Case Study 1

Image filtering using diffusion models

Alfred Ajay Aureate R[1]

[1]EE10B052
Department of Electrical Engineering
Indian Institute of Technology Madras

MA5710, Dec 2014

Outline

1. Models Description
   - Linear Diffusion model
   - Non-linear Perona-Malik diffusion model
   - Non-linear Edge enhancing diffusion model
2. Implementation of models
   - Linear diffusion model
   - Perona Malik model
   - PMC model
   - Edge enhancing diffusion model

Diffusion models description

- The intensity of each pixel of an image is collectively assumed as a matrix U.

Diffusion models description

- The intensity of each pixel of an image is collectively assumed as a matrix U.
- Now, when noise is added to this image, it creates glitches and discontinuities at random pixels. So, by using a diffusion model, we try to smooth-en those discontinuities.

Diffusion models description

- The intensity of each pixel of an image is collectively assumed as a matrix U.
- Now, when noise is added to this image, it creates glitches and discontinuities at random pixels. So, by using a diffusion model, we try to smooth-en those discontinuities.
- Some invariant properties for image:
  - Clean and translate, and translate and clean should give same output
  - Clean and rotate, and rotate and clean should give same output
  - Clean and zoom, and zoom and clean should give same output
  - When a constant is added, we should be able to remove it

Diffusion models description

- The intensity of each pixel of an image is collectively assumed as a matrix U.
- Now, when noise is added to this image, it creates glitches and discontinuities at random pixels. So, by using a diffusion model, we try to smooth-en those discontinuities.
- Some invariant properties for image:
  - Clean and translate, and translate and clean should give same output
  - Clean and rotate, and rotate and clean should give same output
  - Clean and zoom, and zoom and clean should give same output
  - When a constant is added, we should be able to remove it
- $U_t = \nabla.\left(D\left(U\right)\nabla U\right)$; $U\left(x, y, 0\right) = U_0\left(x, y\right)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$.

Outline

1. Models Description
   - Linear Diffusion model
   - Non-linear Perona-Malik diffusion model
   - Non-linear Edge enhancing diffusion model

2. Implementation of models
   - Linear diffusion model
   - Perona Malik model
   - PMC model
   - Edge enhancing diffusion model

Linear diffusion model

- Here $D(U) = I$.

Linear diffusion model

- Here $D(U) = I$.
- So, the model becomes $U_t = \nabla.(I\nabla U)$; $U(x, y, 0) = U_0(x, y)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$.

Linear diffusion model

- Here $D(U) = I$.
- So, the model becomes $U_t = \nabla.(I \nabla U)$; $U(x, y, 0) = U_0(x, y)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$.
- $U(x, y, 0) = U_0(x, y)$. It is the noise-added image before filtering

Linear diffusion model

- Here $D\left(U\right) = I$.
- So, the model becomes $U_t = \nabla.\left(I\nabla U\right)$; $U\left(x,y,0\right) = U_0\left(x,y\right)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$.
- $U\left(x,y,0\right) = U_0\left(x,y\right)$. It is the noise-added image before filtering
- Here, we use Neumann boundary conditions for images

Linear diffusion model

- Here $D(U) = I$.
- So, the model becomes $U_t = \nabla.(I \nabla U)$; $U(x, y, 0) = U_0(x, y)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$.
- $U(x, y, 0) = U_0(x, y)$. It is the noise-added image before filtering
- Here, we use Neumann boundary conditions for images
- It can be proved that the solution for the above equation could be given by:

$$U(x, y, t) = \begin{cases} U_0(x, y) & (t = 0) \\ (k_{\sqrt{2t}} * U_0)(x, y) & (t > 0) \end{cases}$$

where $k_{\sqrt{2t}}(x, y)$ is the gaussian kernel with variance $\sigma = \sqrt{2t}$

Linear diffusion model

-

Outline

1. Models Description
   - Linear Diffusion model
   - Non-linear Perona-Malik diffusion model
   - Non-linear Edge enhancing diffusion model

2. Implementation of models
   - Linear diffusion model
   - Perona Malik model
   - PMC model
   - Edge enhancing diffusion model

Perona-Malik diffusion model

- This is part of the non-linear scalar diffusion model. So, $D\left(U\right) = c\left(\|\nabla U\|\right) I$, where $c\left(\|\nabla U\|\right) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$.

