Image Processing using Partial Differential Equations (PDE)

Restoration, segmentation, tracking, optical flow estimation, registration

Pierre Kornprobst

NeuroMathComp project team
INRIA Sophia Antipolis - Méditerranée

Vision Student Talks [ViST]

April 2014

My goals today

- Introduce methodology

My goals today

- Introduce methodology
- Show links between formulations (min, $\frac{\partial .}{\partial t}$, $\int\int$)

My goals today

- Introduce methodology
- Show links between formulations (min, $\frac{\partial .}{\partial t}$, $\int\int$)
- Show a success story: *level-sets*

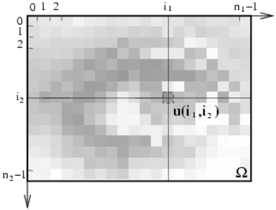
1. Can we use PDEs to do some interesting image processing?
   - Definitions
   - From Gaussian filtering to the heat equation

1. Can we use PDEs to do some interesting image processing?
   - Definitions
   - From Gaussian filtering to the heat equation

2. Three solutions to go further
   - Solution 1: Make convolution "nonlinear"
   - Solution 2: Modify the heat equation
   - Solution 3: Define an optimisation problem

3. A success story: Levels-sets

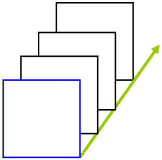
An image is seen as a function defined in continuous space



$$u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$

A PDE defines an evolution

$$u(t,x) \; / \; \frac{\partial u}{\partial t} = H(t,x,u,\nabla u,\nabla^2 u) \;\leadsto\; v(x) \equiv u(\infty,x)$$



1. Can we use PDEs to do some interesting image processing?
   - Definitions
   - From Gaussian filtering to the heat equation

2. Three solutions to go further
   - Solution 1: Make convolution "nonlinear"
   - Solution 2: Modify the heat equation
   - Solution 3: Define an optimisation problem

3. A success story: Levels-sets

4. Conclusion

Gaussian filtering

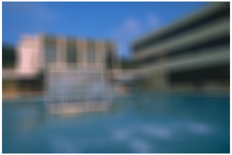
- Let u_0 an image, we define :

$$u_{\sigma}(x) = (G_{\sigma} * u_0)(x) \quad \text{avec} \quad G_{\sigma}(x) = \frac{1}{2\pi\,\sigma^2}\ \exp\left(-\frac{|x|^2}{2\,\sigma^2}\right).$$

$\sigma = 0$

$\sigma = 5$

$\sigma = 11$

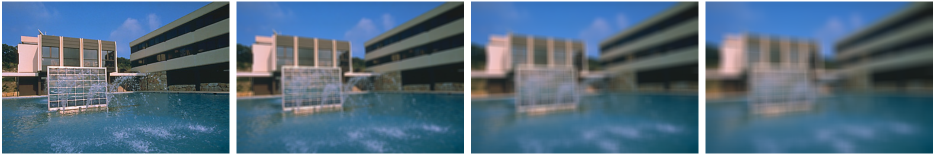


$\sigma = 17$

Heat equation

- A linear PDE

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) = \Delta u(t, x), \quad t \geq 0, \\ u(0, x) = u_0(x). \end{cases}$$



$t = 0$ $t = 12.5$ $t = 60.5$ $t = 93.5$

- A notion of scale

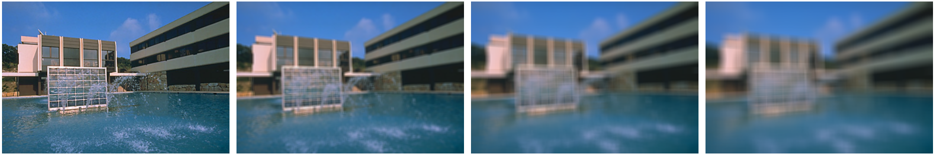
Solution of the heat equation is a convolution

$u(t,x) = (G_{\sqrt{2t}} * u_0)(x)$

Gaussian filtering

One operation in a large neighbourhood

$\sigma = 0$	$\sigma = 5$	$\sigma = 11$	$\sigma = 17$



$t = 0$	$t = 12.5$	$t = 60.5$	$t = 93.5$

Heat equation

A succession of local operations

Take home messages

- PDE appear as a natural way to smooth images.
- When it is linear, a PDE (or equivalently the convolution) do not preserve edges.

