \sum_{x',y'} (\nabla G_\sigma * u)^T (\nabla G_\sigma * u)$$

- Eigenvectors $w_+$, $w_-$: directions of maximal and minimal variation of $u$
- Eigenvalues: amounts of minimal and maximal variation $u$




Slide credit: I. Kokkinos

## Edge Enhancing Diffusion

- Proposed by Weickert (1994)
- an anisotropic nonlinear diffusion model with better edge enhancing capabilities than the Perona-Malik model
- can be described by the equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(\nabla u)\nabla u)$$

where
- $u$ is the smoothed image,
- $f$ is the input image ($u^0(x) = f(x)$),
- $D$ represents a matrix-valued diffusion tensor that describes the smoothing directions and the corresponding diffusivities

## Edge Enhancing Diffusion

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(\nabla u)\nabla u)$$

- For linear diffusion the diffusion tensor can be defined as $D(\nabla u) = I$ with $I$ denoting the identity matrix.
  - This results in a constant diffusion coefficient for all image points in all directions.
- For Perona-Malik type nonlinear diffusion, $D(\nabla u) = g(|\nabla u_\sigma|)I$.
  - Such a choice reduces the amount of smoothing at image edges, but in an equal amount in all directions.
- In actual anisotropic setting, the diffusion tensor $D$ is defined as a function of the structure tensor $J(\nabla u)$.

## Edge Enhancing Diffusion

- use the structure tensor as an image/edge descriptor to construct a diffusion tensor that
  - reduces the amount of smoothing across the edges
  - while smoothing is still carried out along the edges
- Weickert proposed to utilize same orthonormal basis of eigenvectors $v_1 \parallel \nabla u_\sigma$ and $v_2 \perp \nabla u_\sigma$ estimated from the structure tensor $J(\nabla u_\sigma)$ with the following choice of eigenvalues satisfying
  - $\frac{\lambda_1(|\nabla u_\sigma|)}{\lambda_2(|\nabla u_\sigma|)} \to 0$ for $|\nabla u_\sigma| \to \infty$

## Edge Enhancing Diffusion

- Suggested eigenvalues are

$$\lambda_1(|\nabla u_\sigma|) = \begin{cases} 1 & \text{if } |\nabla u_\sigma| = 0 \\ 1 - exp\left(-\frac{3.31488}{(|\nabla u_\sigma|/\lambda)^8}\right) & \text{otherwise,} \end{cases}$$

$$\lambda_2(|\nabla u_\sigma|) = 1$$

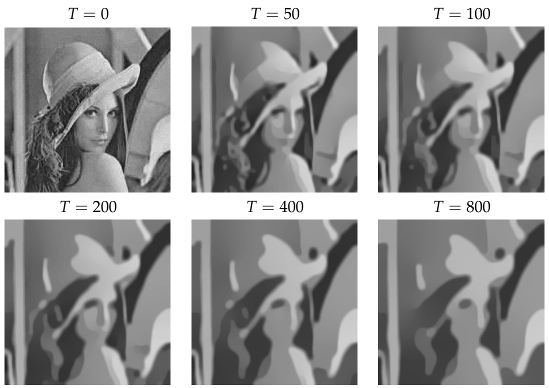
where $\lambda$ denotes the contrast parameter.

- preserves and enhances image edges by reducing the diffusivity $\lambda_1$ perpendicular to edges for sufficiently large values of $|\nabla u_\sigma|$.
- Specifically, the diffusion tensor is given by the formula:

$$D = \begin{bmatrix} (u_\sigma)_x & -(u_\sigma)_y \\ (u_\sigma)_y & (u_\sigma)_x \end{bmatrix} \cdot \begin{bmatrix} \lambda_1(|\nabla u_\sigma|) & 0 \\ 0 & \lambda_2(|\nabla u_\sigma|) \end{bmatrix} \cdot \begin{bmatrix} (u_\sigma)_x & -(u_\sigma)_y \\ (u_\sigma)_y & (u_\sigma)_x \end{bmatrix}^{-1}$$

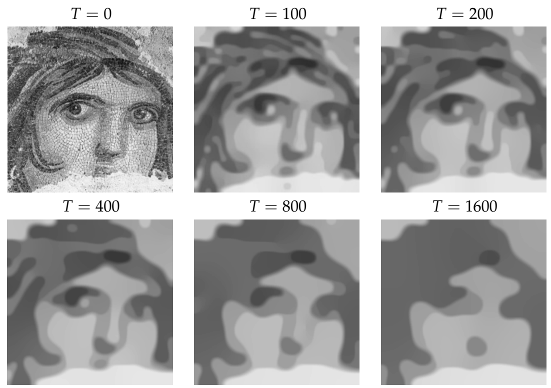
## Sample Results of Edge Enhancing Diffusion

- Smoothing process diminishes noise and fine image details while retaining and enhancing edges as in the Perona-Malik type nonlinear diffusion.




## Sample Results of Edge Enhancing Diffusion

- Corners become more rounded in the anisotropic model compared to the Perona-Malik filter.
- Smoothing along edges and not across them causes a slight shrinking effect in the image structures, eliminating fine or thin structures




$(\lambda = 1.8, \ \sigma = 1)$