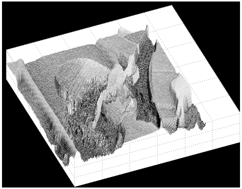
# What is an image ?

- Digital images are sampled 2-D analogue signals
- Black and white images $\equiv f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$
- $f(x) \equiv$ intensity level at that point, which varies from zero to 255
- An image can be postulated as an $L^2(\Omega)$ object

(a)

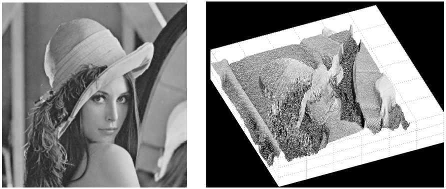


(b)

**Figure:** (a) Image of Lenna and (b) Image of Lenna as a graph of a function

# What is an image ?

- Digital images are sampled 2-D analogue signals
- Black and white images $\equiv f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$
- $f(x) \equiv$ intensity level at that point, which varies from zero to 255
- An image can be postulated as an $L^2(\Omega)$ object

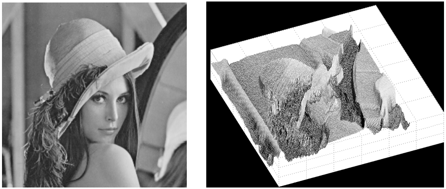


(a) (b)

**Figure:** (a) Image of Lenna and (b) Image of Lenna as a graph of a function

# What is an image ?

- Digital images are sampled 2-D analogue signals
- Black and white images $\equiv f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$
- $f(x) \equiv$ intensity level at that point, which varies from zero to 255
- An image can be postulated as an $L^2(\Omega)$ object

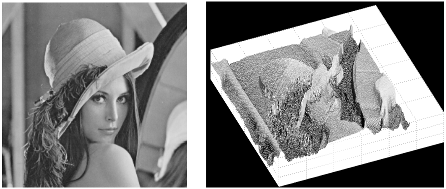


(a) (b)

**Figure:** (a) Image of Lenna and (b) Image of Lenna as a graph of a function

# What is an image ?

- Digital images are sampled 2-D analogue signals
- Black and white images $\equiv f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$
- $f(x) \equiv$ intensity level at that point, which varies from zero to 255
- An image can be postulated as an $L^2(\Omega)$ object



(a) (b)

**Figure:** (a) Image of Lenna and (b) Image of Lenna as a graph of a function

- **Image deblurring**
- $f = TU$ for a deblurring operator $T : L^2(\Omega) \to L^2(\Omega)$
  $T$ may not be invertible : ill-posed problem.
- Given $f$ we need to get back the deblurred image $U$.

(a) (b)

**Figure:** Can we go from a blurred image (a) to a restored image in (b) **?**

# Problems in image processing

- **Image deblurring**
- $f = TU$ for a deblurring operator $T : L^2(\Omega) \to L^2(\Omega)$
  $T$ may not be invertible : ill-posed problem.
- Given $f$ we need to get back the deblurred image $U$.

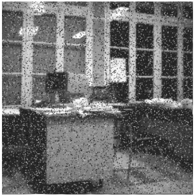
(a)

(b)

**Figure:** Can we go from a blurred image (a) to a restored image in (b) **?**

# Problems in image processing...

- **Image denoising**: $f$ may have some noise $\eta$ in it.
- $f = U + \eta$, we need to get back the denoised image $U$.



(a)

(b)

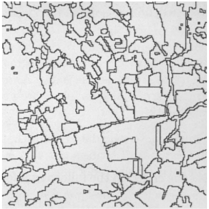
**Figure:** Can we go from a noisy image (a) to a restored image in (b) **?**

- $f$ may be blurry and noisy $f = TU + \eta$

# Problems in image processing...

- **Image segmentation** $\equiv$ identifying 'components' in $f$ $\equiv$ edge detection

(a)

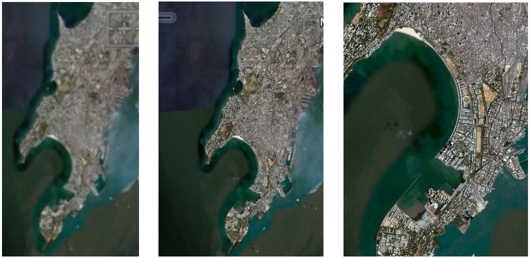


(b)

**Figure:** Can we identify components in (a) and get a segmented image as in (b) **?**

# Problems in image processing...

- **Multiscale image representation**: Finding different level of ‘scales’ in $f$



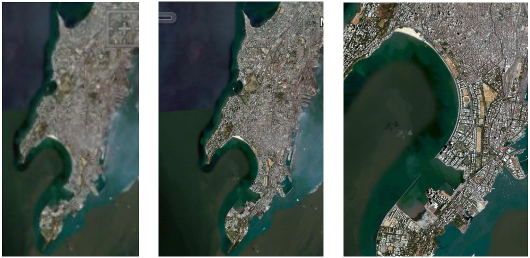
(a) (b) (c)

**Figure:** Multiscale images of the city of Mumbai.

- Multiscale representation: Family of images $\{U(t)\}$ for a scaling parameter $t$
- **Forward marching**: $U(0) = 0, U(t) \to U$
- **Backward marching**: $U(0) = f, U(t) \to U$

# Problems in image processing...

- **Multiscale image representation**: Finding different level of ‘scales’ in $f$



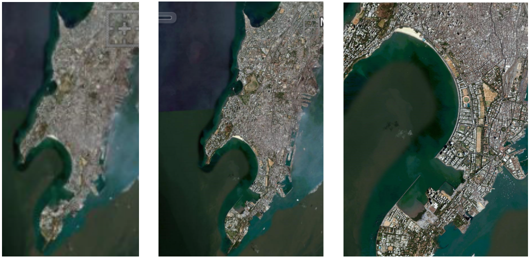
(a) (b) (c)

**Figure:** Multiscale images of the city of Mumbai.

- Multiscale representation: Family of images $\{U(t)\}$ for a scaling parameter $t$
- **Forward marching**: $U(0) = 0, U(t) \to U$
- **Backward marching**: $U(0) = f, U(t) \to U$

# Problems in image processing...

- **Multiscale image representation**: Finding different level of 'scales' in $f$



(a) (b) (c)

**Figure:** Multiscale images of the city of Mumbai.

- Multiscale representation: Family of images $\{U(t)\}$ for a scaling parameter $t$
- **Forward marching**: $U(0) = 0, U(t) \to U$
- **Backward marching**: $U(0) = f, U(t) \to U$

There are two main approaches to solve above problems:

- **Variational approaches** - Tikhonov regularization, greedy algorithms, wavelets shrinkage etc.
- **PDE based approaches** - diffusion, Perona-Malik etc.

The approaches are related -

- We need to solve the ill posed problem $f = Tu$ :

  Consider interpolation functional

  $$\inf_{u \in X} \left( \|u_\lambda\|_X + \lambda \|f - Tu_\lambda\|_Y^2 \right)$$

  $X \subset Y$

  $\|u\|_X$ : regularizing term

  $\|f - Tu\|_Y^2$ : fidelity term

- $(X, Y) \equiv (BV, L^2)$: Rudin-Osher-Fatemi-Vese.

  $$\inf_{\{f = u_\lambda + v_\lambda\}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - Tu_\lambda|^2 \right)$$

# Variational methods in image processing: Tikhonov regularization

- We need to solve the ill posed problem $f = Tu$ :

  Consider interpolation functional

  $$\inf_{u \in X} \left( \|u_\lambda\|_X + \lambda \|f - Tu_\lambda\|_Y^2 \right)$$

  $X \subset Y$

  $\|u\|_X$ : regularizing term

  $\|f - Tu\|_Y^2$ : fidelity term

- $(X, Y) \equiv (BV, L^2)$: Rudin-Osher-Fatemi-Vese.

  $$\inf_{\{f = u_\lambda + v_\lambda\}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - Tu_\lambda|^2 \right)$$

# Variational methods in image processing

- **Rudin-Osher-Fatemi (ROF) decomposition**
  $f = u_\lambda + v_\lambda$ for scale parameter $\lambda$.

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\text{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - u_\lambda|^2 \right)$$

- The BV norm $\int_\Omega |\nabla u_\lambda|$ is a regularizing term
- $\int_\Omega |f - u_\lambda|^2$: a fidelity term
- $\lambda$ : acts as an **inverse scale** of the $u_\lambda$ part ( smaller $\lambda \equiv$ larger scale )

- $u_\lambda :=$ smooth parts and edges in $f$
  $v_\lambda := f - u_\lambda$ texture, finer details, noise
- Many other variational methods ...

- **Rudin-Osher-Fatemi (ROF) decomposition**
  $f = u_\lambda + v_\lambda$ for scale parameter $\lambda$.

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\text{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - u_\lambda|^2 \right)$$

- The BV norm $\int_\Omega |\nabla u_\lambda|$ is a regularizing term
- $\int_\Omega |f - u_\lambda|^2$: a fidelity term
- $\lambda$ : acts as an **inverse scale** of the $u_\lambda$ part ( smaller $\lambda \equiv$ larger scale )

- $u_\lambda :=$ smooth parts and edges in $f$
  $v_\lambda := f - u_\lambda$ texture, finer details, noise
- Many other variational methods ...

- **Rudin-Osher-Fatemi (ROF) decomposition**
  $f = u_\lambda + v_\lambda$ for scale parameter $\lambda$.

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\text{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - u_\lambda|^2 \right)$$

- The BV norm $\int_\Omega |\nabla u_\lambda|$ is a regularizing term
- $\int_\Omega |f - u_\lambda|^2$: a fidelity term
- $\lambda$ : acts as an **inverse scale** of the $u_\lambda$ part ( smaller $\lambda \equiv$ larger scale )

- $u_\lambda :=$ smooth parts and edges in $f$
  $v_\lambda := f - u_\lambda$ texture, finer details, noise
- Many other variational methods ...

# Variational methods in image processing

- **Rudin-Osher-Fatemi (ROF) decomposition**
  $f = u_\lambda + v_\lambda$ for scale parameter $\lambda$.

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - u_\lambda|^2 \right)$$

- The BV norm $\int_\Omega |\nabla u_\lambda|$ is a regularizing term
- $\int_\Omega |f - u_\lambda|^2$: a fidelity term
- $\lambda$ : acts as an **inverse scale** of the $u_\lambda$ part ( smaller $\lambda \equiv$ larger scale )

- $u_\lambda :=$ smooth parts and edges in $f$
  $v_\lambda := f - u_\lambda$ texture, finer details, noise
- Many other variational methods ...