1. Can we use PDEs to do some interesting image processing?
   - Definitions
   - From Gaussian filtering to the heat equation

2. Three solutions to go further
   - Solution 1: Make convolution "nonlinear"
   - Solution 2: Modify the heat equation
   - Solution 3: Define an optimisation problem

3. A success story: Levels-sets

4. Conclusion

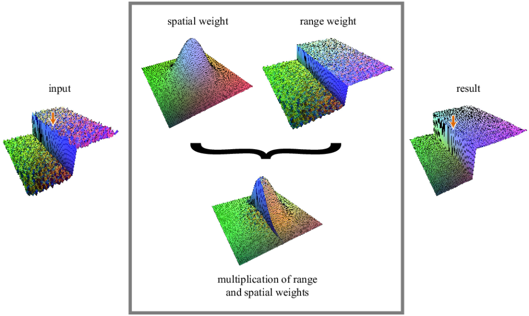
Solution 1: Make convolution "nonlinear"

Bilateral filtering (Tomasi, Manduchi [1998])

- Given u_0, we define u by :

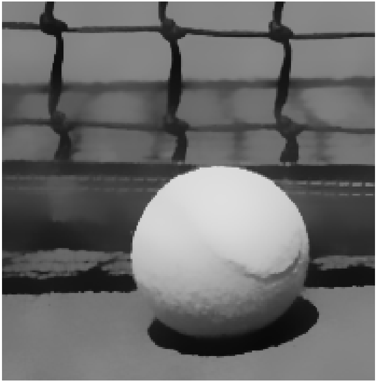
$$u(x) = \frac{1}{w(x)} \iint d(x-\xi)\tilde{d}(u_0(x) - u_0(\xi))u_0(\xi)d\xi \quad \text{with}$$

$$w(x) = \iint d(x-\xi)\tilde{d}(u_0(x) - u_0(\xi))d\xi$$



Denoising and Simplification

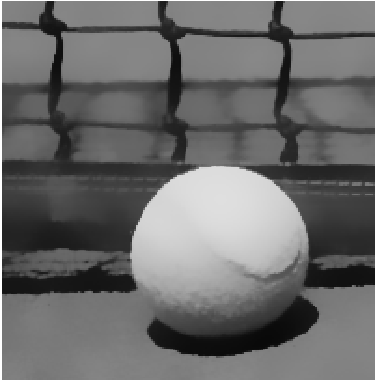
before



after

Denoising and Simplification

before



after

Computer Graphics community like it!

Numerous improvements, extensions, efficient implementations and great applications

Do your own comics

Winnemoller, Olsen, Gooch [2006]

(video-spiderman.avi)

before

after

Tone management for photographic look

Bae, Paris, Durand [2006]

Tone management for photographic look

Bae, Paris, Durand [2006]

"Clearing winter storm", Ansel Adams

Tone management for photographic look

Bae, Paris, Durand [2006]

Our classical picture

Tone management for photographic look

Bae, Paris, Durand [2006]

model

before

Tone management for photographic look

Bae, Paris, Durand [2006]

model

after

Solution 2: Modify the heat equation

- Heat equation

$$\frac{\partial u}{\partial t} = \Delta u = \mathrm{div}(\ ?\ \nabla u)$$

Solution 2: Modify the heat equation

- Heat equation

$$\frac{\partial u}{\partial t} = \Delta u = \mathrm{div}(\ ?\ \nabla u)$$

- Perona and Malik model [1990]

$$\frac{\partial u}{\partial t} = \mathrm{div}\left(\underbrace{c(|\nabla u|^2)}_{\text{scalar}}\nabla u\right) \text{ with } c(s) = \begin{cases} 1/\sqrt{1+s} \\ \exp(-s) \end{cases}$$