Perona-Malik diffusion model

- This is part of the non-linear scalar diffusion model. So, $D(U) = c(\|\nabla U\|)\, I$, where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$.
- So, the model becomes $U_t = \nabla . \left(c(\|\nabla U\|)\, I \nabla U\right)$; $U(x, y, 0) = U_0(x, y)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$; where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$

Perona-Malik diffusion model

- This is part of the non-linear scalar diffusion model. So, $D\left(U\right)=c\left(\|\nabla U\|\right)I$, where $c\left(\|\nabla U\|\right)=\frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$.
- So, the model becomes $U_t=\nabla.\left(c\left(\|\nabla U\|\right)I\nabla U\right)$; $U\left(x,y,0\right)=U_0\left(x,y\right)$; $\frac{\partial U}{\partial n}=0$ on $\partial\Omega$; where $c\left(\|\nabla U\|\right)=\frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$
- $U\left(x,y,0\right)=U_0\left(x,y\right)$. It is the noise-added image before filtering

Perona-Malik diffusion model

- This is part of the non-linear scalar diffusion model. So, $D(U) = c(\|\nabla U\|) I$, where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$.
- So, the model becomes $U_t = \nabla . (c(\|\nabla U\|) I \nabla U)$; $U(x, y, 0) = U_0(x, y)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$; where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$
- $U(x, y, 0) = U_0(x, y)$. It is the noise-added image before filtering
- Here, we use Neumann boundary conditions for images

Perona-Malik diffusion model

- This is part of the non-linear scalar diffusion model. So, $D(U) = c(\|\nabla U\|)\, I$, where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$.
- So, the model becomes $U_t = \nabla.\left(c(\|\nabla U\|)\, I \nabla U\right)$; $U(x,y,0) = U_0(x,y)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$; where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$
- $U(x,y,0) = U_0(x,y)$. It is the noise-added image before filtering
- Here, we use Neumann boundary conditions for images
- Catte extended this model, which is now called the PMC diffusion model.

Perona-Malik diffusion model

- This is part of the non-linear scalar diffusion model. So, $D(U) = c(\|\nabla U\|)\, I$, where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$.
- So, the model becomes $U_t = \nabla.\left(c(\|\nabla U\|)\, I \nabla U\right)$; $U(x,y,0) = U_0(x,y)$; $\frac{\partial U}{\partial n} = 0$ on $\partial\Omega$; where $c(\|\nabla U\|) = \frac{1}{1+\frac{\|\nabla U\|^2}{\lambda^2}}$
- $U(x,y,0) = U_0(x,y)$. It is the noise-added image before filtering
- Here, we use Neumann boundary conditions for images
- Catte extended this model, which is now called the PMC diffusion model. Basically, here PM model is applied to U_σ, which is the solution of the linear diffusion model, rather than using just U for the PM model. So, in PMC model, $U_t = \nabla.\left(c(\|\nabla U_\sigma\|)\, I \nabla U_\sigma\right)$. U_σ is the solution got from the linear diffusion model, where U_0 is convolved with a gaussian kernel of variance σ.

Outline

1. Models Description
   - Linear Diffusion model
   - Non-linear Perona-Malik diffusion model
   - Non-linear Edge enhancing diffusion model

2. Implementation of models
   - Linear diffusion model
   - Perona Malik model
   - PMC model
   - Edge enhancing diffusion model

Edge enhancing diffusion model

- This is part of the non-linear tensor diffusion model. So, $D = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

Edge enhancing diffusion model

- This is part of the non-linear tensor diffusion model. So, $D = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.
- Now, for the edge-enhancing diffusion model, the diffusion tensor is given by:

$$D = R^T \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} R$$

where R is the rotation matrix describing the local coordinate system aligned with the gradient vector observed at scale u.

Edge enhancing diffusion model

- This is part of the non-linear tensor diffusion model. So, $D = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.
- Now, for the edge-enhancing diffusion model, the diffusion tensor is given by:
$$D = R^T \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} R$$
where R is the rotation matrix describing the local coordinate system aligned with the gradient vector observed at scale u.
- $$R = \frac{1}{\sqrt{(L_x^u)^2 + \left(L_y^u\right)^2}} \begin{pmatrix} L_x^u & -L_y^u \\ L_y^u & L_x^u \end{pmatrix}$$
where L^u denotes the image observed at scale u.

Edge enhancing diffusion model

- c_1 is the conductivity in the direction of the gradient (observed at scale u) and the c_2 is the conductivity along the isophote. Inorder to compare edge-enhancing diffusion with scalar difusion (Perona and Malik type) we set the diffusion along the edge to be equal to the isotropic diffusion in the Perona malik diffusion discussed earlier and set the conductivity across the edge to be one fifth of the conductivity along the edge.

$$c_2\left(L_w^u\right) = \frac{1}{1 + \frac{\left(L_w^u\right)^2}{\lambda^2}}$$

$$c_1\left(L_w^u\right) = \frac{1}{5} c_2\left(L_w^u\right)$$

Here, $L_w^u = \sqrt{\left(L_x^u\right)^2 + \left(L_y^u\right)^2}$ is the gradient norm.