# Other variational methods in image processing...

- **Mumford-Shah segmentation (1985)**

$$[u, v, \mathcal{C}] = \underset{\{f=u+v,\mathcal{C}\}}{\operatorname{arginf}} \left( \int_{\Omega-\mathcal{C}} |f-u|^2 + \lambda_1 \int_{\Omega-\mathcal{C}} |\nabla u|^2 + \lambda_2 \oint_{\mathcal{C}} d\sigma \right).$$

  $u : \Omega \to \mathbb{R}$ : piecewise smooth image
  $\mathcal{C} \in \Omega$ : the set of jump discontinuities

- **Ambrosio and Tortorelli approximation (1992)**
- **Kass-Witkin-Terzopoulos model (1988)**

$$\inf_{c \in \mathcal{C}} \left( \int_a^b |c'|^2 + \lambda_1 \int_a^b |c''|^2 + \lambda_2 \int_a^b g^2(|\nabla f(c)|) \right)$$

  $\mathcal{C}$ : closed, piecewise regular, parametric curves (snakes)
  $g$ : a decreasing function vanishing at infinity

- **Caselles, Kimmel, Sapiro: Geodesic active contours (1997)**
- **Osher, Sethian: Level set method (1988)**
- ... ...
- Now we look at some PDE methods ...

# Other variational methods in image processing...

- **Mumford-Shah segmentation (1985)**

$$[u, v, \mathcal{C}] = \underset{\{f=u+v,\mathcal{C}\}}{\operatorname{arginf}} \left( \int_{\Omega-\mathcal{C}} |f-u|^2 + \lambda_1 \int_{\Omega-\mathcal{C}} |\nabla u|^2 + \lambda_2 \oint_{\mathcal{C}} d\sigma \right).$$

$u : \Omega \to \mathbb{R}$ : piecewise smooth image
$\mathcal{C} \in \Omega$ : the set of jump discontinuities

- **Ambrosio and Tortorelli approximation (1992)**
- **Kass-Witkin-Terzopoulos model (1988)**

$$\inf_{c \in \mathcal{C}} \left( \int_a^b |c'|^2 + \lambda_1 \int_a^b |c''|^2 + \lambda_2 \int_a^b g^2(|\nabla f(c)|) \right)$$

$\mathcal{C}$ : closed, piecewise regular, parametric curves (snakes)
$g$ : a decreasing function vanishing at infinity

- **Caselles, Kimmel, Sapiro: Geodesic active contours (1997)**
- **Osher, Sethian: Level set method (1988)**
- ... ...
- Now we look at some PDE methods ...

# Other variational methods in image processing...

- **Mumford-Shah segmentation (1985)**

$$[u, v, \mathcal{C}] = \underset{\{f=u+v,\mathcal{C}\}}{\operatorname{arginf}} \left( \int_{\Omega-\mathcal{C}} |f-u|^2 + \lambda_1 \int_{\Omega-\mathcal{C}} |\nabla u|^2 + \lambda_2 \oint_{\mathcal{C}} d\sigma \right).$$

$u : \Omega \to \mathbb{R}$ : piecewise smooth image
$\mathcal{C} \in \Omega$ : the set of jump discontinuities

- **Ambrosio and Tortorelli approximation (1992)**
- **Kass-Witkin-Terzopoulos model (1988)**

$$\inf_{c\in\mathcal{C}} \left( \int_a^b |c'|^2 + \lambda_1 \int_a^b |c''|^2 + \lambda_2 \int_a^b g^2(|\nabla f(c)|) \right)$$

$\mathcal{C}$ : closed, piecewise regular, parametric curves (snakes)
$g$ : a decreasing function vanishing at infinity

- **Caselles, Kimmel, Sapiro: Geodesic active contours (1997)**
- **Osher, Sethian: Level set method (1988)**
- ... ...
- Now we look at some PDE methods ...

# Other variational methods in image processing...

- **Mumford-Shah segmentation (1985)**

$$[u, v, \mathcal{C}] = \underset{\{f=u+v,\mathcal{C}\}}{\text{arginf}} \left( \int_{\Omega-\mathcal{C}} |f-u|^2 + \lambda_1 \int_{\Omega-\mathcal{C}} |\nabla u|^2 + \lambda_2 \oint_{\mathcal{C}} d\sigma \right).$$

  $u : \Omega \to \mathbb{R}$ : piecewise smooth image
  $\mathcal{C} \in \Omega$ : the set of jump discontinuities
- **Ambrosio and Tortorelli approximation (1992)**
- **Kass-Witkin-Terzopoulos model (1988)**

$$\inf_{c \in \mathcal{C}} \left( \int_a^b |c'|^2 + \lambda_1 \int_a^b |c''|^2 + \lambda_2 \int_a^b g^2(|\nabla f(c)|) \right)$$

  $\mathcal{C}$ : closed, piecewise regular, parametric curves (snakes)
  $g$ : a decreasing function vanishing at infinity
- **Caselles, Kimmel, Sapiro: Geodesic active contours (1997)**
- **Osher, Sethian: Level set method (1988)**
- ... ...
- Now we look at some PDE methods ...

# PDE methods in image processing: Heat equation

- **Heat equation** $\frac{\partial U}{\partial t} = \Delta U$. (Koenderink 1984, Witkin 1983)
- **Backward marching**: $U(0) = f$ and $U(t) \to U$.
- This gives us **Gaussian smoothing** with variance=$t$
- **Forward scaling**: Anything with scale smaller than $t$ is smoothed out
  **Thus $t$ acts as the scale parameter**

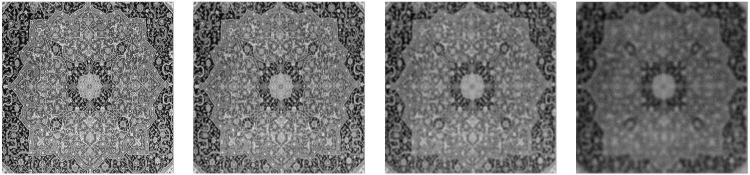


**Figure:** Different scales in a carpet obtained by heat equation

# PDE methods in image processing: Heat equation

- **Heat equation** $\frac{\partial U}{\partial t} = \Delta U$. (Koenderink 1984, Witkin 1983)
- **Backward marching**: $U(0) = f$ and $U(t) \to U$.
- This gives us **Gaussian smoothing** with variance=$t$
- **Forward scaling**: Anything with scale smaller than $t$ is smoothed out
  **Thus $t$ acts as the scale parameter**

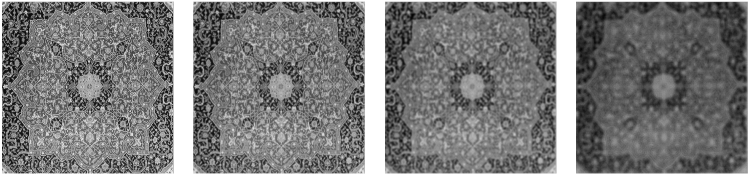


**Figure:** Different scales in a carpet obtained by heat equation

# PDE methods in image processing: Heat equation

- **Heat equation** $\frac{\partial U}{\partial t} = \Delta U$. (Koenderink 1984, Witkin 1983)
- **Backward marching**: $U(0) = f$ and $U(t) \to U$.
- This gives us **Gaussian smoothing** with variance=$t$
- **Forward scaling**: Anything with scale smaller than $t$ is smoothed out
  **Thus $t$ acts as the scale parameter**

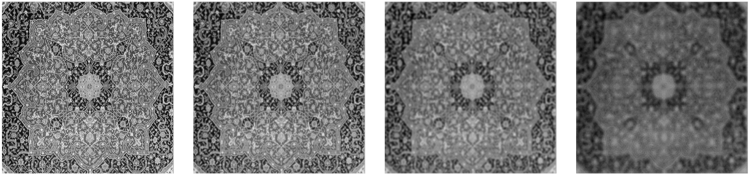
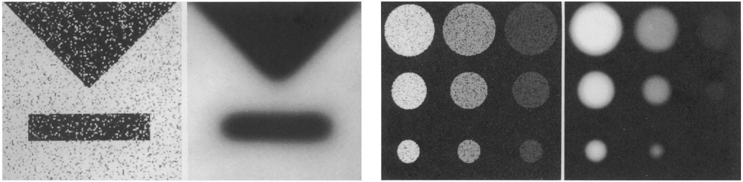


**Figure:** Different scales in a carpet obtained by heat equation

- Denoising with heat equation:

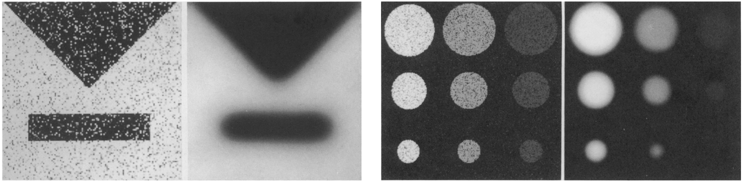


(a) (b)

**Figure:** Result of isotropic diffusion: reduction of noise at the expense of losing information at the edges

- **Problem 1**: cannot distinguish between noise and boundaries of regions
- **Problem 2**: where to stop ?

- Denoising with heat equation:



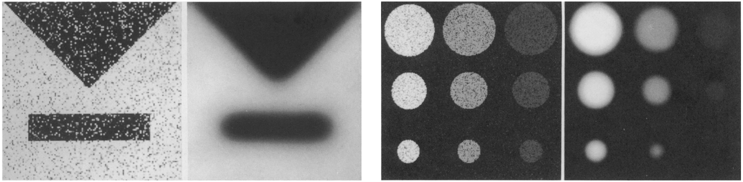
(a) (b)

**Figure:** Result of isotropic diffusion: reduction of noise at the expense of losing information at the edges

- **Problem 1**: cannot distinguish between noise and boundaries of regions
- **Problem 2**: where to stop ?

# PDE methods in image processing: Heat equation...

- Denoising with heat equation:



(a) (b)

**Figure:** Result of isotropic diffusion: reduction of noise at the expense of losing information at the edges

- **Problem 1**: cannot distinguish between noise and boundaries of regions
- **Problem 2**: where to stop ?

# PDE methods in image processing: Perona-Malik model (1988)

- Heat equation $\equiv$ **isotropic diffusion** $\Rightarrow$ we lose information about edges
- Perona-Malik proposed an **anisotropic diffusion method**

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = f$$

- **The idea: preserve the edges**
  - Smooth regions $\equiv |\nabla U|$ is weak $\Rightarrow$ we need an isotropic smoothing
  - Near the edges $\equiv |\nabla U|$ is large $\Rightarrow$ we need to control the diffusion
  - Examples of suitable function $g(s)$ : $e^{-s}$, $\frac{1}{1+s^2}$, $\frac{1}{\sqrt{1+s}}$
- Perona-Malik is not well posed ! Catté et.al. modification[2] :

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla G \star U|)\nabla U\right),$$

$G$ is Gaussian kernel.

