

Partie 3: Restauration d'images, Minimisation de fonctionnelles

Pierre Maurel

Visages, IRISA/INRIA

pierre.maurel@irisa.fr

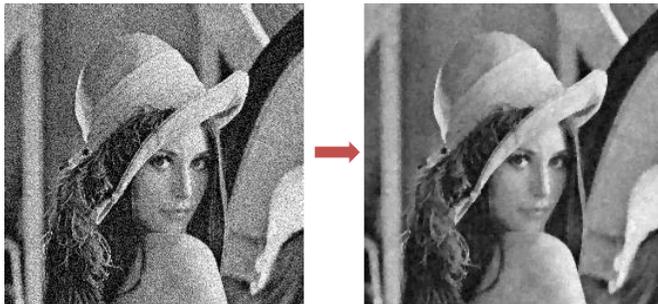
<http://www.normalesup.org/~pmaurel/IMA/>

Restauration d'images

- le plus vieux problème en traitement d'images et toujours une étape de pré-traitement nécessaire pour beaucoup d'applications en analyse d'images
- Dans son processus d'acquisition, une image peut subir des dégradations (mais également pendant sa transmission ou son enregistrement, par exple)
- Ces dégradations peuvent être dues
 - ① aux capteurs
 - le flou du à l'optique et à l'intégration de la lumière sur chaque pixel
 - bruit de quantification (2ⁿ niveaux de gris)
 - ② aux conditions de prise de vues
 - flou de bougé
 - perturbation atmosphérique
- supprimer ou diminuer les effets d'une telle détérioration : **restauration**

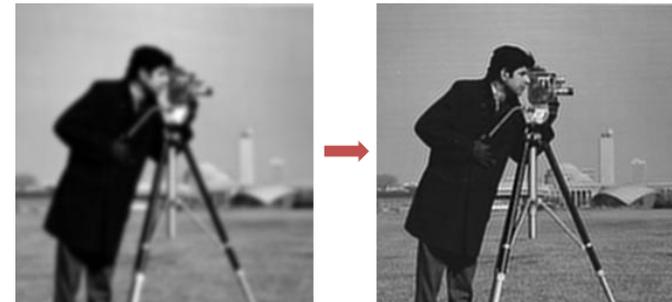
Restauration d'images

Débruitage



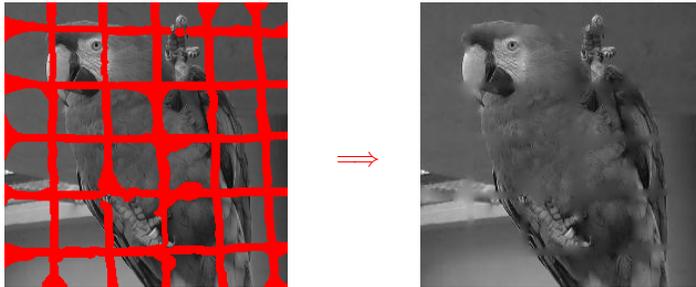
Restauration d'images

Déconvolution



Restauration d'images

Inpainting (ou désocclusion)



6 / 82

Restauration d'images

Exemple du débruitage



- On veut débruiter → c'est à dire ?
- Pourquoi dit-on que cette image est bruitée ? → trop "irrégulière"
- débruiter = faire disparaître les irrégularités ?
- Besoin de "modéliser" (traduire en termes mathématiques) :
 - irrégulier = ?
 - régulariser = ?

7 / 82

Restauration d'images

- on note f_0 l'image originale **inconnue** (image idéale sans détérioration)
- on note f l'image observée (f est donc une version dégradée de f_0)

Modélisation de la dégradation subie

on suppose que :

$$f = \mathcal{R}f_0 + \eta$$

où :

- \mathcal{R} est un opérateur représentant la dégradation déterministe
- η est une image "aléatoire" : le bruit

8 / 82

Restauration d'images : exemple de modélisation de la dégradation subie



f

$\mathcal{R}f_0$

η



$\mathcal{R}f_0$

f_0

9 / 82

Modélisation de la dégradation subie en fonction de l'application souhaitée

- Débruitage seul : $\mathcal{R} = \text{Id}$
- Déconvolution : $\mathcal{R} = \text{convolution avec une gaussienne}$

$$\mathcal{R} \left(\text{Image} \right) = \text{Image}$$

- Inpainting : $\mathcal{R} = \text{opérateur de masquage}$

$$\mathcal{R} \left(\text{Image} \right) = \text{Image avec masque}$$

10 / 82

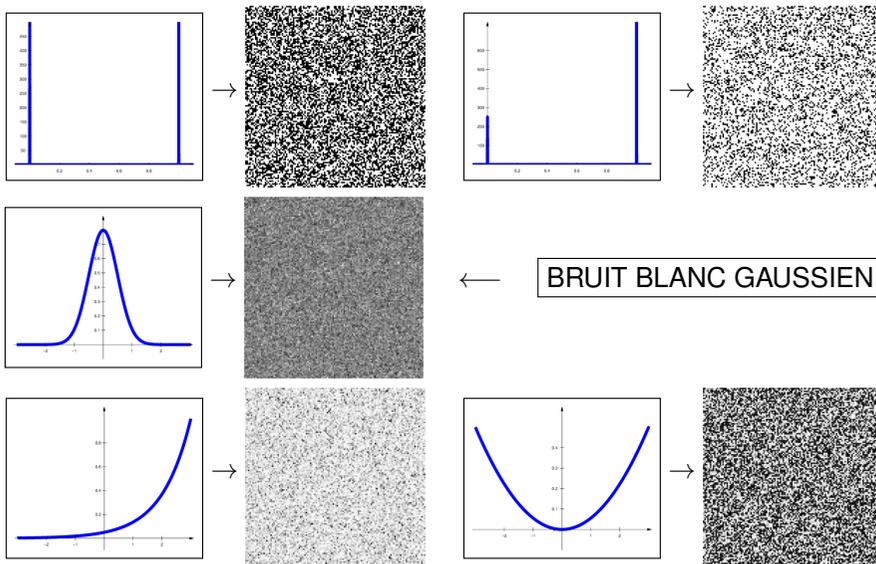
Modélisation du bruit

$$f = f_0 + \eta$$

- η est une image aléatoire donc inconnue.
- on fait des hypothèses sur les propriétés de cette image
- Bruit blanc : réalisation d'un processus aléatoire dans lequel toutes les fréquences ont la même importance
- Exemple : chaque pixel est "tiré" de manière indépendante des autres selon une loi de probabilité

11 / 82

Modélisation du bruit



12 / 82

Modélisation du bruit

Bruit blanc gaussien

- On note η un bruit blanc gaussien d'écart-type correspondant à l'image en indice.

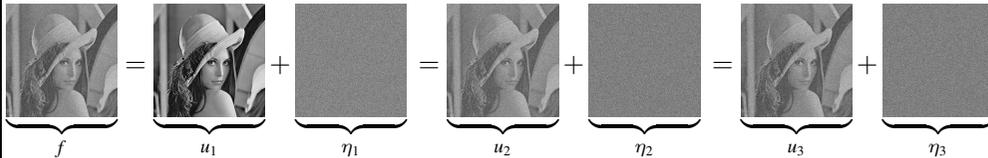
$$\text{Image} + \eta = \text{Image bruitée}$$

14 / 82

Restauration d'images

Pour une image f donnée, le problème est donc de reconstruire f_0 sachant que l'on a $f = \mathcal{R} f_0 + \eta$

- On cherche donc une image u vérifiant : $f = \mathcal{R} u + \eta$
 - on connaît f qui est l'image de départ et on fixe \mathcal{R} (en fonction de ce qu'on veut faire)
 - on ne connaît pas η mais on connaît des propriétés (exple : bruit blanc gaussien)
- Il s'agit d'un problème inverse mal posé. Il existe une infinité de solution.