Solution 2: Modify the heat equation

- Heat equation

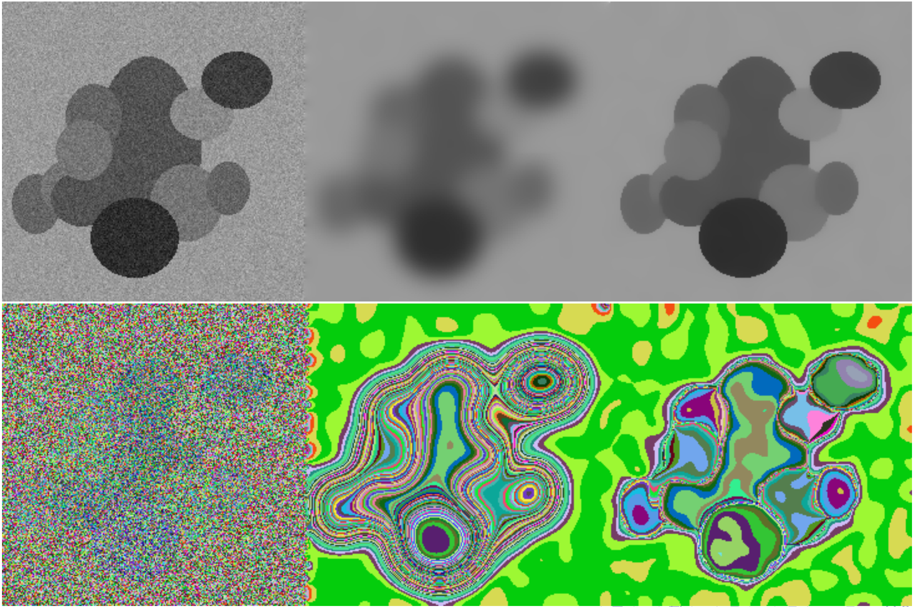
$$\frac{\partial u}{\partial t} = \Delta u = \mathrm{div}(\ ?\ \nabla u)$$

- Perona and Malik model [1990]

$$\frac{\partial u}{\partial t} = \mathrm{div}\left(\underbrace{c(|\nabla u|^2)}_{\text{scalar}}\nabla u\right) \text{ with } c(s) = \begin{cases} 1/\sqrt{1+s} \\ \exp(-s) \end{cases}$$

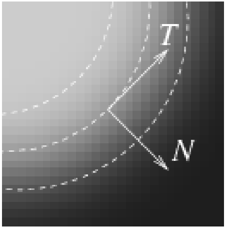
- Buades, Coll et Morel [2005]: Bilateral filter is related to Perona and Malik model

Diffusion acts on isophotes



Diffusion can interpreted w.r.t. local image structures

- Nonlinear diffusion is non only a "controlled" diffusion but it is also related to a directional diffusion depending on local image structures



- Most diffusion operators can be rewritten as:

$$\Big(\dots\Big)u_{TT} + \Big(\dots\Big)u_{NN}$$

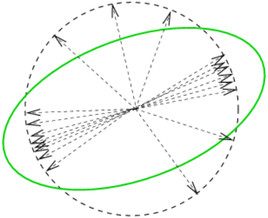
How to better take into account local image structure?

- Limitation of previous models: Estimation of T, N is local

How to better take into account local image structure?

- Limitation of previous models: Estimation of T, N is local
- Solution: Use structure tensor

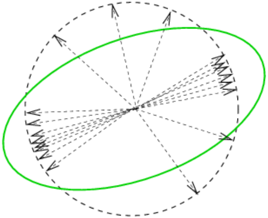
$$k_\rho * \nabla u_\sigma \nabla u_\sigma^t = k_\rho * \begin{pmatrix} u_{\sigma xx} & u_{\sigma xy} \\ u_{\sigma xy} & u_{\sigma yy} \end{pmatrix}$$



How to better take into account local image structure?

- Limitation of previous models: Estimation of T, N is local
- Solution: Use structure tensor

$$k_\rho * \nabla u_\sigma \nabla u_\sigma^t = k_\rho * \begin{pmatrix} u_{\sigma xx} & u_{\sigma xy} \\ u_{\sigma xy} & u_{\sigma yy} \end{pmatrix}$$

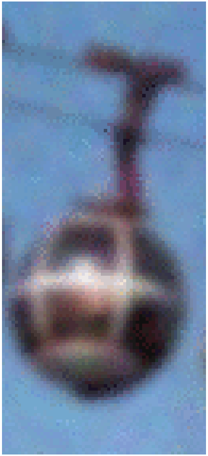


- Weickert [1996]

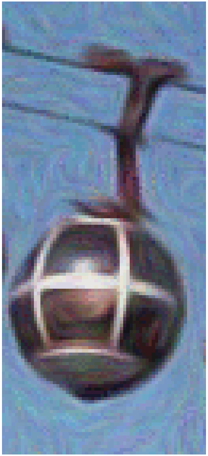
$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\underbrace{D(k_\rho * \nabla u_\sigma \nabla u_\sigma^t)}_{\text{matrix}} \nabla u \right)$$