[2] F. Catté, P-L. Lions, J-M. Morel, T. Coll (1992)

# PDE methods in image processing: Perona-Malik model (1988)

- Heat equation $\equiv$ **isotropic diffusion** $\Rightarrow$ we lose information about edges
- Perona-Malik proposed an **anisotropic diffusion method**

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = f$$

- **The idea: preserve the edges**

  Smooth regions $\equiv |\nabla U|$ is weak $\Rightarrow$ we need an isotropic smoothing

  Near the edges $\equiv |\nabla U|$ is large $\Rightarrow$ we need to control the diffusion

  Examples of suitable function $g(s)$ : $e^{-s}$, $\frac{1}{1+s^2}$, $\frac{1}{\sqrt{1+s}}$

- Perona-Malik is not well posed ! Catté et.al. modification[2] :

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla G \star U|)\nabla U\right),$$

$G$ is Gaussian kernel.

[2] F. Catté, P-L. Lions, J-M. Morel, T. Coll (1992)

# PDE methods in image processing: Perona-Malik model (1988)

- Heat equation $\equiv$ **isotropic diffusion** $\Rightarrow$ we lose information about edges
- Perona-Malik proposed an **anisotropic diffusion method**

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = f$$

- **The idea: preserve the edges**

  Smooth regions $\equiv |\nabla U|$ is weak $\Rightarrow$ we need an isotropic smoothing
  Near the edges $\equiv |\nabla U|$ is large $\Rightarrow$ we need to control the diffusion
  Examples of suitable function $g(s)$ : $e^{-s}$, $\frac{1}{1+s^2}$, $\frac{1}{\sqrt{1+s}}$

- Perona-Malik is not well posed ! Catté et.al. modification[2] :

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla G \star U|)\nabla U\right),$$

$G$ is Gaussian kernel.

[2] F. Catté, P-L. Lions, J-M. Morel, T. Coll (1992)

# PDE methods in image processing: Perona-Malik model (1988)

- Heat equation $\equiv$ **isotropic diffusion** $\Rightarrow$ we lose information about edges
- Perona-Malik proposed an **anisotropic diffusion method**

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = f$$

- **The idea: preserve the edges**

  Smooth regions $\equiv |\nabla U|$ is weak $\Rightarrow$ we need an isotropic smoothing
  Near the edges $\equiv |\nabla U|$ is large $\Rightarrow$ we need to control the diffusion
  Examples of suitable function $g(s)$ : $e^{-s}$, $\frac{1}{1+s^2}$, $\frac{1}{\sqrt{1+s}}$

- Perona-Malik is not well posed ! Catté et.al. modification[2] :

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla G \star U|)\nabla U\right),$$

$G$ is Gaussian kernel.

[2] F. Catté, P-L. Lions, J-M. Morel, T. Coll (1992)

- Heat equation $\equiv$ **isotropic diffusion** $\Rightarrow$ we lose information about edges
- Perona-Malik proposed an **anisotropic diffusion method**

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = f$$

- **The idea: preserve the edges**
  Smooth regions $\equiv |\nabla U|$ is weak $\Rightarrow$ we need an isotropic smoothing
  Near the edges $\equiv |\nabla U|$ is large $\Rightarrow$ we need to control the diffusion
  Examples of suitable function $g(s)$ : $e^{-s}$, $\frac{1}{1+s^2}$, $\frac{1}{\sqrt{1+s}}$
- Perona-Malik is not well posed ! Catté et.al. modification[2] :

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla G \star U|)\nabla U\right),$$

$G$ is Gaussian kernel.

[2] F. Catté, P-L. Lions, J-M. Morel, T. Coll (1992)

# PDE methods in image processing: Perona-Malik model (1988)

- Heat equation $\equiv$ **isotropic diffusion** $\Rightarrow$ we lose information about edges
- Perona-Malik proposed an **anisotropic diffusion method**

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = f$$

- **The idea: preserve the edges**
  Smooth regions $\equiv |\nabla U|$ is weak $\Rightarrow$ we need an isotropic smoothing
  Near the edges $\equiv |\nabla U|$ is large $\Rightarrow$ we need to control the diffusion
  Examples of suitable function $g(s)$ : $e^{-s}$, $\frac{1}{1+s^2}$, $\frac{1}{\sqrt{1+s}}$
- Perona-Malik is not well posed ! Catté et.al. modification[2] :

$$\frac{\partial U}{\partial t} = \operatorname{div}\left(g(|\nabla G \star U|)\nabla U\right),$$

$G$ is Gaussian kernel.

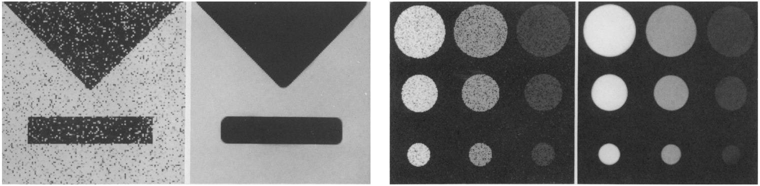
[2] F. Catté, P-L. Lions, J-M. Morel, T. Coll (1992)

# PDE methods in image processing: Alvarez et. al.

- **L. Alvarez P-L. Lions and J-M Morel's model (1992)**

$$\frac{\partial U}{\partial t} = g(|G \star \nabla U|)|\nabla U| \operatorname{div}\left(\frac{\nabla U}{|\nabla U|}\right), \quad U(0) = f$$

- **Idea**: Diffuse $U$ only in the direction orthogonal to its gradient $\nabla U$.
- The term $|\nabla U| \operatorname{div}\left(\frac{\nabla U}{|\nabla U|}\right)$ does exactly this.
- $g$ is a diffusion controlling function as before.



(a) (b)

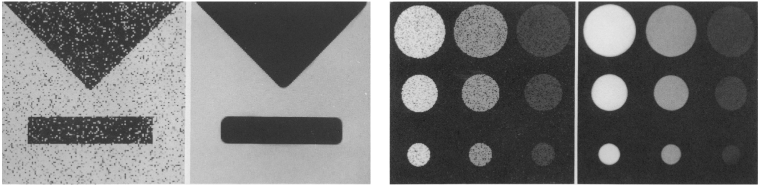
**Figure:** Result of anisotropic diffusion: edges are preserved.

# PDE methods in image processing: Alvarez et. al.

- **L. Alvarez P-L. Lions and J-M Morel's model (1992)**

$$\frac{\partial U}{\partial t} = g(|G \star \nabla U|)|\nabla U| \operatorname{div}\left(\frac{\nabla U}{|\nabla U|}\right), \quad U(0) = f$$

- **Idea**: Diffuse $U$ only in the direction orthogonal to its gradient $\nabla U$.
- The term $|\nabla U| \operatorname{div}\left(\frac{\nabla U}{|\nabla U|}\right)$ does exactly this.
- $g$ is a diffusion controlling function as before.



(a) (b)

**Figure:** Result of anisotropic diffusion: edges are preserved.

# PDE methods in image processing: Alvarez et. al.

- **L. Alvarez P-L. Lions and J-M Morel's model (1992)**

$$\frac{\partial U}{\partial t} = g(|G \star \nabla U|)|\nabla U| \operatorname{div}\left(\frac{\nabla U}{|\nabla U|}\right), \quad U(0) = f$$

- **Idea**: Diffuse $U$ only in the direction orthogonal to its gradient $\nabla U$.
- The term $|\nabla U| \operatorname{div}\left(\frac{\nabla U}{|\nabla U|}\right)$ does exactly this.
- $g$ is a diffusion controlling function as before.

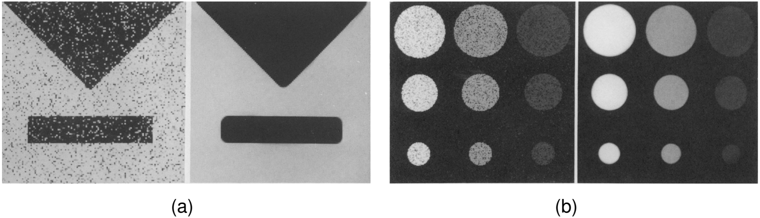



**Figure:** Result of anisotropic diffusion: edges are preserved.

- **Problem**: As $t \to \infty$ the models discussed before diffuse completely.
  **... so where to stop ?**
- **Solution**: **Nordström** modified Perona-Malik model.

$$\frac{\partial U}{\partial t} = f - U + \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \qquad U(0) = 0.$$

- This equation has **non-trivial steady state**.
- **Forward marching**: $U(0) = 0$ and $U(t) \to U$.

- **Problem**: As $t \to \infty$ the models discussed before diffuse completely.
  ... **so where to stop ?**
- **Solution**: **Nordström** modified Perona-Malik model.

$$\frac{\partial U}{\partial t} = f - U + \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = 0.$$

- This equation has **non-trivial steady state**.
- **Forward marching**: $U(0) = 0$ and $U(t) \to U$.

- **Problem**: As $t \to \infty$ the models discussed before diffuse completely.
  ... **so where to stop ?**
- **Solution**: **Nordström** modified Perona-Malik model.

$$\frac{\partial U}{\partial t} = f - U + \operatorname{div}\left(g(|\nabla U|)\nabla U\right), \quad U(0) = 0.$$

- This equation has **non-trivial steady state**.
- **Forward marching**: $U(0) = 0$ and $U(t) \to U$.

**Rudin-Osher-Fatemi decomposition**

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - u_\lambda|^2 \right)$$

The Euler-Lagrange equation:

$$f - u + \frac{1}{2\lambda} \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) = 0.$$

**Nordström's modification of Perona-Malik:**

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

$g(s) = \frac{1}{\lambda s} \Rightarrow$ steady-state of Nordström $\equiv$ Euler-Lagrange of ROF !

Let us look at our model now ...

# PDE approach $\leftrightsquigarrow$ variational approach

**Rudin-Osher-Fatemi decomposition**

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - u_\lambda|^2 \right)$$

The Euler-Lagrange equation:

$$f - u + \frac{1}{2\lambda} \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) = 0.$$

**Nordström's modification of Perona-Malik**:

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

$g(s) = \frac{1}{\lambda s} \Rightarrow$ steady-state of Nordström $\equiv$ Euler-Lagrange of ROF !

Let us look at our model now ...

# PDE approach $\leftrightsquigarrow$ variational approach

**Rudin-Osher-Fatemi decomposition**

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\text{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f-u_\lambda|^2 \right)$$

The Euler-Lagrange equation:

$$f - u + \frac{1}{2\lambda} \operatorname{div}\left( \frac{\nabla u}{|\nabla u|} \right) = 0.$$

**Nordström's modification of Perona-Malik**:

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

$g(s) = \frac{1}{\lambda s} \Rightarrow$ steady-state of Nordström $\equiv$ Euler-Lagrange of ROF !

Let us look at our model now ...