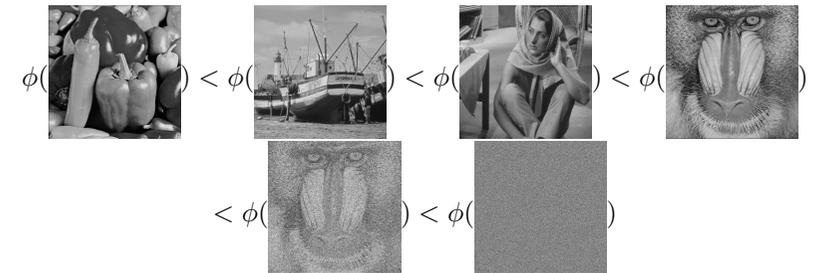


- Quelle solution choisir ?

15/82

Restauration d'images : régularisation

- S'il y a une infinité de solutions, c'est que notre modèle est trop "faible"
- On doit donc rajouter des informations "a priori"
- exemple : on sait que la vraie image f_0 était "lisse" (en tout cas plus lisse que l'image dégradée)
- on cherche une approximation de f_0 vérifiant certaines hypothèses de régularité.
- besoin d'un "score de régularité" ϕ :



16/82

Restauration d'images : régularisation

- Critère de régularisation ? souvent basé sur la norme du gradient

$$\phi(u) = \int_{\Omega} \|\nabla u(x)\|^2 d\Omega$$

- On cherche donc maintenant une image u telle que

$\mathcal{R} u$ soit proche de f **ET QUE** u soit régulière.

- Modélisation possible :

- $\mathcal{R} u$ proche de f $\iff \|f - \mathcal{R} u\|_{\mathcal{L}^2}^2 = \int_{\Omega} (f(x) - \mathcal{R} u(x))^2 d\Omega$ **petit**

- u régulière $\iff \|\nabla u\|_{\mathcal{L}^2}^2 = \int_{\Omega} \|\nabla u(x)\|^2 d\Omega$ **petit**

17/82

Restauration d'images : régularisation

- On cherche donc u tel que $\|f - \mathcal{R} u\|_{\mathcal{L}^2}^2$ et $\|\nabla u\|_{\mathcal{L}^2}^2$ soit tous les deux "petits"
- Une possibilité : minimiser la somme des deux
- Tikhonov 1977 : on cherche

$$\inf_u J(u) = \inf_u \underbrace{\int_{\Omega} (f(x) - \mathcal{R} u(x))^2 d\Omega}_{\text{terme de fidélité, attache aux données}} + \lambda \underbrace{\int_{\Omega} \|\nabla u(x)\|^2 d\Omega}_{\text{régularisation}}$$

18/82

Régularisation de Tikhonov

$$\inf_u J(u) = \inf_u \underbrace{\int_{\Omega} (f(x) - \mathcal{R}u(x))^2 d\Omega}_{\text{terme de fidélité, attache aux données}} + \lambda \underbrace{\int_{\Omega} \|\nabla u(x)\|^2 d\Omega}_{\text{régularisation}}$$

- λ est un paramètre de poids positif qu'il faut ajuster en fonction de la quantité de bruit estimée.
- Exemple : si $\mathcal{R} = \text{Id}$ (débruitage), à quoi correspond le cas $\lambda = 0$?
il ne reste plus que le premier terme $\inf_u \int_{\Omega} (f(x) - u(x))^2 d\Omega \Rightarrow u = f$
- Si λ est très grand ? seul le second terme à un effet et donc on cherche $\inf_u \int_{\Omega} \|\nabla u(x)\|^2 d\Omega \Rightarrow u = \text{cste}$ (ne dépend plus de la donnée f)
- On choisit λ proportionnellement au σ estimé du bruit

Comment minimiser ?

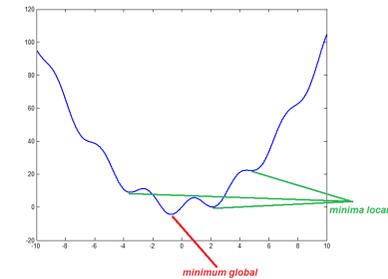
19 / 82

Le problème général

$$\text{Trouver } \inf_{x \in X, x \in \mathcal{C}} E(x)$$

Trois éléments

- un espace d'états X (connexe ? convexe ? borné ? discret ?)
- une fonction objectif (énergie) $E : X \rightarrow \mathbb{R}$ (linéaire ? quadratique ? dérivable ? convexe ?)
- des contraintes $\mathcal{C} \subset X$ (égalités ? inégalités ? linéaires ?)



21 / 82

Minimisation : exemples simples

$$\text{Trouver } \inf_{x \in X, x \in \mathcal{C}} E(x)$$

1 $X = \mathbb{R}$, $\mathcal{C} = \mathbb{R}$, $E(x) = (x - a)^2$

$$\inf_{x \in X \cap \mathcal{C}} E(x) = 0$$

$$\arg \inf_{x \in X \cap \mathcal{C}} E(x) = a$$

2 $X = \mathbb{R}$, $\mathcal{C} = \{y \in \mathbb{R} \mid y \geq 1\}$, $E(x) = (x - 1)^3$

$$\inf_{x \in X \cap \mathcal{C}} E(x) = 0$$

$$\arg \inf_{x \in X \cap \mathcal{C}} E(x) = 1$$

3 $X = \mathbb{R}$, $\mathcal{C} = \mathbb{R}$, $E(x) = (x - a)^2 + (x - b)^2$

$$\inf_{x \in X \cap \mathcal{C}} E(x) = \frac{(b - a)^2}{2}$$

$$\arg \inf_{x \in X \cap \mathcal{C}} E(x) = \frac{(a + b)}{2}$$

22 / 82

Principales méthodes

- En général : plus compliqué
- Méthodes locales : on part d'un point x_0 et on construit une suite de points jusqu'à ce qu'on décide de s'arrêter
 - descentes de gradient et méthodes dérivées
 - recuit simulé
 - ...
- Méthodes globales : on cherche directement le minimum global de la fonction (souvent pour type particulier de fonctions)
 - moindre carré
 - simplexe
 - algorithmes génétiques
 - ...

23 / 82

Descente de gradient

- **Problème :**

On veut minimiser $E(x) : \mathbb{R}^n \rightarrow \mathbb{R}$

On cherche donc $x_0 \in \mathbb{R}^n$ tel que $E(x_0) = \inf_{x \in \mathbb{R}^n} E(x)$

- **Idée générale :** progresser par étapes proportionnelles à l'opposé du gradient. L'opposé du gradient donne en effet la direction *de plus grande descente*.

- si $n = 1$, on a $E(x+h) = E(x) + hE'(x) + o(h)$

- donc $E(x - \varepsilon E'(x)) = E(x) - \varepsilon E'(x)^2 + o(\varepsilon)$

- donc si ε est suffisamment petit, on a $E(x - \varepsilon E'(x)) \leq E(x)$

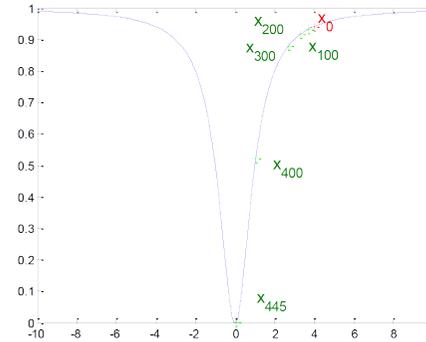
- on a égalité si et seulement si $E'(x) = 0$

25 / 82

Illustration de la descente de gradient 1D

DEMO MATLAB

$$E(x) = 1 - \frac{1}{1+x^2} \Rightarrow E'(x) = \frac{2x}{(1+x^2)^2}$$



$\varepsilon = 0.1$

$$x_{n+1} = x_n - \varepsilon E'(x_n)$$

$$\frac{\partial x(t)}{\partial t} = -E'(x(t)) \quad \leftarrow \text{EDP!!}$$

critère d'arrêt : $|x_{n+1} - x_n| < 10^{-5}$?