Example


Original

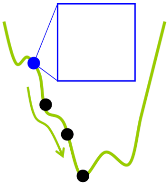

Chaleur


Perona-Malik


Weickert

When a PDE is a gradient descent of an optimisation problem (also called variational problem)

$$v(x) = \underset{u(x)}{\text{Argmin}} \; E(u) = \int_\Omega F(x, u, \nabla u, \nabla^2 u) dx$$



Solution 3: Define an optimisation problem

- Let us start from a model of formation of images, for example :

$$u_0 = R\,u + \eta$$

where η is a white Gaussian noise and R is a linear operator.

Solution 3: Define an optimisation problem

- Let us start from a model of formation of images, for example :

$$u_0 = R\,u + \eta$$

where η is a white Gaussian noise and R is a linear operator.

- Solve the least-square problem

$$\inf_u \int_\Omega |u_0 - Ru|^2\,dx \quad \rightarrow \quad R^* u_0 - R^* R u = 0$$

is an ill-posed problem

- Why? Opertor $R^* R$ is usually hard to invert (not bijective or low eingenvalues)

Constrain possible solutions thanks to a regularity constraint

- Tikhonov et Arsenin [1977]

$$\inf_u \int_\Omega |u_0 - Ru|^2 \, dx + \lambda \int_\Omega |\nabla u|^2 dx$$

Constrain possible solutions thanks to a regularity constraint

- Tikhonov et Arsenin [1977]

$$\inf_{u} \int_{\Omega} |u_0 - Ru|^2 \, dx + \lambda \int_{\Omega} |\nabla u|^2 dx$$

- The two main ingredients are here!

Constrain possible solutions thanks to a regularity constraint

- Tikhonov et Arsenin [1977]

$$\inf_u \int_\Omega |u_0 - Ru|^2 \, dx + \lambda \int_\Omega |\nabla u|^2 dx$$

- The two main ingredients are here!
- To minimise, compute the Euler-Lagrange equation, and in that case, we find again the Laplacian operator.

$$EL(u) = \lambda \triangle u - (R^* Ru - R^* u_0) = 0, \quad \frac{\partial u}{\partial t} = EL(u).$$

So, quadratic penalty gives linear diffusion, and othewise?

- Consider a general formulation :

$$\inf_u \; E(u) = \int_\Omega (u_0 - Ru)^2 \; dx + \lambda \int_\Omega \phi(|\nabla u|) \; dx$$

- Diffusion operator :

$$\mathrm{div}\left(\frac{\phi'(|\nabla u|)}{|\nabla u|}\nabla u\right) = \frac{\phi'(|\nabla u|)}{|\nabla u|} \; u_{TT} + \phi''(|\nabla u|) \; u_{NN}$$

So, quadratic penalty gives linear diffusion, and othewise?

- Consider a general formulation :

$$\inf_u \; E(u) = \int_\Omega (u_0 - Ru)^2 \; dx + \lambda \int_\Omega \phi(|\nabla u|) \; dx$$

- Diffusion operator :

$$\operatorname{div}\left(\frac{\phi'(|\nabla u|)}{|\nabla u|}\nabla u\right) = \frac{\phi'(|\nabla u|)}{|\nabla u|} \; u_{TT} + \phi''(|\nabla u|) \; u_{NN}$$

- Choosing ϕ will have consequences on the solution regularity and as a consequence on the functional space to consider to study the variational formulation (not commented today, but lots of interesting maths here)

$$\phi(s) = 2\sqrt{1+s^2} - 2 \quad \rightarrow u \in BV(\Omega) \;\; \text{(preserve discontinuities)}$$

Take home messages

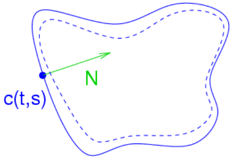
- PDE appear as a natural way to smooth images.
- When it is linear, a PDE (or equivalently the convolution) do not preserve edges.
- Nonlinearity is needed to preserve discontinuities (seen in all formulations)
- PDE may or may not derive from an optimisation problem.