**Rudin-Osher-Fatemi decomposition**

$$[u_\lambda, v_\lambda] = \underset{\{f=u_\lambda+v_\lambda\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_\lambda| + \lambda \int_\Omega |f - u_\lambda|^2 \right)$$

The Euler-Lagrange equation:

$$f - u + \frac{1}{2\lambda} \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) = 0.$$

**Nordström's modification of Perona-Malik**:

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

$g(s) = \frac{1}{\lambda s} \Rightarrow$ steady-state of Nordström $\equiv$ Euler-Lagrange of ROF !

Let us look at our model now ...

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆⋆⋆ QUESTIONS ⋆⋆⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

# A novel integro-differential model

- We propose a novel model.

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

- An **Integro**-differential equation.
- The scaling function $\lambda(t)$ : increasing function at our disposal.
- This model gives an inverse scale representation.
- Compare this with Nordström's model:

$$U(t) + \frac{\partial U(t)}{\partial t} = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla U(t)}{|\nabla U(t)|}\right).$$

⋆ ⋆ ⋆ **QUESTIONS** ⋆ ⋆ ⋆

What is the **motivation ?**

Where to **start ?**

Where to **stop ?**

What does the scaling function $\lambda(t)$ mean **?**

⋆

## Starting point: the idea of Tadmor-Nezzar-Vese (TNV) 2004, 2008

- **Recall ROF decomposition**: $f = u_{\lambda_0} + v_{\lambda_0}$, where $\lambda_0$ dictates the scale.
- $v_{\lambda_0}$ can be decomposed with a scaling parameter $\lambda_1 > \lambda_0$.

$$v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}, \quad [u_{\lambda_1}, v_{\lambda_1}] = \underset{\{v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_1}| + \lambda_1 \int_\Omega |v_{\lambda_0} - u_{\lambda_1}|^2 \right).$$

- **TNV multiscale decomposition**

$$v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}, \quad [u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{k-1} - u_k|^2 \right)$$

- With this scheme after $N + 1$ steps we get:

$$\begin{aligned} f &= u_{\lambda_0} + v_{\lambda_0} \\ &= u_{\lambda_0} + u_{\lambda_1} + v_{\lambda_1} \\ &= \dots \\ &= u_{\lambda_0} + u_{\lambda_1} + \dots + u_{\lambda_N} + v_{\lambda_N}. \end{aligned}$$

i.e. a nonlinear multiscale decomposition: $f \sim \sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N}$.

# Starting point: the idea of Tadmor-Nezzar-Vese (TNV) 2004, 2008

- **Recall ROF decomposition**: $f = u_{\lambda_0} + v_{\lambda_0}$, where $\lambda_0$ dictates the scale.
- $v_{\lambda_0}$ can be decomposed with a scaling parameter $\lambda_1 > \lambda_0$.

$$v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}, \quad [u_{\lambda_1}, v_{\lambda_1}] = \underset{\{v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_1}| + \lambda_1 \int_\Omega |v_{\lambda_0} - u_{\lambda_1}|^2 \right).$$

- **TNV multiscale decomposition**

$$v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}, \quad [u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{k-1} - u_k|^2 \right)$$

- With this scheme after $N+1$ steps we get:

$$\begin{aligned} f &= u_{\lambda_0} + v_{\lambda_0} \\ &= u_{\lambda_0} + u_{\lambda_1} + v_{\lambda_1} \\ &= \dots \\ &= u_{\lambda_0} + u_{\lambda_1} + \dots + u_{\lambda_N} + v_{\lambda_N}. \end{aligned}$$

i.e. a nonlinear multiscale decomposition: $f \sim \sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N}$.

# Starting point: the idea of Tadmor-Nezzar-Vese (TNV) 2004, 2008

- **Recall ROF decomposition**: $f = u_{\lambda_0} + v_{\lambda_0}$, where $\lambda_0$ dictates the scale.
- $v_{\lambda_0}$ can be decomposed with a scaling parameter $\lambda_1 > \lambda_0$.

$$v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}, \quad [u_{\lambda_1}, v_{\lambda_1}] = \underset{\{v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_1}| + \lambda_1 \int_\Omega |v_{\lambda_0} - u_{\lambda_1}|^2 \right).$$

- **TNV multiscale decomposition**

$$v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}, \quad [u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{k-1} - u_k|^2 \right)$$

- With this scheme after $N+1$ steps we get:

$$\begin{aligned} f &= u_{\lambda_0} + v_{\lambda_0} \\ &= u_{\lambda_0} + u_{\lambda_1} + v_{\lambda_1} \\ &= \ldots \\ &= u_{\lambda_0} + u_{\lambda_1} + \ldots + u_{\lambda_N} + v_{\lambda_N}. \end{aligned}$$

i.e. a nonlinear multiscale decomposition: $f \sim \sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N}$.

# Starting point: the idea of Tadmor-Nezzar-Vese (TNV) 2004, 2008

- **Recall ROF decomposition**: $f = u_{\lambda_0} + v_{\lambda_0}$, where $\lambda_0$ dictates the scale.
- $v_{\lambda_0}$ can be decomposed with a scaling parameter $\lambda_1 > \lambda_0$.

$$v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}, \quad [u_{\lambda_1}, v_{\lambda_1}] = \underset{\{v_{\lambda_0} = u_{\lambda_1} + v_{\lambda_1}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_1}| + \lambda_1 \int_\Omega |v_{\lambda_0} - u_{\lambda_1}|^2 \right).$$

- **TNV multiscale decomposition**

$$v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}, \quad \left[u_{\lambda_k}, v_{\lambda_k}\right] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{k-1} - u_k|^2 \right)$$

- With this scheme after $N + 1$ steps we get:

$$\begin{aligned} f &= u_{\lambda_0} + v_{\lambda_0} \\ &= u_{\lambda_0} + u_{\lambda_1} + v_{\lambda_1} \\ &= ... \\ &= u_{\lambda_0} + u_{\lambda_1} + ... + u_{\lambda_N} + v_{\lambda_N}. \end{aligned}$$

i.e. a nonlinear multiscale decomposition: $f \sim \sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N}$.

# TNV scheme

- $k^{th}$ step in TNV scheme: $u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - u_{\lambda_k}|^2 \right)$$

$$u_{\lambda_k} \underbrace{- \frac{1}{2\lambda_k} \text{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)}_{v_{\lambda_k}} = v_{\lambda_{k-1}}$$

- TNV iteration:

$$u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$$

Telescopic sum of the above gives us:

$$\sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N} = f$$

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \text{div} \left( \frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|} \right)$$

# TNV scheme

- $k^{th}$ step in TNV scheme: $u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - u_{\lambda_k}|^2 \right)$$

$$u_{\lambda_k} \underbrace{- \frac{1}{2\lambda_k} \operatorname{div}\left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)}_{v_{\lambda_k}} = v_{\lambda_{k-1}}$$

- TNV iteration:

$$u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$$

Telescopic sum of the above gives us:

$$\sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N} = f$$

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left( \frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|} \right)$$

# TNV scheme

- $k^{th}$ step in TNV scheme: $u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - u_{\lambda_k}|^2 \right)$$

$$u_{\lambda_k} \underbrace{- \frac{1}{2\lambda_k} \operatorname{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)}_{v_{\lambda_k}} = v_{\lambda_{k-1}}$$

- TNV iteration:

$$u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$$

Telescopic sum of the above gives us:

$$\sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N} = f$$

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div} \left( \frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|} \right)$$

# TNV scheme

- $k^{th}$ step in TNV scheme: $u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - u_{\lambda_k}|^2 \right)$$

$$u_{\lambda_k} \underbrace{- \frac{1}{2\lambda_k} \operatorname{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)}_{v_{\lambda_k}} = v_{\lambda_{k-1}}$$

- TNV iteration:

$$u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$$

Telescopic sum of the above gives us:

$$\sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N} = f$$

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div} \left( \frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|} \right)$$

# TNV scheme

- $k^{th}$ step in TNV scheme: $u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - u_{\lambda_k}|^2 \right)$$

$$u_{\lambda_k} \underbrace{- \frac{1}{2\lambda_k} \operatorname{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)}_{v_{\lambda_k}} = v_{\lambda_{k-1}}$$

- TNV iteration:

$$u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$$

Telescopic sum of the above gives us:

$$\sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N} = f$$

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div} \left( \frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|} \right)$$

# TNV scheme

- $k^{th}$ step in TNV scheme: $u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - u_{\lambda_k}|^2 \right)$$

$$u_{\lambda_k} \underbrace{- \frac{1}{2\lambda_k} \operatorname{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)}_{v_{\lambda_k}} = v_{\lambda_{k-1}}$$

- TNV iteration:

$$u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$$

Telescopic sum of the above gives us:

$$\sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N} = f$$

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div} \left( \frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|} \right)$$

# TNV scheme

- $k^{th}$ step in TNV scheme: $u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = u_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - u_{\lambda_k}|^2 \right)$$

$$u_{\lambda_k} \underbrace{- \frac{1}{2\lambda_k} \operatorname{div}\left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)}_{v_{\lambda_k}} = v_{\lambda_{k-1}}$$

- TNV iteration:

$$u_{\lambda_k} + v_{\lambda_k} = v_{\lambda_{k-1}}$$

Telescopic sum of the above gives us:

$$\sum_{k=0}^{N} u_{\lambda_k} + v_{\lambda_N} = f$$

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left( \frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|} \right)$$

# A new formulation of the TNV scheme

- We have the TNV scheme as follows:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right).$$

- Define $U^N$ as the sum $U^N = \sum_{k=0}^{N} u_k \Rightarrow u^N = U^N - U^{N-1}$, we get :

**New formulation of TNV**

$$U^N = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla(U^N - U^{N-1})}{|\nabla(U^N - U^{N-1})|}\right)$$

- **Question**: How do we solve this numerically ?

# A new formulation of the TNV scheme

- We have the TNV scheme as follows:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right).$$

- Define $U^N$ as the sum $U^N = \sum_{k=0}^{N} u_k \Rightarrow u^N = U^N - U^{N-1}$, we get :

**New formulation of TNV**

$$U^N = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla(U^N - U^{N-1})}{|\nabla(U^N - U^{N-1})|}\right)$$

- **Question**: How do we solve this numerically ?

# A new formulation of the TNV scheme

- We have the TNV scheme as follows:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right).$$

- Define $U^N$ as the sum $U^N = \sum_{k=0}^{N} u_k \Rightarrow u^N = U^N - U^{N-1}$, we get :

**New formulation of TNV**

$$U^N = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla(U^N - U^{N-1})}{|\nabla(U^N - U^{N-1})|}\right)$$

- **Question**: How do we solve this numerically ?