$$x_{final} = 4.96 \cdot 10^{-4}$$

26 / 82

Descente de gradient

- de la même manière pour $n \geq 1$:

$$E(x+h) = E(x_1+h_1, \dots, x_n+h_n) = E(x) + h_1 \frac{\partial E}{\partial x_1}(x) + \dots + h_n \frac{\partial E}{\partial x_n}(x) + o(\|h\|)$$

$$E(x+h) = E(x) + \langle h | \nabla E(x) \rangle + o(\|h\|)$$

$$E(x - \varepsilon \nabla E(x)) = E(x) - \varepsilon \|\nabla E(x)\|^2 + o(\varepsilon)$$

- On part d'un $x_0 \in \mathbb{R}^n$ et on avance par pas successifs de $-\varepsilon \nabla E(x)$ jusqu'à ce qu'on n'avance plus assez

DEMO MATLAB

27 / 82

Descente de gradient : un exemple concret

- On cherche x, y et z qui minimise la fonction (ou énergie) :

$$E(x, y, z) = (xy - z + 1)^2 + (x + yz - 2)^2$$

- $\nabla E(x, y, z) = \begin{pmatrix} 2(xy - z + 1)y + 2(x + yz - 2) \\ 2(xy - z + 1)x + 2(x + yz - 2)z \\ -2(xy - z + 1) + 2(x + yz - 2)y \end{pmatrix}$

- On choisit $\varepsilon > 0$ et $(x_0, y_0, z_0) \in \mathbb{R}^3$ et on itère :

$$\begin{cases} x_{n+1} = x_n - \varepsilon \left(2(x_n y_n - z_n + 1)y_n + 2(x_n + y_n z_n - 2) \right) \\ y_{n+1} = y_n - \varepsilon \left(2(x_n y_n - z_n + 1)x_n + 2(x_n + y_n z_n - 2)z_n \right) \\ z_{n+1} = z_n - \varepsilon \left(-2(x_n y_n - z_n + 1) + 2(x_n + y_n z_n - 2)y_n \right) \end{cases}$$

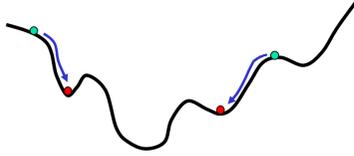
jusqu'à ce que les valeurs ne bougent plus assez.

DEMO MATLAB

28 / 82

Descente de gradient

- On ne converge que vers un minimum local de la fonction (qui dépend donc du point de départ).



- OK si pas de minima locaux : cas particulier → fonctions convexes
- Algorithme très lent dans les régions “plates”
- améliorations possibles : recherche du pas optimal (critères de Wolfe, règles d’Armijo, ...)
- méthodes dérivées de la descente de gradient de base :
 - Méthode de Newton (second ordre)
 - gradient conjugué
 - Broyden-Fletcher-Goldfarb-Shanno (BFGS), Quasi-Newton
 - ...

29 / 82

Minimisation en dimension infinie

- dans notre cas, dimension infinie : fonction de fonctions (= images)
- On ne peut donc pas définir de “gradient”, mais on va faire des choses très similaires
- On cherche une fonction u telle que $J(u)$ soit minimal (où J est notre énergie à minimiser)
- L’énergie est minimale en u quand elle est insensible aux faibles variations $\delta u = \varepsilon \phi$, avec ϕ une fonction quelconque. (“équivalent” de dérivée nulle dans toutes les directions)

31 / 82

Minimisation en dimension infinie

Dérivée de Gâteaux

- La **dérivée de Gâteaux** de J au point u dans la direction de ϕ est définie ainsi

$$\lim_{\varepsilon \rightarrow 0} \frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon}$$

- Dérivée directionnelle
- Lorsqu’on cherche les extremums de J on cherche donc u tel que

$$\lim_{\varepsilon \rightarrow 0} \frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = 0 \quad , \text{ quelque soit } \phi$$

- Comment calculer cette dérivée de Gâteaux ? Pour en faire quoi ?

32 / 82

Minimisation de fonctionnelles

Définition

Une fonction J sera appelée **fonctionnelle** si elle prend en argument une fonction et qu’elle peut s’écrire sous la forme d’une intégrale :

$$J(u) = \int_{\Omega} F\left(\mathbf{x}, u(\mathbf{x}), \frac{\partial u}{\partial x_i}(\mathbf{x}), \dots, \frac{\partial^k u}{\partial x_i^k}(\mathbf{x})\right) d\Omega$$

- F : la fonction de définition de la fonctionnelle J
- $\mathbf{x} = (x_1, \dots, x_n)$: le vecteur des variables
- u : la fonction argument de J , $u : \mathbb{R}^n \rightarrow \mathbb{R}$
- Ω : le domaine de définition, $\Omega \subset \mathbb{R}^n$ (support de l’image)

Remarque : la notation $\int_{\Omega} f(\mathbf{x}) d\Omega$ signifie “intégrale sur le domaine Ω ”. Dans le cas d’une image plane rectangle ($n = 2$) on a $\int_{\Omega} f(\mathbf{x}) d\Omega = \int_x \int_y f(x, y) dy dx$.

On cherche u qui minimise $J(u)$, on veut donc résoudre : $\inf_u J(u)$.

33 / 82

Minimisation de fonctionnelles

- On considère les fonctionnelles du type :

$$J(u) = \int_{\Omega} F(x, u(x), \frac{\partial u}{\partial x_1}(x), \dots, \frac{\partial u}{\partial x_n}(x)) d\Omega = \int_{\Omega} F(x, u(x), \nabla u(x)) d\Omega$$

- exemple : restauration, régularisation de Tikhonov avec $\mathcal{R} = \text{Id}$ (débruitage)

$$\begin{aligned} J(u) &= \int_{\Omega} (u(x) - f(x))^2 d\Omega + \lambda \int_{\Omega} \|\nabla u(x)\|^2 d\Omega \\ &= \int_{\Omega} F(x, u(x), \nabla u(x)) d\Omega \end{aligned}$$

avec

$$F(a, b, \xi) = (b - f(a))^2 + \lambda \|\xi\|^2 = (b - f(a))^2 + \lambda(\xi_1^2 + \dots + \xi_n^2)$$

(attention $\xi \in \mathbb{R}^n$!)

34/82

Minimisation de fonctionnelles

- La fonctionnelle $J(u)$ est extrémale lorsqu'elle est insensible à de petites variations de u du type

$$\delta u = \varepsilon \phi$$

où ϕ est une fonction quelconque.

- si u est le minimum de $J(u)$ on a donc :

$$\lim_{\varepsilon \rightarrow 0} \frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = 0, \quad \text{pour tout } \phi$$

(dérivée de Gâteaux de J au point u dans la direction de ϕ)

- on a :

$$J(u + \varepsilon \phi) = \int_{\Omega} F(x, u(x) + \varepsilon \phi(x), \frac{\partial(u + \varepsilon \phi)}{\partial x_1}(x), \dots, \frac{\partial(u + \varepsilon \phi)}{\partial x_n}(x)) d\Omega$$

- développement de Taylor.

35/82

Minimisation de fonctionnelles

$$\begin{aligned} J(u + \varepsilon \phi) &= \int_{\Omega} F(x, u(x) + \varepsilon \phi(x), \frac{\partial(u + \varepsilon \phi)}{\partial x_1}(x), \dots, \frac{\partial(u + \varepsilon \phi)}{\partial x_n}(x)) d\Omega \\ &= J(u) + \int_{\Omega} \left[\varepsilon \phi(x) \frac{\partial F}{\partial b} + \varepsilon \frac{\partial \phi}{\partial x_1} \frac{\partial F}{\partial \xi_1} + \dots + \varepsilon \frac{\partial \phi}{\partial x_n} \frac{\partial F}{\partial \xi_n} \right] d\Omega + o(\varepsilon) \end{aligned}$$

avec :

- $F(a, b, \xi) = F(a, b, \xi_1, \dots, \xi_n)$
- $\frac{\partial F}{\partial b}$: dérivée par rapport à la seconde variable (qui sera égale à $u(x)$ dans notre cas)
- $\frac{\partial F}{\partial \xi_i}$: dérivée par rapport à la i -ème composante de la troisième variable (qui sera égale à $\nabla u(x)$ dans notre cas).