1. Can we use PDEs to do some interesting image processing?
   - Definitions
   - From Gaussian filtering to the heat equation
2. Three solutions to go further
   - Solution 1: Make convolution "nonlinear"
   - Solution 2: Modify the heat equation
   - Solution 3: Define an optimisation problem
3. A success story: Levels-sets
4. Conclusion

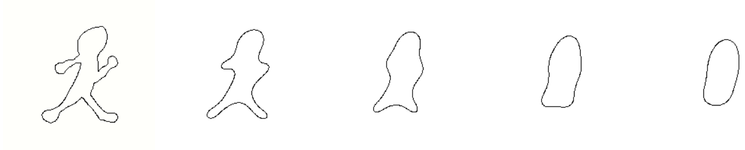
A curve evolves

- Lagrangian formulation

$$\begin{cases} \frac{\partial c}{\partial t}(t,q) = v(\kappa, \ldots)\, N \\ c(0,q) = c_0(q). \end{cases}$$



- Example with $v = \kappa$



A curve evolves... but what could be the interest?

A curve evolves... but what could be the interest?

Example: Segmentation

A curve evolves... but what could be the interest?

Example: Segmentation

A curve evolves... but what could be the interest?

Example: Segmentation

... How to formalise the problem? By defining optimisation problems!

*** IF TIME *** Active contours, Kass, Witkin etal [1987]

$$\inf_c J(c) = \underbrace{\int_a^b |\, c'(q)\,|^2 dq + \beta \int_a^b |\, c''(q)\,|^2 dq}_{\text{internal energy}} + \lambda \underbrace{\int_a^b g^2(\,|\nabla I(c(q))|)\, dq}_{\text{external energy}},$$

(g decreasing)

- – : $J(c)$ not intrinsic (depends on paramerisation)
- – : Because of regularity constraint, topology changes are impossible (restricted to a single convex object).
- – : Numerically, curve has to be initialised close to the object to segment

*** IF TIME *** What about a model with curvature?

$$\inf_c J_1(c) = \int_a^b |c'(q)|^2 \, dq + \lambda \int_a^b g^2(\,|\nabla I(c(q))|\,) dq$$

- – : Still not intrinsic.
- $+$: No high order terms in the Euler equation.
- $+$: It can be shown that curvature also decreases.

*** IF TIME *** Idea!

- Caselles, Kimmel, Kichenassamy [1995,..] :

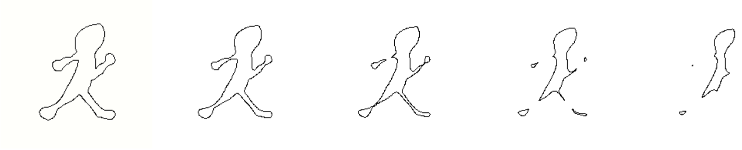
$$\inf_c J_2(c) = \int_a^b g(\,|\nabla I(c(q))|\,)\,|c'(q)|\,dq.$$

- $+$: Model is intrinsic!
- $+$: Equivalent to $\inf_c J_1$, Aubert, Blanc-Férraud [1999]
- $+$: Euler equation is a curve evolution :

$$\frac{\partial c}{\partial t} = (\kappa g - \langle \nabla g, N \rangle)\, N$$

Ok, but numerically, evolving a curve is not trivial

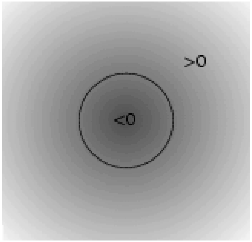
- Distribution of points won't stay homogeneous
- Stability problems
- Topology changes...

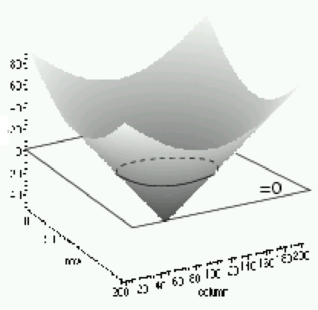


Idea: The *Level-Sets* method

Dervieux, Thomasset [1980], Osher, Sethian [1988]

- A curve seen as an isophote of a function



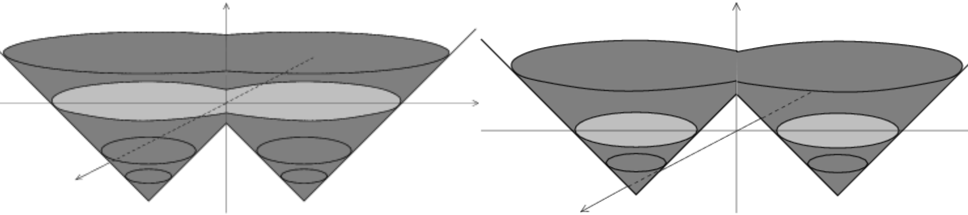


- Evolve curve is equivalent to evolve function

$$\begin{cases} \dfrac{\partial c}{\partial t} = v\, N, \\ c(0,q) = c_0(q). \end{cases} \Longrightarrow \begin{cases} \dfrac{\partial u}{\partial t} = v\, |\nabla u| \\ u(0,x) = u_0(x). \end{cases}$$