# How to solve TNV numerically ?

- We need to solve this : $U^N = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla(U^N - U^{N-1})}{|\nabla(U^N - U^{N-1})|}\right)$.

$$c_E(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{+x}U_{i,j})^2 + (\Delta_{0y}U_{i,j})^2}}, \quad c_W(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{-x}U_{i,j})^2 + (\Delta_{0y}U_{i-1,j})^2}},$$

$$c_S(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{0x}U_{i,j})^2 + (\Delta_{+y}U_{i,j})^2}}, \quad c_N(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{0x}U_{i,j-1})^2 + (\Delta_{-y}U_{i,j})^2}}.$$

- Given $U^{N-1}$ we get $U^N$ by solving the following fixed point iteration.

$$U^n_{i,j} = \frac{2\lambda h f_{i,j} + c_E(U^{n-1}_{i+1,j} - U^{N-1}_{i+1,j}) + c_W(U^{n-1}_{i-1,j} - U^{N-1}_{i-1,j}) + c_S(U^{n-1}_{i,j+1} - U^{N-1}_{i,j+1}) + c_N(U^{n-1}_{i,j-1} - U^{N-1}_{i,j-1}) + (\sum c)U^{N-1}}{2\lambda h f_{i,j} + \sum c},$$

where $c_E \equiv c_E(U^{n-1} - U^{N-1})$ etc. and $\sum_c \equiv c_E + c_W + c_N + c_S$.

# How to solve TNV numerically ?

- We need to solve this : $U^N = f + \frac{1}{2\lambda_N}\,\mathrm{div}\left(\frac{\nabla(U^N - U^{N-1})}{|\nabla(U^N - U^{N-1})|}\right)$.

$$
\begin{aligned}
c_E(U) &\equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{+x}U_{i,j})^2 + (\Delta_{0y}U_{i,j})^2}},\; c_W(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{-x}U_{i,j})^2 + (\Delta_{0y}U_{i-1,j})^2}},\\
c_S(U) &\equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{0x}U_{i,j})^2 + (\Delta_{+y}U_{i,j})^2}},\; c_N(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{0x}U_{i,j-1})^2 + (\Delta_{-y}U_{i,j})^2}}.
\end{aligned}
$$

- Given $U^{N-1}$ we get $U^N$ by solving the following fixed point iteration.

$$
U^n_{i,j} = \frac{2\lambda h f_{i,j} + c_E(U^{n-1}_{i+1,j} - U^{N-1}_{i+1,j}) + c_W(U^{n-1}_{i-1,j} - U^{N-1}_{i-1,j}) + c_S(U^{n-1}_{i,j+1} - U^{N-1}_{i,j+1}) + c_N(U^{n-1}_{i,j-1} - U^{N-1}_{i,j-1}) + (\sum_c)U^{N-1}}{2\lambda h f_{i,j} + \sum_c},
$$

where $c_E \equiv c_E(U^{n-1} - U^{N-1})$ etc. and $\sum_c \equiv c_E + c_W + c_N + c_S$.

# How to solve TNV numerically ?

- We need to solve this : $U^N = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla(U^N - U^{N-1})}{|\nabla(U^N - U^{N-1})|}\right)$.

$$c_E(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{+x}U_{i,j})^2 + (\Delta_{0y}U_{i,j})^2}},\ c_W(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{-x}U_{i,j})^2 + (\Delta_{0y}U_{i-1,j})^2}},$$

$$c_S(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{0x}U_{i,j})^2 + (\Delta_{+y}U_{i,j})^2}},\ c_N(U) \equiv \frac{1}{\sqrt{(\varepsilon h)^2 + (\Delta_{0x}U_{i,j-1})^2 + (\Delta_{-y}U_{i,j})^2}}.$$

- Given $U^{N-1}$ we get $U^N$ by solving the following fixed point iteration.

$$U^n_{i,j} = \frac{2\lambda h f_{i,j} + c_E(U^{n-1}_{i+1,j} - U^{N-1}_{i+1,j}) + c_W(U^{n-1}_{i-1,j} - U^{N-1}_{i-1,j}) + c_S(U^{n-1}_{i,j+1} - U^{N-1}_{i,j+1}) + c_N(U^{n-1}_{i,j-1} - U^{N-1}_{i,j-1}) + (\sum c)U^{N-1}}{2\lambda h f_{i,j} + \sum c},$$

where $c_E \equiv c_E(U^{n-1} - U^{N-1})$ etc. and $\sum_c \equiv c_E + c_W + c_N + c_S$.

# Numerical results of standard TNV continued ...

Numerical results of $U^N = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla(U^N - U^{N-1})}{|\nabla(U^N - U^{N-1})|}\right)$.

**Figure:** Numerical results with standard TNV model. We used $\lambda_k = (0.0005) \times 2^k$.

# Going from TNV to a novel integro-differential equation

Recall for TNV formulation: $U_N = \sum_{k=0}^{N} u_k$ and $u_N = U_N - U_{N-1}$.

$$
\begin{aligned}
U_N &= f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla(U_N - U_{N-1})}{|\nabla(U_N - U_{N-1})|}\right) \\
\sum_{k=0}^{N} u_k &= f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_N}{|\nabla u_N|}\right).
\end{aligned}
$$

This motivates us to write the following model.

**The novel integro-differential model**

$$
\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).
$$

where $\lambda(t) > 0$ is an increasing scaling function at our disposal.

# Going from TNV to a novel integro-differential equation

Recall for TNV formulation: $U_N = \sum_{k=0}^{N} u_k$ and $u_N = U_N - U_{N-1}$.

$$\begin{aligned} U_N &= f + \frac{1}{2\lambda_N}\,\mathrm{div}\left(\frac{\nabla(U_N - U_{N-1})}{|\nabla(U_N - U_{N-1})|}\right) \\ \sum_{k=0}^{N} u_k &= f + \frac{1}{2\lambda_N}\,\mathrm{div}\left(\frac{\nabla u_N}{|\nabla u_N|}\right). \end{aligned}$$

This motivates us to write the following model.

**The novel integro-differential model**

$$\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)}\,\mathrm{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

where $\lambda(t) > 0$ is an increasing scaling function at our disposal.

# How to solve it numerically ?

- Let $\Delta t$ be the time interval step. Thus, after $N$ steps:

$$U(t) := \int_0^t u(x,s)ds = \sum_{k=0}^{N-1} \int_{k\Delta t}^{(k+1)\Delta t} u(x,s)ds$$

- $U^N := \int_0^{N\Delta t} u(x,s)ds$ and $u^{k+1} := u((k+1)\Delta t)$, with this we have

$$U^N \approx U^{N-1} + u^N \Delta t.$$

- Thus, we have the following fixed point iteration.

$$u_{i,j}^n = \frac{2\lambda^N h(f_{i,j} - U_{i,j}^{N-1}) + c_E u_{i+1,j}^{n-1} + c_W u_{i-1,j}^{n-1} + c_S u_{i,j+1}^{n-1} + c_N u_{i,j-1}^{n-1}}{2\lambda^N h\Delta t + c_E + c_W + c_S + c_N}.$$

- This fixed point implementation gives us $u^N$ and thus $U^N = U^{N-1} + u^N \Delta t$

# How to solve it numerically ?

- Let $\Delta t$ be the time interval step. Thus, after $N$ steps:

$$U(t) := \int_0^t u(x,s)ds = \sum_{k=0}^{N-1} \int_{k\Delta t}^{(k+1)\Delta t} u(x,s)ds$$

- $U^N := \int_0^{N\Delta t} u(x,s)ds$ and $u^{k+1} := u((k+1)\Delta t)$, with this we have

$$U^N \approx U^{N-1} + u^N \Delta t.$$

- Thus, we have the following fixed point iteration.

$$u_{i,j}^n = \frac{2\lambda^N h(f_{i,j} - U_{i,j}^{N-1}) + c_E u_{i+1,j}^{n-1} + c_W u_{i-1,j}^{n-1} + c_S u_{i,j+1}^{n-1} + c_N u_{i,j-1}^{n-1}}{2\lambda^N h\Delta t + c_E + c_W + c_S + c_N}.$$

- This fixed point implementation gives us $u^N$ and thus $U^N = U^{N-1} + u^N \Delta t$

# How to solve it numerically ?

- Let $\Delta t$ be the time interval step. Thus, after $N$ steps:

$$U(t) := \int_0^t u(x,s)ds = \sum_{k=0}^{N-1} \int_{k\Delta t}^{(k+1)\Delta t} u(x,s)ds$$

- $U^N := \int_0^{N\Delta t} u(x,s)ds$ and $u^{k+1} := u((k+1)\Delta t)$, with this we have

$$U^N \approx U^{N-1} + u^N \Delta t.$$

- Thus, we have the following fixed point iteration.

$$u_{i,j}^n = \frac{2\lambda^N h(f_{i,j} - U_{i,j}^{N-1}) + c_E u_{i+1,j}^{n-1} + c_W u_{i-1,j}^{n-1} + c_S u_{i,j+1}^{n-1} + c_N u_{i,j-1}^{n-1}}{2\lambda^N h\Delta t + c_E + c_W + c_S + c_N}.$$

- This fixed point implementation gives us $u^N$ and thus $U^N = U^{N-1} + u^N \Delta t$

# How to solve it numerically ?

- Let $\Delta t$ be the time interval step. Thus, after $N$ steps:

$$U(t) := \int_0^t u(x,s)ds = \sum_{k=0}^{N-1} \int_{k\Delta t}^{(k+1)\Delta t} u(x,s)ds$$

- $U^N := \int_0^{N\Delta t} u(x,s)ds$ and $u^{k+1} := u((k+1)\Delta t)$, with this we have

$$U^N \approx U^{N-1} + u^N \Delta t.$$

- Thus, we have the following fixed point iteration.

$$u_{i,j}^n = \frac{2\lambda^N h(f_{i,j} - U_{i,j}^{N-1}) + c_E u_{i+1,j}^{n-1} + c_W u_{i-1,j}^{n-1} + c_S u_{i,j+1}^{n-1} + c_N u_{i,j-1}^{n-1}}{2\lambda^N h\Delta t + c_E + c_W + c_S + c_N}.$$

- This fixed point implementation gives us $u^N$ and thus $U^N = U^{N-1} + u^N \Delta t$

# Proposed model $\lambda(t) = (0.02)2^t$, on $256 \times 256$ image of Lenna.

Numerical result for $\int_0^t u(x,s)ds = f(x) + \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$.

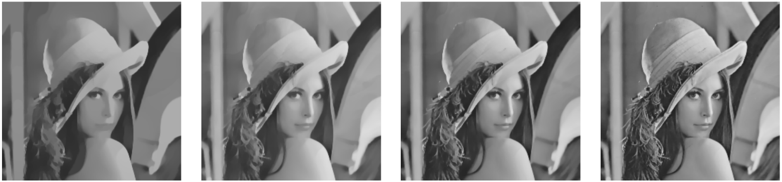


**Figure:** As $\lambda(t) \to \infty$, the image $\int_0^t u(x,s)ds$ approaches the given image $f$.

# Modified integro-differential model

- We propose a modified version of our model

## Modified integro differential model

$$\int_0^t u(x,s)ds = f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

where $g$ is diffusion controlling function. (Recall Perona-Malik.)