Donc

$$\frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = \int_{\Omega} \phi(x) \frac{\partial F}{\partial b} d\Omega + \sum_{i=1}^n \int_{\Omega} \frac{\partial \phi}{\partial x_i} \frac{\partial F}{\partial \xi_i} d\Omega + o(\varepsilon)$$

36/82

Minimisation de fonctionnelles

- La condition devient donc :

$$J(u) \text{ minimum} \Leftrightarrow \int_{\Omega} \phi(x) \frac{\partial F}{\partial b} d\Omega + \sum_{i=1}^n \int_{\Omega} \frac{\partial \phi}{\partial x_i} \frac{\partial F}{\partial \xi_i} d\Omega = 0 \quad \text{pour tout } \phi$$

- On utilise la **formule de Green** qui généralise l'intégration par partie :

$$\int_{\Omega} \frac{\partial f}{\partial x_i} d\Omega = \int_{\Gamma} f n_i d\Gamma$$

avec : Γ la courbe-frontière de Ω et \mathbf{n} la normale au bord Γ .

- On a aussi : $\int_{\Omega} \frac{\partial}{\partial x_i} \left(\phi \frac{\partial F}{\partial \xi_i} \right) d\Omega = \int_{\Omega} \frac{\partial \phi}{\partial x_i} \frac{\partial F}{\partial \xi_i} d\Omega + \int_{\Omega} \phi \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) d\Omega$

$$\text{Donc : } \int_{\Omega} \frac{\partial \phi}{\partial x_i} \frac{\partial F}{\partial \xi_i} d\Omega = \int_{\Gamma} \phi \frac{\partial F}{\partial \xi_i} n_i d\Gamma - \int_{\Omega} \phi \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) d\Omega$$

37/82

Équation d'Euler-Lagrange

- La condition devient donc

$$J(u) \text{ minimum} \Leftrightarrow \int_{\Omega} \left(\frac{\partial F}{\partial b} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) \right) \phi \, d\Omega + \int_{\Gamma} \left(\sum_{i=1}^n \frac{\partial F}{\partial \xi_i} \mathbf{n}_i \right) \phi \, d\Gamma = 0$$

- cette relation doit être vérifiée pour toute fonction ϕ et en particulier pour celles à valeurs nulles sur Γ
- on a donc :

$$\frac{\partial F}{\partial b}(x, u(x), \nabla u(x)) - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i}(x, u(x), \nabla u(x)) \right) = 0$$

équation (**EDP !!**) connue sous le nom d'**équation d'Euler-Lagrange** associée au problème $\inf_u J(u)$.

- remarque : le second terme (sur Γ) doit aussi être nul et on retrouve donc les conditions aux bords dites de Neumann.

38 / 82

Équation d'Euler-Lagrange

L'équation d'Euler-Lagrange transforme un problème de **minimisation** en une **équation aux dérivées partielles**

$$u \text{ est minimum de } J(u) = \int_{\Omega} F(x, u(x), \nabla u(x)) \, d\Omega \quad \leftarrow \text{Pb d'optimisation}$$

\Leftrightarrow

$$u \text{ vérifie } \frac{\partial F}{\partial b}(x, u(x), \nabla u(x)) - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i}(x, u(x), \nabla u(x)) \right) = 0 \quad \leftarrow \text{EDP}$$

39 / 82

Exemple : débruitage par régularisation de Tikhonov

$$\begin{aligned} J(u) &= \int_{\Omega} (u(x) - f(x))^2 \, d\Omega + \lambda \int_{\Omega} \|\nabla u(x)\|^2 \, d\Omega \\ &= \int_{\Omega} F(x, u(x), \nabla u(x)) \, d\Omega \end{aligned}$$

$$\text{avec } F(a, b, \xi) = (b - f(a))^2 + \lambda \|\xi\|^2 = (b - f(a))^2 + \lambda(\xi_1^2 + \xi_2^2)$$

Équation d'Euler-Lagrange associée ?

$$\begin{cases} \frac{\partial F}{\partial b}(a, b, \xi) = 2(b - f(a)) \\ \frac{\partial F}{\partial \xi_1}(a, b, \xi) = 2\lambda \xi_1 \\ \frac{\partial F}{\partial \xi_2}(a, b, \xi) = 2\lambda \xi_2 \end{cases} \implies \begin{cases} \frac{\partial F}{\partial b}(x, u(x), \nabla u(x)) = 2(u(x) - f(x)) \\ \frac{\partial F}{\partial \xi_1}(x, u(x), \nabla u(x)) = 2\lambda \frac{\partial u(x)}{\partial x_1} \\ \frac{\partial F}{\partial \xi_2}(x, u(x), \nabla u(x)) = 2\lambda \frac{\partial u(x)}{\partial x_2} \end{cases}$$

40 / 82

Exemple : débruitage par régularisation de Tikhonov

Équation d'Euler-Lagrange associée

$$\begin{aligned} \frac{\partial F}{\partial b}(x, u(x), \nabla u(x)) - \frac{\partial}{\partial x_1} \left(\frac{\partial F}{\partial \xi_1}(x, u(x), \nabla u(x)) \right) - \frac{\partial}{\partial x_2} \left(\frac{\partial F}{\partial \xi_2}(x, u(x), \nabla u(x)) \right) &= 0 \\ = 2(u(x) - f(x)) - \frac{\partial}{\partial x_1} \left(2\lambda \frac{\partial u(x)}{\partial x_1} \right) - \frac{\partial}{\partial x_2} \left(2\lambda \frac{\partial u(x)}{\partial x_2} \right) &= 0 \end{aligned}$$

$$u(x) - f(x) - \lambda \left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} \right) = 0$$

$$\Rightarrow \boxed{u(x) - f(x) - \lambda \Delta u = 0} \quad \leftarrow \text{EDP !!} \quad (1)$$

On a donc montré que :

$$\boxed{u \text{ minimise } J(u)} \iff \boxed{u \text{ vérifie (1)}}$$

41 / 82

Équation d'Euler-Lagrange et descente de gradient

- **Mais** : l'équation d'Euler-Lagrange n'a souvent pas de solution explicite
- on définit donc une séquence minimisante en introduisant une variabilité temporelle (\leftrightarrow descente de gradient)

• on pose :
$$\frac{\partial u(x, t)}{\partial t} = - \left(\frac{\partial F}{\partial b} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) \right)$$

et on cherche un état stationnaire, c'est à dire : $\frac{\partial u}{\partial t} = 0$.

- En effet, on a montré que

$$\frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = \int_{\Omega} \left(\frac{\partial F}{\partial b} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) \right) \phi d\Omega + o(1)$$

donc si on choisit $\phi = - \left(\frac{\partial F}{\partial b} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) \right)$ on obtient :

$$J(u + \varepsilon \phi) = J(u) - \varepsilon \int_{\Omega} \left(\frac{\partial F}{\partial b} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) \right)^2 d\Omega + o(\varepsilon) \leq J(u)$$

42 / 82

Comparaison dimension finie / infinie

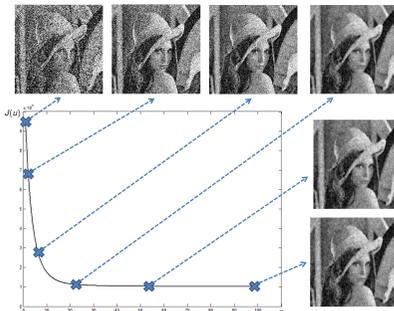
Dim. finie ($x \in \mathbb{R}^n$)	Dimension infinie ($u : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$)	
$x_0 = \arg \inf_{x \in \mathbb{R}^n} E(x)$	$u_0 = \arg \inf_u J(u) = \int_{\Omega} F(x, u(x), \nabla u(x)) d\Omega$	Minimisation
\downarrow	\downarrow	
$E'(x_0) = 0$	$\frac{\partial F}{\partial b}(x, u_0(x), \nabla u_0(x)) - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i}(x, u_0(x), \nabla u_0(x)) \right) = 0$	EDP
\downarrow	\downarrow	
$\frac{\partial x(t)}{\partial t} = -E'(x(t))$	$\frac{\partial u(x, t)}{\partial t} = - \left(\frac{\partial F}{\partial b} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) \right)$	Desc. de gradient
\downarrow	\downarrow	
$x_{n+1} = x_n - \Delta t E'(x_n)$	$u_{i,j}^{n+1} = u_{i,j}^n - \Delta t \left(\frac{\partial F}{\partial b} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) \right)$	Discrétisation