Many advantages

- Fixed system of coordinates
- Easy handling of topology changes



- Applicable in any dimension (think about surfaces in 3D!)
- Equation can be solved with suitable numerical schemes coming from hyperbolic equations.

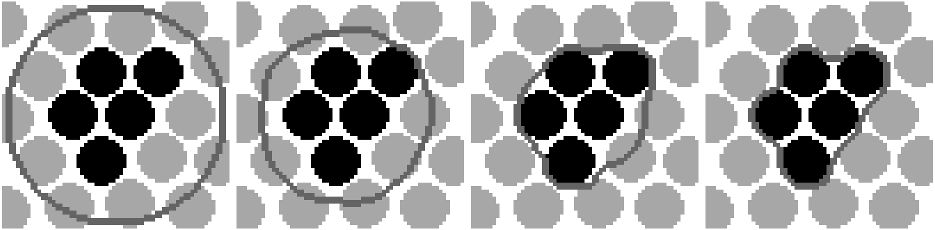
Exemple: Segmentation based on objects contours

Caselles, Kimmel, etal [1997], etc

$$\frac{\partial u}{\partial t} = g(|\nabla I|)\,|\nabla u|\,\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \alpha g(|\nabla I|)\,|\nabla u| + \langle \nabla g, \nabla u \rangle$$

Exemple: Segmentation based on regions

Chan, Vese [1999]



$$\inf_{i_1,i_2,c} F(i_1,i_2,c) = \mu|c| + \int_{\text{inside}(c)} |u_0 - i_1|^2 dx + \int_{\text{outside}(c)} |u_0 - i_2|^2 dx$$

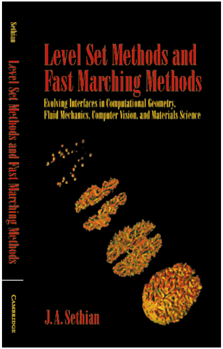
These are just two examples...

- Many other applications have been considered (e..g, textures, 3D objects in medical images, tracking of moving objects).
- New methods to improve speed (e.g., fast marching).
- Use shape priors.
- Keep function as a distance function across iterations to avoid periodic re-initialisation steps.
- etc.

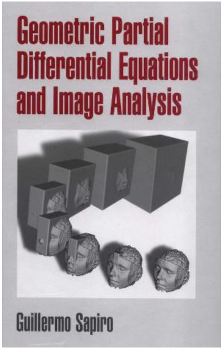
Take home messages

- PDE appear as a natural way to smooth images.
- When it is linear, a PDE (or equivalently the convolution) do not preserve edges.
- Nonlinearity is needed to preserve discontinuities (seen in all formulations)
- PDE may or may not derive from an optimisation problem.
- The notion of time evolution can be related to a notion of scale (in image restoration) but also to different aspects like a motion (in level-sets)
- (not shown here) Many theoretical results allow to prove if your problem is well defined or not.
- Giving formulations in a continuous setting offers high intuitions and discretisation aspects only come when simulations are needed.
- There has been a high activity in this area in [1990–2010] with lots of papers!

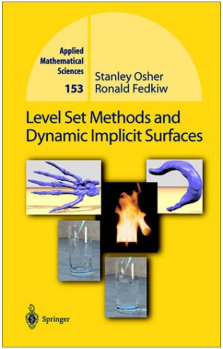
If you want to learn more



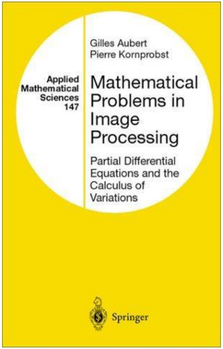
1999
Sethian



2001
Sapiro



2002
Osher, Fedwik



2002, 2006, *2014*
Aubert, Kornprobst

Thank you!

http://www-sop.inria.fr/members/Pierre.Kornprobst