- **The motivation**: numerical implementation of the ROF model.
- Euler-Lagrange differential equation for ROF:

$$u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

- There are two problems here.
  - **Problem 1** : $|\nabla u| \approx 0$.
  - **Problem 2** : $|\nabla u|$ is undefined at the sharp discontinuities.

# Modified integro-differential model

- We propose a modified version of our model

## Modified integro differential model

$$\int_0^t u(x,s)ds = f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

where $g$ is diffusion controlling function. (Recall Perona-Malik.)

- **The motivation**: numerical implementation of the ROF model.
- Euler-Lagrange differential equation for ROF:

$$u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

- There are two problems here.
  - **Problem 1** : $|\nabla u| \approx 0$.
  - **Problem 2** : $|\nabla u|$ is undefined at the sharp discontinuities.

- We propose a modified version of our model

**Modified integro differential model**

$$\int_0^t u(x,s)ds = f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

where $g$ is diffusion controlling function. (Recall Perona-Malik.)

- **The motivation**: numerical implementation of the ROF model.
- Euler-Lagrange differential equation for ROF:

$$u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

- There are two problems here.
  - **Problem 1** : $|\nabla u| \approx 0$.
  - **Problem 2** : $|\nabla u|$ is undefined at the sharp discontinuities.

# Modified integro-differential model

- We propose a modified version of our model

**Modified integro differential model**

$$\int_0^t u(x,s)ds = f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

where $g$ is diffusion controlling function. (Recall Perona-Malik.)

- **The motivation**: numerical implementation of the ROF model.
- Euler-Lagrange differential equation for ROF:

$$u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

- There are two problems here.
  - **Problem 1** : $|\nabla u| \approx 0$.
  - **Problem 2** : $|\nabla u|$ is undefined at the sharp discontinuities.

# The motivation: numerical implementation of the ROF model.

- **Problem 1** : $|\nabla u| \approx 0$ : $u = f + \frac{1}{2\lambda}\,\mathrm{div}\left(\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}\right)$
- **Problem 2** : How to deal with the sharp discontinuities ?
- **Idea** : increase the cell size $h$ if $|\nabla u|$ is large: **non-uniform grid**.
- Let $\hat{h} = \frac{h}{g(|G \star \nabla u|)}$ with $g(0) = 1$ and vanishing at infinity $\Rightarrow$

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda}\,\mathrm{div}\left(\frac{\nabla u}{\sqrt{(\varepsilon \hat{h})^2 + |\nabla u|^2}}\right).$$

- This motivates us to look at :

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda}\,\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Numerical results ...

## The motivation: numerical implementation of the ROF model.

- **Problem 1** : $|\nabla u| \approx 0$ : $u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}\right)$
- **Problem 2** : How to deal with the sharp discontinuities ?
- **Idea** : increase the cell size $h$ if $|\nabla u|$ is large: **non-uniform grid**.
- Let $\hat{h} = \frac{h}{g(|G \star \nabla u|)}$ with $g(0) = 1$ and vanishing at infinity $\Rightarrow$

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{(\varepsilon \hat{h})^2 + |\nabla u|^2}}\right).$$

- This motivates us to look at :

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Numerical results ...

# The motivation: numerical implementation of the ROF model.

- **Problem 1** : $|\nabla u| \approx 0$ : $u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}\right)$
- **Problem 2** : How to deal with the sharp discontinuities ?
- **Idea** : increase the cell size $h$ if $|\nabla u|$ is large: **non-uniform grid**.
- Let $\hat{h} = \frac{h}{g(|G \star \nabla u|)}$ with $g(0) = 1$ and vanishing at infinity $\Rightarrow$

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{(\varepsilon \hat{h})^2 + |\nabla u|^2}}\right).$$

- This motivates us to look at :

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Numerical results ...

# The motivation: numerical implementation of the ROF model.

- **Problem 1** : $|\nabla u| \approx 0$ : $u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}\right)$
- **Problem 2** : How to deal with the sharp discontinuities ?
- **Idea** : increase the cell size $h$ if $|\nabla u|$ is large: **non-uniform grid**.
- Let $\hat{h} = \frac{h}{g(|G \star \nabla u|)}$ with $g(0) = 1$ and vanishing at infinity $\Rightarrow$

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{(\varepsilon \hat{h})^2 + |\nabla u|^2}}\right).$$

- This motivates us to look at :

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Numerical results ...

# The motivation: numerical implementation of the ROF model.

- **Problem 1** : $|\nabla u| \approx 0$ : $u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}\right)$
- **Problem 2** : How to deal with the sharp discontinuities ?
- **Idea** : increase the cell size $h$ if $|\nabla u|$ is large: **non-uniform grid**.
- Let $\hat{h} = \frac{h}{g(|G \star \nabla u|)}$ with $g(0) = 1$ and vanishing at infinity $\Rightarrow$

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{(\varepsilon \hat{h})^2 + |\nabla u|^2}}\right).$$

- This motivates us to look at :

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Numerical results ...

# The motivation: numerical implementation of the ROF model.

- **Problem 1** : $|\nabla u| \approx 0$ : $u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}\right)$
- **Problem 2** : How to deal with the sharp discontinuities ?
- **Idea** : increase the cell size $h$ if $|\nabla u|$ is large: **non-uniform grid**.
- Let $\hat{h} = \frac{h}{g(|G \star \nabla u|)}$ with $g(0) = 1$ and vanishing at infinity $\Rightarrow$

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{(\varepsilon \hat{h})^2 + |\nabla u|^2}}\right).$$

- This motivates us to look at :

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Numerical results ...

# The motivation: numerical implementation of the ROF model.

- **Problem 1** : $|\nabla u| \approx 0$ : $u = f + \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}\right)$
- **Problem 2** : How to deal with the sharp discontinuities ?
- **Idea** : increase the cell size $h$ if $|\nabla u|$ is large: **non-uniform grid**.
- Let $\hat{h} = \frac{h}{g(|G \star \nabla u|)}$ with $g(0) = 1$ and vanishing at infinity $\Rightarrow$

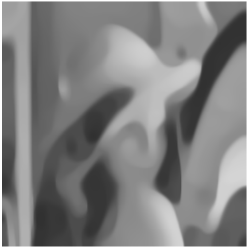
$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{\sqrt{(\varepsilon \hat{h})^2 + |\nabla u|^2}}\right).$$

- This motivates us to look at :

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Numerical results ...

# Comparison between standard ROF and modified ROF



(a)

(b)

**Figure:** (a) Result of standard ROF and (b) result of the modified ROF with $g(|G \star \nabla u|)$; both for the same $\lambda = 0.0001$.

# Modified TNV and the proposed model

- TNV scheme:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- We looked at modified ROF:

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$$

- Can we modify TNV ?

**Modified TNV scheme**

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G \star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- Numerical results ...

# Modified TNV and the proposed model

- TNV scheme:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- We looked at modified ROF:

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$$

- Can we modify TNV ?

**Modified TNV scheme**

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G \star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- Numerical results ...

# Modified TNV and the proposed model

- TNV scheme:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- We looked at modified ROF:

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$$

- Can we modify TNV ?

**Modified TNV scheme**

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G \star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- Numerical results ...

# Modified TNV and the proposed model

- TNV scheme:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{1}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- We looked at modified ROF:

$$u = f + \frac{g(|G \star \nabla u|)}{2\lambda} \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$$

- Can we modify TNV ?

## Modified TNV scheme

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G \star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- Numerical results ...

# Numerical results of modified TNV.

- Numerical results of $\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G_\star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$.

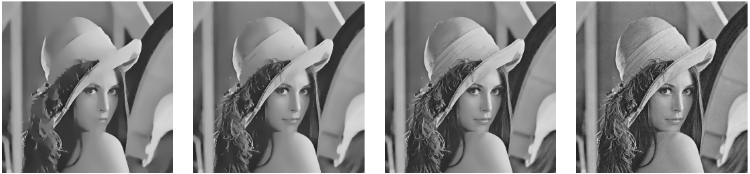


**Figure:** Numerical results of TNV with diffusion controlling function $g(s) = \frac{1}{1+s^2}$, with initial $\lambda_k = (0.0005) \times 2^k$.

- Can we modify the integro-differential model ?

# Numerical results of modified TNV.

- Numerical results of $\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G_\star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$.

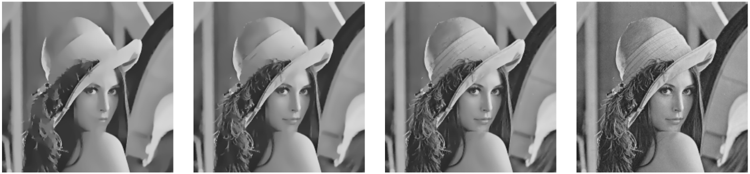


**Figure:** Numerical results of TNV with diffusion controlling function $g(s) = \frac{1}{1+s^2}$, with initial $\lambda_k = (0.0005) \times 2^k$.

- Can we modify the integro-differential model ?

- We looked at modified TNV:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G \star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- This leads to modified Integro-differential equation:

$$\int_0^t u(x,s)ds = f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$$

- Numerical results ...

- We looked at modified TNV:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G \star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- This leads to modified Integro-differential equation:

$$\int_0^t u(x,s)ds = f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$$

- Numerical results ...

- We looked at modified TNV:

$$\sum_{k=0}^{N} u_{\lambda_k} = f + \frac{g(|G \star \nabla u_{\lambda_N}|)}{2\lambda_N} \operatorname{div}\left(\frac{\nabla u_{\lambda_N}}{|\nabla u_{\lambda_N}|}\right)$$

- This leads to modified Integro-differential equation:

$$\int_0^t u(x,s)ds = f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$$

- Numerical results ...

# Numerical results of the modified integro-differential model

Numerical results of $\int_0^t u(x,s)ds = f(x) + \frac{g(|G_\star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$.

**Figure:** Numerical results of modified integro-differential mode with diffusion controlling function $g(s) = \frac{1}{1+s^2}$. The function $\lambda(t) = 10^{-4}e^{2t} \times 2^t$.

# What does the scaling function $\lambda(t)$ mean ?

- Recall the integro-differential model

$$\int_0^t u(x,s)ds - f(x) = \frac{1}{2\lambda(t)} \operatorname{div}\left( \frac{\nabla u(x,t)}{|\nabla u(x,t)|} \right).$$

  where $\lambda(t)$ is an increasing function at our disposal.

- Star-norm is the dual of the $BV$ norm w.r.t. the $L^2$ scalar product

$$\|v\|_* := \sup_{\phi \neq 0} \frac{\langle v, \phi \rangle}{\int_\Omega |\nabla \phi|}.$$

- **Theorem**: Let us define for the integro-differential equation the error term as $\int_0^t u(x,s)ds - f(x)$, then

$$\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}.$$

- **Proof**: The proof follows from Meyer's theorem: for ROF decomposition $f = u + v$ with scale $\lambda$, we get $\int_\Omega uv = \|u\|_{BV}\|v\|_*$ and $\|v\|_* = \frac{1}{2\lambda}$.