43 / 82

Diffusion par régularisation de Tikhonov

$$\frac{\partial u}{\partial t}(x, y, t) = f(x) - u(x, y, t) + \lambda \Delta u(x, y, t)$$

- Ici l'énergie $J(u)$ est convexe \rightarrow pas de minima locaux \rightarrow résultat final indépendant de l'initialisation
- \rightarrow on prend $u_{i,j}^0 = f_{i,j}$ pour accélérer la convergence

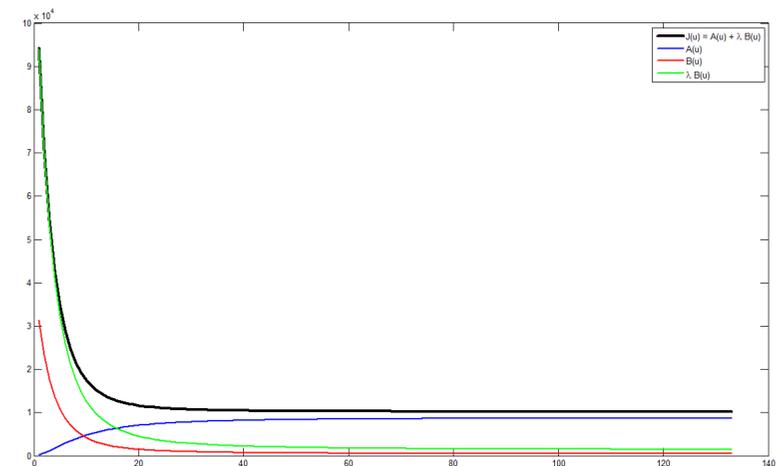


DEMO MATLAB

45 / 82

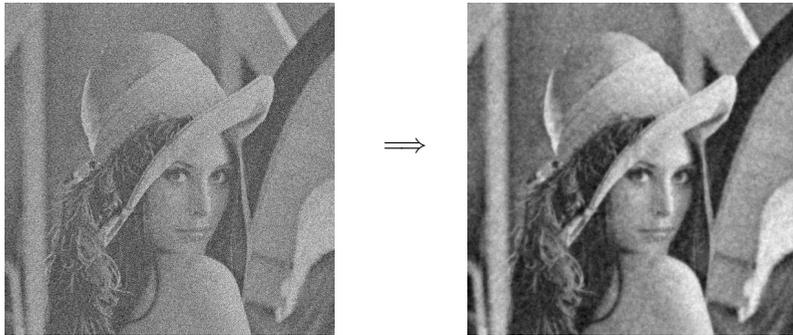
Diffusion par régularisation de Tikhonov

$$A(u) = \int_{\Omega} (u(x) - f(x))^2 d\Omega \quad \text{et} \quad B(u) = \int_{\Omega} \|\nabla u(x)\|^2 d\Omega$$



46 / 82

Diffusion par régularisation de Tikhonov



Remarque : importance du paramètre λ



$\lambda = 0.25$

$\lambda = 3$

$\lambda = 10$

$\lambda = 15$

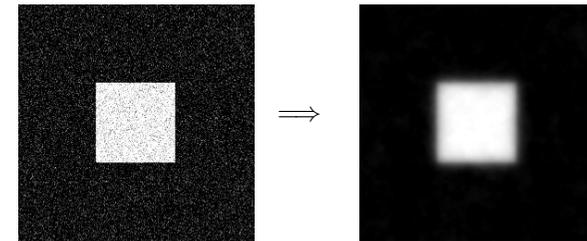
47 / 82

Défaut de la diffusion linéaire

- problème : l'opérateur Laplacien lisse de manière **isotrope**
- ne préserve pas les arêtes, les contours
- on peut aussi dire, de manière équivalente, que dans l'énergie :

$$\int_{\Omega} (u(x) - f(x))^2 d\Omega + \lambda \int_{\Omega} \|\nabla u(x)\|^2 d\Omega$$

la norme \mathcal{L}^2 du gradient permet de supprimer le bruit mais pénalise trop les gradients sur les contours.



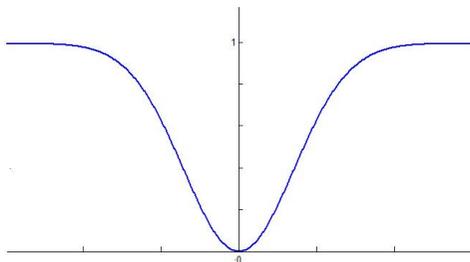
49 / 82

Diffusion non linéaire

$$J(u) = \underbrace{\int_{\Omega} (u(x) - f(x))^2 d\Omega}_{\text{terme de fidélité, attache aux données}} + \lambda \underbrace{\int_{\Omega} \Psi(\|\nabla u(x)\|) d\Omega}_{\text{régularisation}}$$

où Ψ est une fonction moins pénalisante pour les gradients élevés que la fonction quadratique.

Exemple :



50 / 82

Diffusion non linéaire : équation d'Euler-Lagrange

$$\frac{\partial F}{\partial b}(x, u(x), \nabla u(x)) - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i}(x, u(x), \nabla u(x)) \right) = 0$$

avec $F(a, b, \xi) = (b - f(a))^2 + \lambda \Psi(\sqrt{\xi_1^2 + \xi_2^2})$

$$\frac{\partial F}{\partial \xi_1} = \lambda \frac{\xi_1 \Psi'(\sqrt{\xi_1^2 + \xi_2^2})}{\sqrt{\xi_1^2 + \xi_2^2}} = \lambda \frac{\xi_1 \Psi'(\|\xi\|)}{\|\xi\|}$$

$$\begin{aligned} \text{donc } \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\frac{\partial F}{\partial \xi_i} \right) &= \frac{\partial}{\partial x} \left(\lambda \frac{\partial u}{\partial x} \frac{\Psi'(\|\nabla u\|)}{\|\nabla u\|} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial u}{\partial y} \frac{\Psi'(\|\nabla u\|)}{\|\nabla u\|} \right) \\ &= \lambda \operatorname{div} \left(\frac{\Psi'(\|\nabla u\|)}{\|\nabla u\|} \nabla u \right) \end{aligned}$$

$$\text{et finalement } 2(u - f) - \lambda \operatorname{div} \left(\frac{\Psi'(\|\nabla u\|)}{\|\nabla u\|} \nabla u \right) = 0$$

- Exo : si $\Psi(x) = x^2$? on obtient bien le laplacien

51 / 82

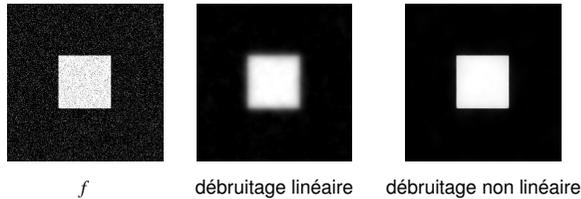
Diffusion non linéaire

Descente de gradient :

$$\frac{\partial u}{\partial t} = 2(f - u) + \lambda \operatorname{div} \left(\frac{\Psi'(\|\nabla u\|)}{\|\nabla u\|} \nabla u \right)$$