Outline

1. Models Description
   - Linear Diffusion model
   - Non-linear Perona-Malik diffusion model
   - Non-linear Edge enhancing diffusion model

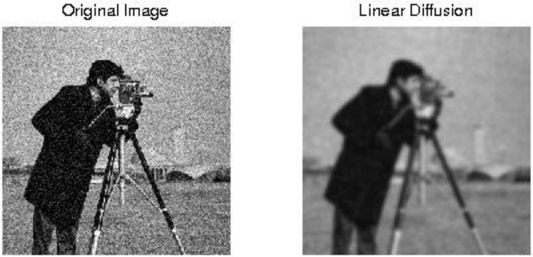
2. Implementation of models
   - Linear diffusion model
   - Perona Malik model
   - PMC model
   - Edge enhancing diffusion model

Linear diffusion model

- The following line implements linear diffusion model in the code:

$$g = gD(g, 0.4, 0, 0)$$

Figure: Original noisy image vs Linear Diffused image

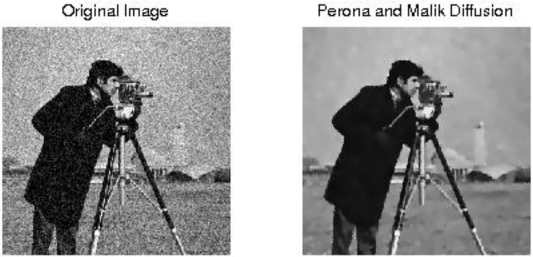


Outline

Perona Malik model

- All the terms involving convolution of u with a gaussian kernel are commented or removed

Figure: Original noisy image vs Perona Malik Diffused image

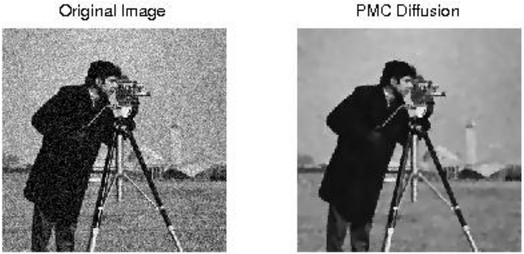


Outline

PMC model

- Similar to the previous one, but, first u is convolved with the gaussian kernel and the resulting u_σ is used like in the previous case.

Figure: Original noisy image vs PMC Diffused image



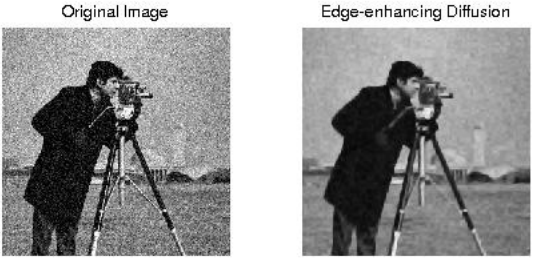
Outline

1. Models Description
   - Linear Diffusion model
   - Non-linear Perona-Malik diffusion model
   - Non-linear Edge enhancing diffusion model
2. Implementation of models
   - Linear diffusion model
   - Perona Malik model
   - PMC model
   - Edge enhancing diffusion model

Edge enhancing diffusion model

- The following lines are added:
  " $uscale = 1;$
  $Rx = gD(g, uscale, 1, 0);$
  $Ry = gD(g, uscale, 0, 1);$
  $Rw2 = Rx.^2 + Ry.^2;$
  $Rw = sqrt(Rw2);$
  $c2 = C(Rw2);$
  $c1 = 1/5 * c2;$
  $dt_a = (c1. * Rx.^2 + c2. * Ry.^2)./(Rw2 + eps);$
  $dt_b = (c2 - c1). * Rx. * Ry./(Rw2 + eps);$
  $dt_c = (c1. * Ry.^2 + c2. * Rx.^2)./(Rw2 + eps);$ "
- Also, the line: "$g = g + stepsize * snldStep(g, c, w, ip);$" is replaced by "$g = g + stepsize * tnldStep(g, dt_a, dt_b, dt_c, ip);$"

Figure: Original noisy image vs Edge enhancing diffused image



Reference(s):

- Algorithms for Non-Linear Diffusion
  Matlab in a Literate programming Style
  Rein van den Boomgaard
  Intelligent Sensory Information Systems
  University of Amsterdam
  The Netherlands