# What does the scaling function $\lambda(t)$ mean ?

- Recall the integro-differential model

$$\int_0^t u(x,s)ds - f(x) = \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

  where $\lambda(t)$ is an increasing function at our disposal.
- Star-norm is the dual of the $BV$ norm w.r.t. the $L^2$ scalar product

$$\|v\|_* := \sup_{\phi \neq 0} \frac{\langle v, \phi \rangle}{\int_\Omega |\nabla \phi|}.$$

- **Theorem**: Let us define for the integro-differential equation the error term as $\int_0^t u(x,s)ds - f(x)$, then

$$\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}.$$

- **Proof**: The proof follows from Meyer's theorem: for ROF decomposition $f = u + v$ with scale $\lambda$, we get $\int_\Omega uv = \|u\|_{BV}\|v\|_*$ and $\|v\|_* = \frac{1}{2\lambda}$.

# What does the scaling function $\lambda(t)$ mean ?

- Recall the integro-differential model

$$\int_0^t u(x,s)ds - f(x) = \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

  where $\lambda(t)$ is an increasing function at our disposal.

- Star-norm is the dual of the $BV$ norm w.r.t. the $L^2$ scalar product

$$\|v\|_* := \sup_{\phi \neq 0} \frac{\langle v, \phi \rangle}{\int_\Omega |\nabla \phi|}.$$

- **Theorem**: Let us define for the integro-differential equation the error term as $\int_0^t u(x,s)ds - f(x)$, then

$$\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}.$$

- **Proof**: The proof follows from Meyer's theorem: for ROF decomposition $f = u + v$ with scale $\lambda$, we get $\int_\Omega uv = \|u\|_{BV}\|v\|_*$ and $\|v\|_* = \frac{1}{2\lambda}$.

# What does the scaling function $\lambda(t)$ mean ?

- Recall the integro-differential model

$$\int_0^t u(x,s)ds - f(x) = \frac{1}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right).$$

  where $\lambda(t)$ is an increasing function at our disposal.
- Star-norm is the dual of the $BV$ norm w.r.t. the $L^2$ scalar product

$$\|v\|_* := \sup_{\phi \neq 0} \frac{\langle v, \phi \rangle}{\int_\Omega |\nabla \phi|}.$$

- **Theorem**: Let us define for the integro-differential equation the error term as $\int_0^t u(x,s)ds - f(x)$, then

$$\| \int_0^t u(x,s)ds - f(x) \|_* = \frac{1}{2\lambda(t)}.$$

- **Proof**: The proof follows from Meyer's theorem: for ROF decomposition $f = u + v$ with scale $\lambda$, we get $\int_\Omega uv = \|u\|_{BV}\|v\|_*$ and $\|v\|_* = \frac{1}{2\lambda}$.

**Remark**: This theorem is important, in the sense that it dictates the star-norm of the residual $\int_0^t u(x,s)ds - f(x)$ at any time. We get the following result using this property.

**Corollary**: The star-norm of the residual vanishes as $t \to \infty$

**Proof**: In our model the function $\lim_{t\to\infty} \lambda(t) = \infty$. Thus, the result follows from the previous theorem :

$$\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}.$$

**Question**: What happens if $\lim_{t\to\infty} \lambda(t) = \lambda_\infty < \infty$ ?

**Remark**: This theorem is important, in the sense that it dictates the star-norm of the residual $\int_0^t u(x,s)ds - f(x)$ at any time. We get the following result using this property.

**Corollary**: The star-norm of the residual vanishes as $t \to \infty$

**Proof**: In our model the function $\lim_{t\to\infty} \lambda(t) = \infty$. Thus, the result follows from the previous theorem :

$$\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}.$$

**Question**: What happens if $\lim_{t\to\infty} \lambda(t) = \lambda_\infty < \infty$ ?

**Remark**: This theorem is important, in the sense that it dictates the star-norm of the residual $\int_0^t u(x,s)ds - f(x)$ at any time. We get the following result using this property.

**Corollary**: The star-norm of the residual vanishes as $t \to \infty$

**Proof**: In our model the function $\lim_{t\to\infty} \lambda(t) = \infty$. Thus, the result follows from the previous theorem :

$$\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}.$$

**Question**: What happens if $\lim_{t\to\infty} \lambda(t) = \lambda_\infty < \infty$ ?

**Remark**: This theorem is important, in the sense that it dictates the star-norm of the residual $\int_0^t u(x,s)ds - f(x)$ at any time. We get the following result using this property.

**Corollary**: The star-norm of the residual vanishes as $t \to \infty$

**Proof**: In our model the function $\lim_{t\to\infty} \lambda(t) = \infty$. Thus, the result follows from the previous theorem :

$$\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}.$$

**Question**: What happens if $\lim_{t\to\infty} \lambda(t) = \lambda_\infty < \infty$ ?

# Where to start and stop ?

- **Starting** : We know that for ROF decomposition if $\|f\|_* < \frac{1}{2\lambda}$ then $v = f$, thus we start with a very small value of $\lambda(t)$.
- **Stopping** : This is an open problem.
  We know : $\| \int_0^t u(x,s)ds - f(x) \|_* = \frac{1}{2\lambda(t)}$.

**Question** : What does the star-norm really mean ?

# Where to start and stop ?

- **Starting** : We know that for ROF decomposition if $\|f\|_* < \frac{1}{2\lambda}$ then $v = f$, thus we start with a very small value of $\lambda(t)$.

- **Stopping** : This is an open problem.
  We know : $\|\int_0^t u(x,s)ds - f(x)\|_* = \frac{1}{2\lambda(t)}$.

**Question** : What does the star-norm really mean ?

**Observation**: Let $f$ and $g$ be in $(X, (\cdot,\cdot))$, an inner product space.

Let $f \sim \{u_k, v_k\}_{k=0}^{\infty}$ and $g \sim \{U_k, V_k\}_{k=0}^{\infty}$ be **any** hierarchical decompositions :

$v_{-1} = f$ and $v_{k-1} = u_k + v_k$ for $k = 0, 1, 2 \ldots \infty$ and
$V_{-1} = g$ and $V_{k-1} = U_k + V_k$ for $k = 0, 1, 2 \ldots \infty$.

Let us define an inner product

$$\langle f, g\rangle = \sum_{k=0}^{\infty} (u_k, U_k) + (v_k, U_k) + (u_k, V_k) + \cancel{(v_k, V_k)}$$

Then $\langle f, g\rangle = (f, g)$ if and only if $\lim_{k\to\infty}(v_k, V_k) = 0$.

Indeed for $(BV, L^2)$ multiscale decompositions: $\|v_{\lambda_k}\|_{L^2} \to 0$ ...

**Observation**: Let $f$ and $g$ be in $(X, (\cdot,\cdot))$, an inner product space.

Let $f \sim \{u_k, v_k\}_{k=0}^{\infty}$ and $g \sim \{U_k, V_k\}_{k=0}^{\infty}$ be **any** hierarchical decompositions :

$v_{-1} = f$ and $v_{k-1} = u_k + v_k$ for $k = 0, 1, 2 \ldots \infty$ and
$V_{-1} = g$ and $V_{k-1} = U_k + V_k$ for $k = 0, 1, 2 \ldots \infty$.

Let us define an inner product

$$\langle f, g \rangle = \sum_{k=0}^{\infty} (u_k, U_k) + (v_k, U_k) + (u_k, V_k) + \cancel{(v_k, V_k)}$$

Then $\langle f, g \rangle = (f, g)$ if and only if $\lim_{k\to\infty}(v_k, V_k) = 0$.

Indeed for $(BV, L^2)$ multiscale decompositions: $\|v_{\lambda_k}\|_{L^2} \to 0$ ...

**Observation**: Let $f$ and $g$ be in $(X, (\cdot,\cdot))$, an inner product space.

Let $f \sim \{u_k, v_k\}_{k=0}^{\infty}$ and $g \sim \{U_k, V_k\}_{k=0}^{\infty}$ be **any** hierarchical decompositions :

$v_{-1} = f$ and $v_{k-1} = u_k + v_k$ for $k = 0, 1, 2 \ldots \infty$ and
$V_{-1} = g$ and $V_{k-1} = U_k + V_k$ for $k = 0, 1, 2 \ldots \infty$.

Let us define an inner product

$$\langle f, g\rangle = \sum_{k=0}^{\infty} (u_k, U_k) + (v_k, U_k) + (u_k, V_k) + \cancel{(v_k, V_k)}$$

Then $\langle f, g\rangle = (f, g)$ if and only if $\lim_{k\to\infty}(v_k, V_k) = 0$.

Indeed for $(BV, L^2)$ multiscale decompositions: $\|v_{\lambda_k}\|_{L^2} \to 0$ ...

**Observation**: Let $f$ and $g$ be in $(X, (\cdot,\cdot))$, an inner product space.

Let $f \sim \{u_k, v_k\}_{k=0}^{\infty}$ and $g \sim \{U_k, V_k\}_{k=0}^{\infty}$ be **any** hierarchical decompositions :

$v_{-1} = f$ and $v_{k-1} = u_k + v_k$ for $k = 0, 1, 2 \ldots \infty$ and
$V_{-1} = g$ and $V_{k-1} = U_k + V_k$ for $k = 0, 1, 2 \ldots \infty$.

Let us define an inner product

$$\langle f, g \rangle = \sum_{k=0}^{\infty} (u_k, U_k) + (v_k, U_k) + (u_k, V_k) + \cancel{(v_k, V_k)}$$

Then $\langle f, g \rangle = (f, g)$ if and only if $\lim_{k\to\infty}(v_k, V_k) = 0$.

Indeed for $(BV, L^2)$ multiscale decompositions: $\|v_{\lambda_k}\|_{L^2} \to 0$ ...

**Observation**: Let $f$ and $g$ be in $(X, (\cdot, \cdot))$, an inner product space.

Let $f \sim \{u_k, v_k\}_{k=0}^{\infty}$ and $g \sim \{U_k, V_k\}_{k=0}^{\infty}$ be **any** hierarchical decompositions :

$v_{-1} = f$ and $v_{k-1} = u_k + v_k$ for $k = 0, 1, 2 \dots \infty$ and
$V_{-1} = g$ and $V_{k-1} = U_k + V_k$ for $k = 0, 1, 2 \dots \infty$.

Let us define an inner product

$$\langle f, g \rangle = \sum_{k=0}^{\infty} (u_k, U_k) + (v_k, U_k) + (u_k, V_k) + \cancel{(v_k, V_k)}$$

Then $\langle f, g \rangle = (f, g)$ if and only if $\lim_{k \to \infty} (v_k, V_k) = 0$.

Indeed for $(BV, L^2)$ multiscale decompositions: $\|v_{\lambda_k}\|_{L^2} \to 0$ ...