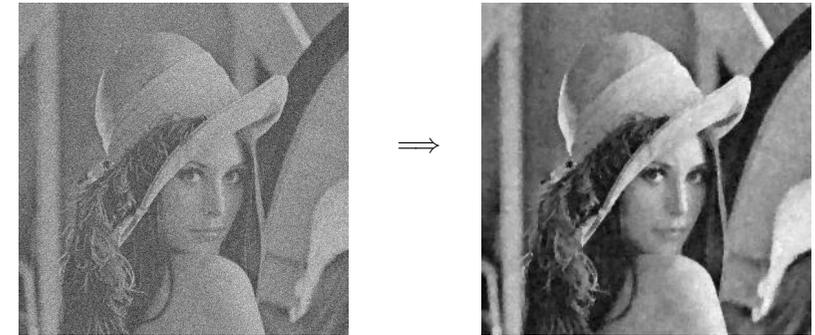
- Exemple : $\Psi(x) = \sqrt{\varepsilon + x^2} \xrightarrow{\varepsilon \rightarrow 0} |x|$
- $\int \Psi(\|\nabla u\|)$ approxime la variation totale $\int \|\nabla u\|$ si ε est petit (mais est différentiable en 0)
- $\Psi'(x) = \frac{x}{\sqrt{\varepsilon + x^2}}$ et $\frac{\Psi'(\|\nabla u\|)}{\|\nabla u\|} \nabla u = \frac{\nabla u}{\sqrt{\varepsilon + \|\nabla u\|^2}}$

DEMO MATLAB

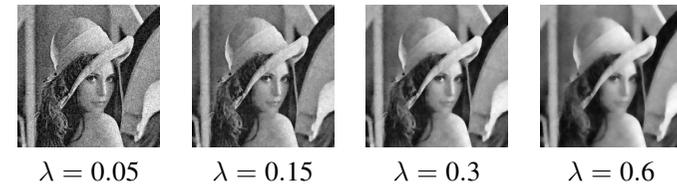


52 / 82

Diffusion non linéaire avec terme de rappel



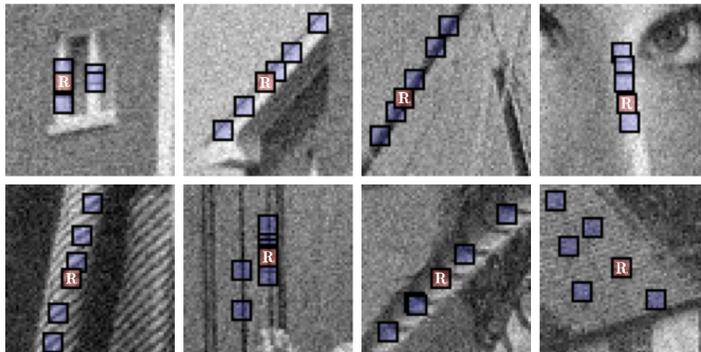
Remarque : importance du paramètre λ .



53 / 82

Autres méthodes de débruitage : NL-means / BM3D

- Non-Local Means : Buades et al., 2005 (CVPR)
- BM3D : Dabov et al., 2006 (SPIE)
- Idée générale : chaque valeur est remplacée par une moyenne pondérée de pixels choisis pour leur texture similaire



54 / 82

Autres méthodes de débruitage : NL-means / BM3D

Principe général des NL-means

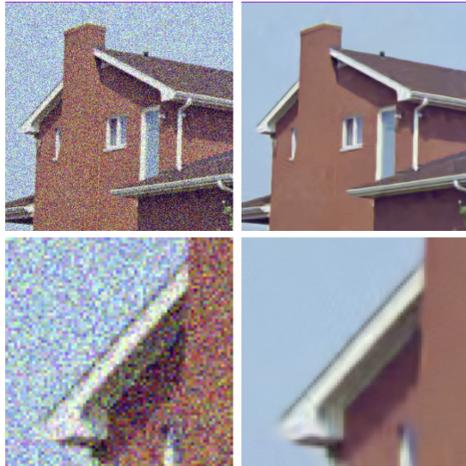
- Pour chaque pixel x :
 - On extrait le "patch" $p(x)$
 - Pour chaque voisin y (dans une fenêtre autour de x)
 - On extrait le patch $p(y)$
 - On calcule le poids : $\exp\left(\frac{-\|p(x)-p(y)\|^2}{2\sigma^2}\right)$
 - On remplace l'ancienne valeur en x , $v(x)$, par la moyenne des $v(y)$ pondérée par les poids ci-dessus

BM3D (reconnu comme "l'état de l'art" en débruitage)

- Reprend l'idée des NL-means pour prendre en compte les corrélations **inter**-patches
- Ajoute un traitement basé "ondelettes" pour prendre en compte les corrélations **intra**-patches

55 / 82

Autres méthodes de débruitage : NL-means / BM3D



K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," Proc. SPIE Electronic Imaging, January 2006.

<http://www.cs.tut.fi/~foi/GCF-BM3D/>

56 / 82

Comparaison



bruitée

originale



Linéaire
(SNR = 19.92 dB)

Non-Linéaire
(SNR = 21.24 dB)

BM3D
(SNR = 23.05 dB)

SNR = "signal-to-noise ratio". Rapport signal sur bruit : mesure de l'écart entre l'image originale sans bruit f_0 et l'image débruitée f .
$$\text{SNR}(f_0, f) = 20 \log_{10} \frac{\|f_0\|_{L^2}}{\|f_0 - f\|_{L^2}}$$

57 / 82

Si $\mathcal{R} \neq \text{Id}$

$$\inf_u J(u) = \underbrace{\inf_u \int_{\Omega} (\mathcal{R}u(x) - f(x))^2 d\Omega}_{\text{terme de fidélité, attache aux données}} + \lambda \underbrace{\int_{\Omega} \Psi(\|\nabla u(x)\|) d\Omega}_{\text{régularisation}}$$

- Pour simplifier, considérons le cas $\lambda = 0$: on suppose donc qu'il n'y a pas de bruit, et on cherche en fait à "inverser" \mathcal{R}

$$\inf_u J(u) = \inf_u \int_{\Omega} (\mathcal{R}u(x) - f(x))^2 d\Omega$$

- De la même manière que précédemment, on cherche u tel que

$$\lim_{\varepsilon \rightarrow 0} \frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = 0, \text{ quelque soit } \phi$$

$$J(u + \varepsilon \phi) - J(u) = \int_{\Omega} \left[2(\mathcal{R}u(x) - f(x)) \varepsilon \mathcal{R}\phi(x) + (\varepsilon \mathcal{R}\phi(x))^2 \right] d\Omega$$

$$\frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = 2 \int_{\Omega} (\mathcal{R}u(x) - f(x)) \mathcal{R}\phi(x) d\Omega + o(1)$$

59 / 82

Parenthèse : produit scalaire de fonction, adjoint d'un opérateur

- On peut définir le produit scalaire (\mathcal{L}^2) entre deux fonctions ainsi :

$$\langle f | g \rangle_{\mathcal{L}^2} = \int_{\Omega} f(x) g(x) d\Omega$$

- Norme associée $\|f\|_{\mathcal{L}^2}^2 = \langle f | f \rangle_{\mathcal{L}^2} = \int_{\Omega} f(x)^2 d\Omega$

- L'**adjoint** d'un opérateur \mathcal{R} , c'est l'opérateur \mathcal{R}^* tel que :

$$\langle \mathcal{R}f | g \rangle = \langle f | \mathcal{R}^*g \rangle, \text{ quelque soit } f \text{ et } g$$

- Exemple : adjoint de l'opérateur div sur l'espace des fonctions nulles aux bords \rightarrow $\overleftarrow{\text{grad}}$

$$\begin{aligned} \langle \overleftarrow{\text{div}} \text{div}(f) | g \rangle &= \int_{\Omega} \text{div}(f) g d\Omega = \int_{\Omega} \text{div} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} g d\Omega = \int_{\Omega} \left(\frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} \right) g d\Omega \\ &= \int_{\Omega} \left(-f_1 \frac{\partial g}{\partial x_1} - f_2 \frac{\partial g}{\partial x_2} \right) d\Omega = - \int_{\Omega} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial g}{\partial x_1} \\ \frac{\partial g}{\partial x_2} \end{pmatrix} d\Omega = \langle f | \overleftarrow{\text{grad}} \overrightarrow{\text{grad}}(g) \rangle \end{aligned}$$

60 / 82

Si $\mathcal{R} \neq \text{Id}$

- On avait obtenu

$$\frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = 2 \int_{\Omega} (\mathcal{R}u(x) - f(x)) \mathcal{R}\phi(x) d\Omega + o(1)$$

- On peut récrire : $\frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = \langle 2(\mathcal{R}u - f) \mid \mathcal{R}\phi \rangle + o(1)$

- Et donc : $\frac{J(u + \varepsilon \phi) - J(u)}{\varepsilon} = \langle 2\mathcal{R}^*(\mathcal{R}u - f) \mid \phi \rangle + o(1)$

- En particulier si on prend $\phi = -2\mathcal{R}^*(\mathcal{R}u - f)$ on obtient

$$J(u + \varepsilon \phi) - J(u) = -\varepsilon \|2\mathcal{R}^*(\mathcal{R}u - f)\|^2 + o(\varepsilon)$$

$$J(u + \varepsilon \phi) - J(u) \leq 0 \quad \text{si } \varepsilon \text{ est suffisamment petit}$$

- la "descente de gradient" sera donc : $\frac{\partial u}{\partial t} = 2\mathcal{R}^*f - 2\mathcal{R}^*\mathcal{R}u$

61/82

Déconvolution, "Défloutage"

- \mathcal{R} est maintenant $\neq \text{Id}$ et représente un opérateur de flou : exemple, convolution avec une gaussienne $\mathcal{R}u = G_{\sigma} * u$.
- Adjoint de l'opérateur "convolution avec une gaussienne" ?

$$\mathcal{R}u(x, y) = (G_{\sigma} * u)(x, y) = \int_{\Omega} G_{\sigma}(x - a, y - b) u(a, b) da db$$

$$\langle \mathcal{R}u \mid v \rangle = \int_{\Omega} (G_{\sigma} * u)(x, y) v(x, y) dx dy$$

$$= \int_{\Omega} \left(\int_{\Omega} G_{\sigma}(x - a, y - b) u(a, b) da db \right) v(x, y) dx dy$$

$$= \int_{\Omega} \left(\int_{\Omega} G_{\sigma}(x - a, y - b) v(x, y) dx dy \right) u(a, b) da db$$

$$= \int_{\Omega} (G_{\sigma} * v)(a, b) u(a, b) da db = \langle u \mid \mathcal{R}v \rangle$$

$$\text{Donc } \mathcal{R}^* = \mathcal{R}$$

63/82

Déconvolution, "Défloutage"



DEMO MATLAB

Remarque : ici on a utilisé un \mathcal{R} connu a priori. En pratique ce n'est pas le cas.

64/82

Inpainting

$$\inf_u J(u) = \inf_u \underbrace{\int_{\Omega} (\mathcal{R}u(x) - f(x))^2 d\Omega}_{\text{terme de fidélité, attache aux données}} + \lambda \underbrace{\int_{\Omega} \Psi(\|\nabla u(x)\|) d\Omega}_{\text{régularisation}}$$

\mathcal{R} est maintenant un opérateur de masquage qui supprime certaines parties de l'image.

- on suppose que l'on a perdu l'information à l'intérieur d'un domaine $D \subset \Omega$
- on définit alors $\mathcal{R}u(x)$ pour $x \in \Omega$ par :

$$(\mathcal{R}u)(x) = \begin{cases} 0 & \text{si } x \in D, \\ u(x) & \text{si } x \notin D \end{cases}$$

$$\bullet \mathcal{R} \left(\text{Image with mask} \right) = \text{Image with mask}$$

66/82

Inpainting

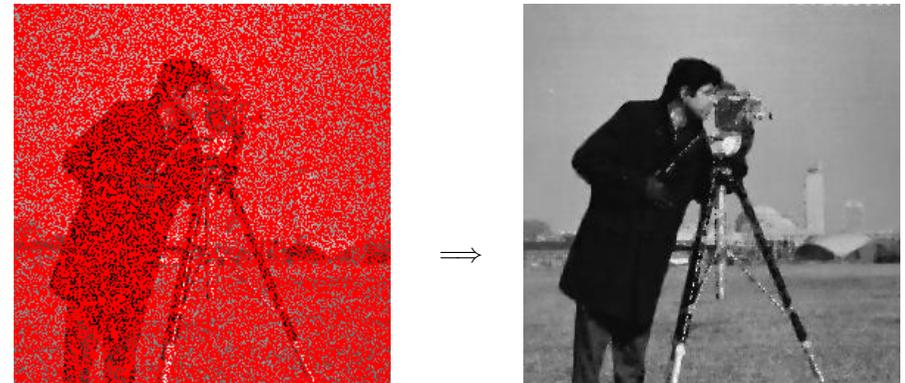
- On cherche donc un u qui, lorsqu'on le masque, $\mathcal{R}u$, soit proche de la donnée f .
- Cela ne suffit pas $\rightarrow u = f$ donne 0
- on doit rajouter une hypothèse de régularité ($\lambda \neq 0$)
- de la même manière que pour la déconvolution, on obtient la "descente de gradient" suivante :

$$\frac{\partial u}{\partial t} = 2\mathcal{R}^*f - 2\mathcal{R}^*\mathcal{R}u + \lambda \operatorname{div}\left(\frac{\Psi'(\|\nabla u\|)}{\|\nabla u\|}\nabla u\right)$$

- Adjoint de l'opérateur de masquage ? \rightarrow on pourra vérifier qu'on a encore $\mathcal{R}^* = \mathcal{R}$

67 / 82

Inpainting

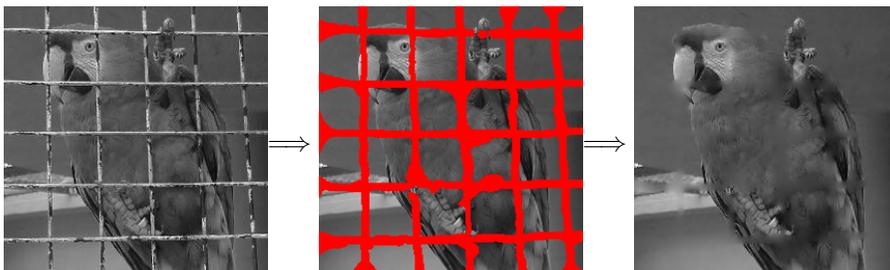


DEMO MATLAB

Remarque : ici on a utilisé un domaine D (en rouge) connu a priori.