# Consequences of this observation

For $(BV, L^2)$ hierarchical decomposition of $f$ we get

$$(f, f)_{L^2} = \langle f, f \rangle = \sum_{k=0}^{\infty} (u_{\lambda_k}, u_{\lambda_k})_{L^2} + (v_{\lambda_k}, u_{\lambda_k})_{L^2} + (u_{\lambda_k}, v_{\lambda_k})_{L^2}$$

$$\|f\|_{L^2}^2 = \sum_{k=0}^{\infty} \|u_{\lambda_k}\|_{L^2}^2 + 2(u_{\lambda_k}, v_{\lambda_k})_{L^2}.$$

Meyer's theorem: $(u_{\lambda_k}, v_{\lambda_k})_{L^2} = \frac{1}{2\lambda_k} \|u_{\lambda_k}\|_{BV}$ we get:

**Energy decomposition (Tadmor et. al. 2004)**

$$\|f\|_{L^2}^2 = \sum_{k=0}^{\infty} \|u_{\lambda_k}\|_{L^2}^2 + \frac{1}{\lambda_k} \|u_{\lambda_k}\|_{BV}.$$

# Consequences of this observation

For $(BV, L^2)$ hierarchical decomposition of $f$ we get

$$(f, f)_{L^2} = \langle f, f \rangle = \sum_{k=0}^{\infty} (u_{\lambda_k}, u_{\lambda_k})_{L^2} + (v_{\lambda_k}, u_{\lambda_k})_{L^2} + (u_{\lambda_k}, v_{\lambda_k})_{L^2}$$

$$\|f\|_{L^2}^2 = \sum_{k=0}^{\infty} \|u_{\lambda_k}\|_{L^2}^2 + 2(u_{\lambda_k}, v_{\lambda_k})_{L^2}.$$

Meyer's theorem: $(u_{\lambda_k}, v_{\lambda_k})_{L^2} = \frac{1}{2\lambda_k} \|u_{\lambda_k}\|_{BV}$ we get:

**Energy decomposition (Tadmor et. al. 2004)**

$$\|f\|_{L^2}^2 = \sum_{k=0}^{\infty} \|u_{\lambda_k}\|_{L^2}^2 + \frac{1}{\lambda_k} \|u_{\lambda_k}\|_{BV}.$$

## Consequences of this observation

For $(BV, L^2)$ hierarchical decomposition of $f$ we get

$$(f, f)_{L^2} = \langle f, f \rangle = \sum_{k=0}^{\infty} (u_{\lambda_k}, u_{\lambda_k})_{L^2} + (v_{\lambda_k}, u_{\lambda_k})_{L^2} + (u_{\lambda_k}, v_{\lambda_k})_{L^2}$$

$$\|f\|_{L^2}^2 = \sum_{k=0}^{\infty} \|u_{\lambda_k}\|_{L^2}^2 + 2(u_{\lambda_k}, v_{\lambda_k})_{L^2}.$$

Meyer's theorem: $(u_{\lambda_k}, v_{\lambda_k})_{L^2} = \frac{1}{2\lambda_k} \|u_{\lambda_k}\|_{BV}$ we get:

**Energy decomposition (Tadmor et. al. 2004)**

$$\|f\|_{L^2}^2 = \sum_{k=0}^{\infty} \|u_{\lambda_k}\|_{L^2}^2 + \frac{1}{\lambda_k} \|u_{\lambda_k}\|_{BV}.$$

# Other works

- **Deblurring**: Tadmor et. al. (2008) consider hierarchical decomposition:

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = Tu_{\lambda_k} + v_{\lambda_k}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - Tu_{\lambda_k}|^2 \right)$$

$$T^* T u_{\lambda_k} = T^* v_{\lambda_{k-1}} + \frac{1}{2\lambda_k} \operatorname{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)$$

(a) (b)

**Figure:** A blurred image (a) is deblurred as shown in (b).

### Novel 'deblurring' integro-differential equation

$$\int_0^t T^* T u(x,s)ds = T^* f(x) + \frac{1}{2\lambda(t)} \operatorname{div} \left( \frac{\nabla u(x,t)}{|\nabla u(x,t)|} \right)$$

# Other works

- **Deblurring**: Tadmor et. al. (2008) consider hierarchical decomposition:

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = Tu_{\lambda_k} + v_{\lambda_k}\}}{\operatorname{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - Tu_{\lambda_k}|^2 \right)$$

$$T^* Tu_{\lambda_k} = T^* v_{\lambda_{k-1}} + \frac{1}{2\lambda_k} \operatorname{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)$$

(a) (b)

**Figure:** A blurred image (a) is deblurred as shown in (b).

**Novel 'deblurring' integro-differential equation**

$$\int_0^t T^* Tu(x,s)ds = T^* f(x) + \frac{1}{2\lambda(t)} \operatorname{div} \left( \frac{\nabla u(x,t)}{|\nabla u(x,t)|} \right)$$

# Other works

- **Deblurring**: Tadmor et. al. (2008) consider hierarchical decomposition:

$$[u_{\lambda_k}, v_{\lambda_k}] = \underset{\{v_{\lambda_{k-1}} = Tu_{\lambda_k} + v_{\lambda_k}\}}{\text{arginf}} \left( \int_\Omega |\nabla u_{\lambda_k}| + \lambda_k \int_\Omega |v_{\lambda_{k-1}} - Tu_{\lambda_k}|^2 \right)$$

$$T^* T u_{\lambda_k} = T^* v_{\lambda_{k-1}} + \frac{1}{2\lambda_k} \text{div} \left( \frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|} \right)$$

(a) (b)

**Figure:** A blurred image (a) is deblurred as shown in (b).

**Novel 'deblurring' integro-differential equation**

$$\int_0^t T^* T u(x,s) ds = T^* f(x) + \frac{1}{2\lambda(t)} \text{div} \left( \frac{\nabla u(x,t)}{|\nabla u(x,t)|} \right)$$

# Other works ...

## Modified TNV deblurring

$$T^* T u_{\lambda_k} = T^* v_{\lambda_{k-1}} + \frac{g(|G \star \nabla u_{\lambda_k}|)}{2\lambda_k} \operatorname{div}\left(\frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|}\right)$$

(a) (b)

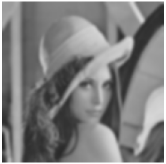
**Figure:** A blurred image (a) is deblurred with modified TNV deblurring as shown in (b).

**Modified 'deblurring' integro-differential equation**

$$\int_0^t T^* T u(x,s)ds = T^* f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$$

## Modified TNV deblurring

$$T^* T u_{\lambda_k} = T^* v_{\lambda_{k-1}} + \frac{g(|G \star \nabla u_{\lambda_k}|)}{2\lambda_k} \operatorname{div}\left(\frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|}\right)$$



(a)

(b)

**Figure:** A blurred image (a) is deblurred with modified TNV deblurring as shown in (b).

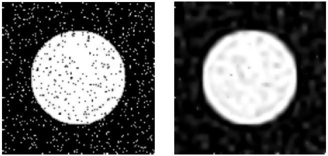
### Modified ‘deblurring’ integro-differential equation

$$\int_0^t T^* T u(x,s) ds = T^* f(x) + \frac{g(|G \star \nabla u(x,t)|)}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$$

# Other works ...

- **Modified TNV deblurring with selective diffusion**:

$$T^* T u_{\lambda_k} = T^* v_{\lambda_{k-1}} + \frac{g(|G \star \nabla u_{\lambda_k}|)|\nabla u_{\lambda_k}|}{2\lambda_k} \operatorname{div}\left(\frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|}\right)$$



(a) (b)

**Figure:** A blurred image (a) is denoised with modified TNV deblurring with selective diffusion as shown in (b).

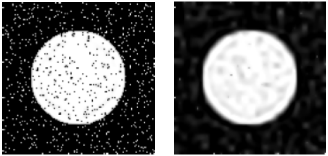
**Modified ‘deblurring’ integro-differential equation**

$$\int_0^t T^* T u(x,s)ds = T^* f(x) + \frac{g(|G \star \nabla u(x,t)|)|\nabla u(x,t)|}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$$

# Other works ...

- **Modified TNV deblurring with selective diffusion**:

$$T^* T u_{\lambda_k} = T^* v_{\lambda_{k-1}} + \frac{g(|G \star \nabla u_{\lambda_k}|)|\nabla u_{\lambda_k}|}{2\lambda_k} \operatorname{div}\left(\frac{\nabla u_{\lambda_k}}{|\nabla u_{\lambda_k}|}\right)$$



(a) (b)

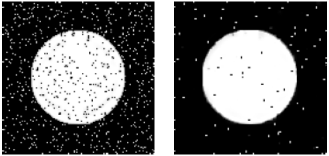
**Figure:** A blurred image (a) is denoised with modified TNV deblurring with selective diffusion as shown in (b).

## Modified 'deblurring' integro-differential equation

$$\int_0^t T^* T u(x,s)ds = T^* f(x) + \frac{g(|G \star \nabla u(x,t)|)|\nabla u(x,t)|}{2\lambda(t)} \operatorname{div}\left(\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right)$$

# Other works ...

- multiscale $(BV, L^1)$
- multiscale $(BV, (L^1)^2)$
- multiscale $(BV, L^1), (BV, (L^1)^2)$ with $g(|G \star \nabla u|)$ and $g(|G \star \nabla u|)|\nabla u|$

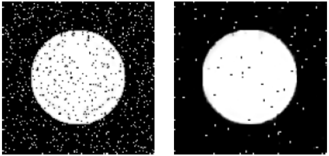


(a) (b)

**Figure:** A blurred image (a) is denoised with multiscale $(BV, (L^1)^2)$ with $g(|G \star \nabla u|)|\nabla u|$ as shown in (b).

# Other works ...

- multiscale $(BV, L^1)$
- multiscale $(BV, (L^1)^2)$
- multiscale $(BV, L^1), (BV, (L^1)^2)$ with $g(|G \star \nabla u|)$ and $g(|G \star \nabla u|)|\nabla u|$

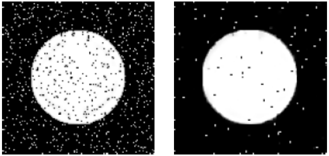


(a) (b)

**Figure:** A blurred image (a) is denoised with multiscale $(BV, (L^1)^2)$ with $g(|G \star \nabla u|)|\nabla u|$ as shown in (b).

# Other works ...

- multiscale $(BV, L^1)$
- multiscale $(BV, (L^1)^2)$
- multiscale $(BV, L^1), (BV, (L^1)^2)$ with $g(|G \star \nabla u|)$ and $g(|G \star \nabla u|)|\nabla u|$



(a) (b)

**Figure:** A blurred image (a) is denoised with multiscale $(BV, (L^1)^2)$ with $g(|G \star \nabla u|)|\nabla u|$ as shown in (b).