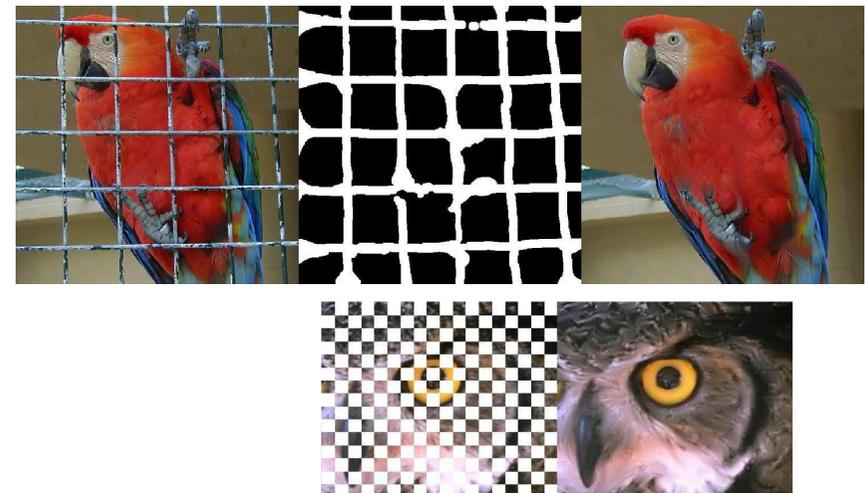
68 / 82

Inpainting



69 / 82

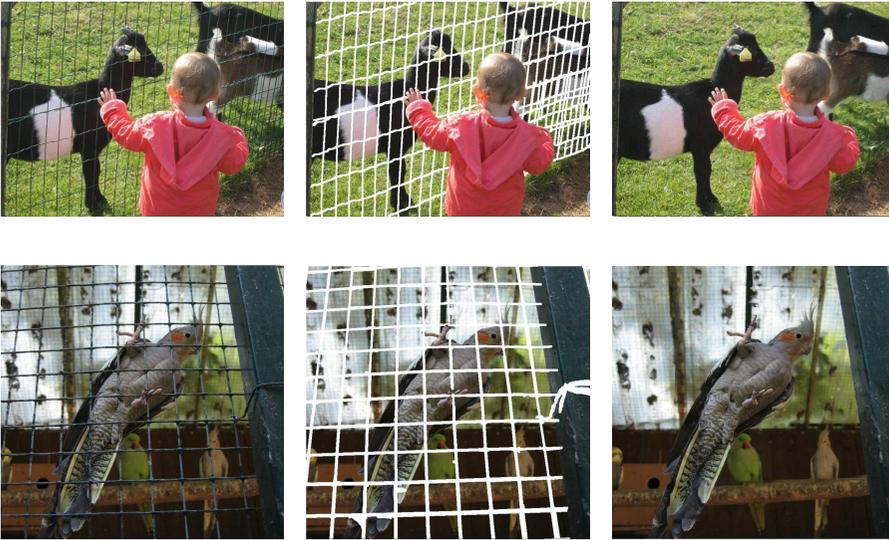
Inpainting



2006 : "Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's" D. Tschumperlé

70 / 82

Inpainting



2006 : "Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's" D. Tschumperlé

73 / 82

Autre type de méthodes : copier-coller

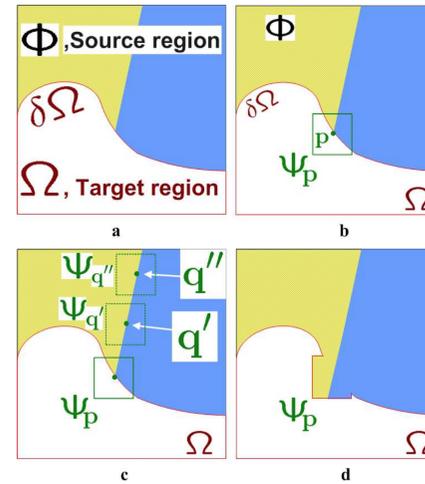
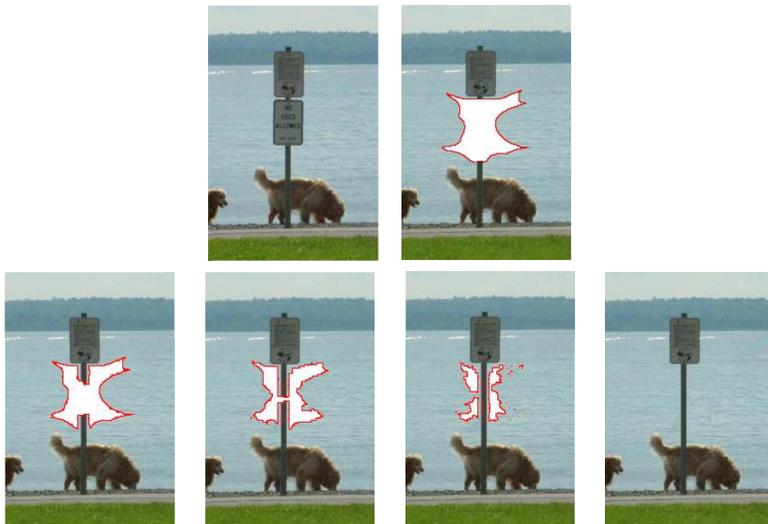


Fig. 2. Structure propagation by exemplar-based texture synthesis. (a) Original image, with the target region Ω , its contour $\delta\Omega$, and the source region Φ clearly marked. (b) We want to synthesize the area delimited by the patch Ψ_p centred on the point $p \in \delta\Omega$. (c) The most likely candidate matches for Ψ_p lie along the boundary between the two textures in the source region, e.g., $\Psi_{q'}$ and $\Psi_{q''}$. (d) The best matching patch in the candidates set has been copied into the position occupied by Ψ_p , thus achieving partial filling of Ω . Notice that both texture and structure (the separating line) have been propagated inside the target region. The target region Ω has, now, shrunk and its front $\delta\Omega$ has assumed a different shape.

"Region Filling and Object Removal by Exemplar-Based Image Inpainting" A. Criminisi, P. Pérez and K. Toyama, 2004

74 / 82

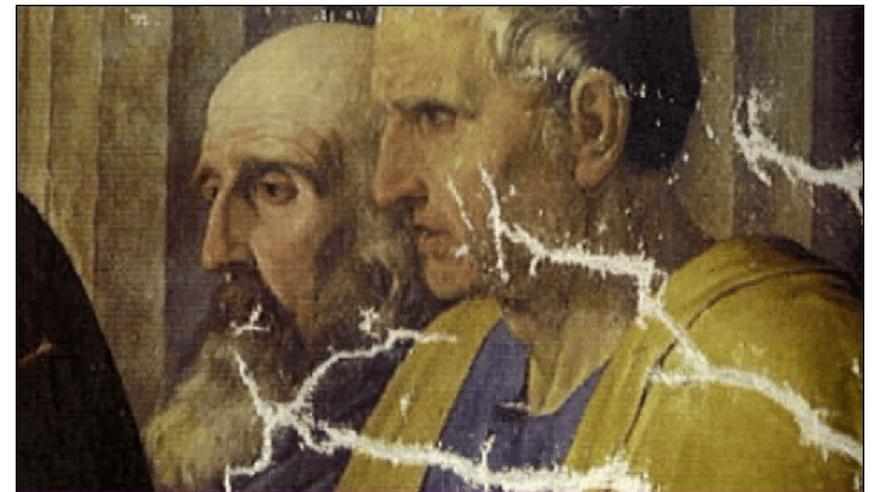
Autre type de méthodes : copier-coller



"Region Filling and Object Removal by Exemplar-Based Image Inpainting" A. Criminisi, P. Pérez and K. Toyama, 2004

75 / 82

Autre type de méthodes : copier-coller



"Exemplar-based Image Inpainting" Criminisi, Pérez and Toyama

76 / 82

Autre type de méthodes : copier-coller



“Exemplar-based Image Inpainting” Criminisi, Pérez and Toyama

77 / 82

Autre type de méthodes : copier-coller

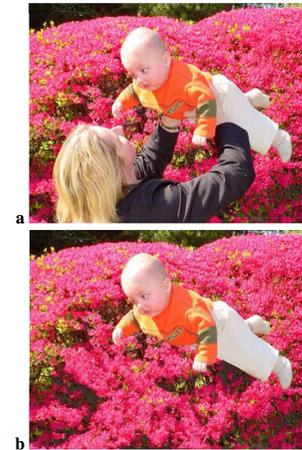


Fig. 22. Removing an object on a highly textured background. (a) Original photograph. (b) One of the two people has been removed. This demonstrates that our algorithm works correctly also for the (simpler) case of “pure” texture.

“Region Filling and Object Removal by Exemplar-Based Image Inpainting” A. Criminisi, P. Pérez and K. Toyama, 2004

78 / 82

Courbe brachistochrone

- Étant donné deux points A et B , quelle la forme doit-on donner à un toboggan pour qu’une bille posée sur celui-ci en A arrive le plus rapidement possible au point B ?

- **Modélisation** : on cherche $f : \mathbb{R} \mapsto \mathbb{R}$ qui minimise $\int_{x_A}^{x_B} \sqrt{\frac{1 + f'(x)^2}{-2gf(x)}} dx$

- **Équation d’Euler-Lagrange** : $f(x)(1 + f'(x)^2) = \text{Cste}$

- **solution explicite** : cycloïde inversée

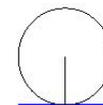
$$x(\theta) = C(\theta - \sin(\theta))$$

$$y(\theta) = -C(1 - \cos(\theta))$$

80 / 82

Courbe cycloïde

Trajectoire d’un point fixé à un cercle qui roule sans glisser sur une droite



Texas A&M
Society of Physics Students
2006-2007

81 / 82

Courbe brachistochrone

