SSII: Bruit et Restauration d'images

SI 3

2016/2017

Diane Lingrand

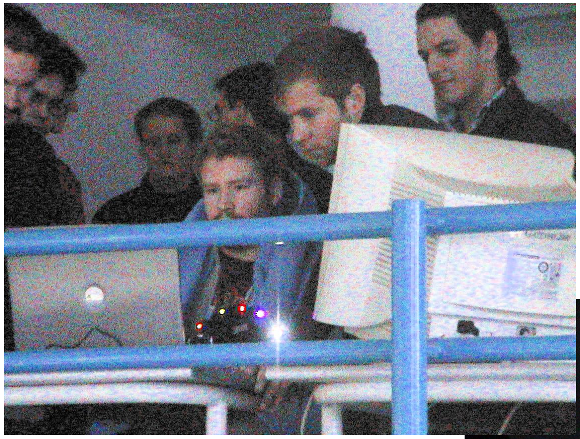
lingrand@polytech.unice.fr

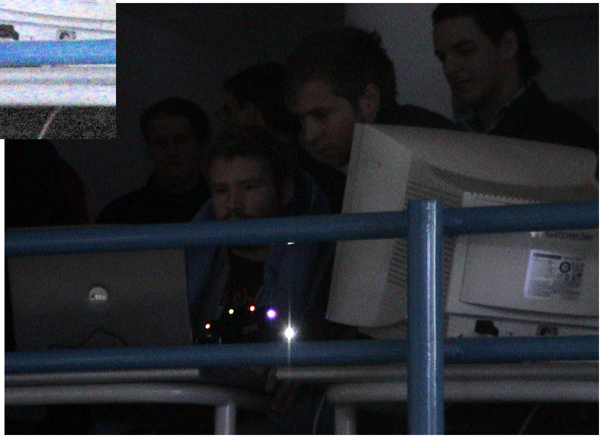
http://www.polytech.unice.fr/~lingrand

Le bruit : causes et modélisation

Un peu de silence svp !


METEO
NICE





Sources de défauts

Contexte d'acquisition
- sur ou sous illumination
- perturbations des capteurs
- nuages en imagerie satellitaire

Capteur
- distorsions (géométriques, d'intensité)
- poussières

Echantillonnage
- phénomène de Moiré (si conditions de Nyquist-Shannon non vérifiées)
- objet dont la taille est = au pixel : bruit de poivre et sel

Quantification
- 256 niveaux de gris ou par composante couleur

Transmission / Stockage
- perte de données, corruption des données

Modélisation du bruit

- Bruit indépendant / bruit dépendant des données de l'image
- Bruit additif / bruit multiplicatif

  pixel + bruit ; pixel * bruit
- bruit: hautes fréquences (généralement + hautes que dans l'image) => filtrage
- types de bruit :
  - bruit impulsionnel: gaussien, exponentiel, …
  - bruit de poivre et sel

Bruit de sel et poivre (*salt and pepper noise)*

ordre n : ajouter n pixels blancs et n pixels noirs aléatoirement dans une image.

souvent exprimé en pourcentages

réalité physique: poussière sur pellicule ou scanner, petits objets, pertes de données

Exemple de bruit poivre et sel

image originale

image bruitée poivre et sel avec p(blanc) =p(noir)=5%

Bruit impusionnel

densité de probabilité de la forme:

$$f(a) = C.e^{-K|a|^{\alpha}}$$

Pour $\alpha = 1$: bruit exponentiel

Pour $\alpha = 2$: bruit gaussien

Bruit gaussien

Bruit additif gaussien de moyenne nulle et variance σ^2

Représente le bruit d'acquisition

Comment implémenter (en Java) ?

```
import java.util.Random;
```

méthode `public double Random.nextGaussian();`

fournit une variable de probabilité gausienne de moyenne nulle et d'écart type 1

⇨ on multiplie par σ pour obtenir une variable de probabilité gaussienne de moy. null et d'écart type σ

Bruit gaussien en C/C++

Initialisation

```
struct timeval tv;
struct timezone tz;
gettimeofday(&tv, &tz);
srand(tv.tv_usec);
```

Variable aléatoire distribution normale entre 0 et 1

```
rand()/(RAND_MAX+1.0f);
```

Variable gaussienne

```
b = rand()/(RAND_MAX+1.0f);
if(b < epsilon) b = epsilon;
c = rand()/(RAND_MAX+1.0f);
a = -2.0 * Math::log(b);
if(a < 0.0) a = 0.0;
else a *= Math::cos(2.0 * Math::pi * c);
```

Exemple de bruit gaussien

image originale

image bruitée par un bruit gaussien d'écart type 4

Encore plus de bruit :

écart type = 8

écart type = 16

Filtrage - Restauration

Filtrage spatial

- Filtre qui s'applique au voisinage d'un pixel dans une image
- Filtrage linéaire / non linéaire
  - linéaire: convolution
- Opérations:
  - lissages (*blurring*)
  - rehaussement de contours (*sharpening*)

Filtres de lissage

- filtres moyenneurs
- filtres gaussiens
- filtres médians
- lissage conservatif
- filtres avec préservation des contours

Filtre moyenneur (*mean filtering)*

- nommé également : *averaging*, *Box filtering*
- Principe : un pixel est remplacé par la moyenne de ses voisins et de lui-même
- Convolution avec un masque de taille variable (2n+1 x 2n+1).

ex:

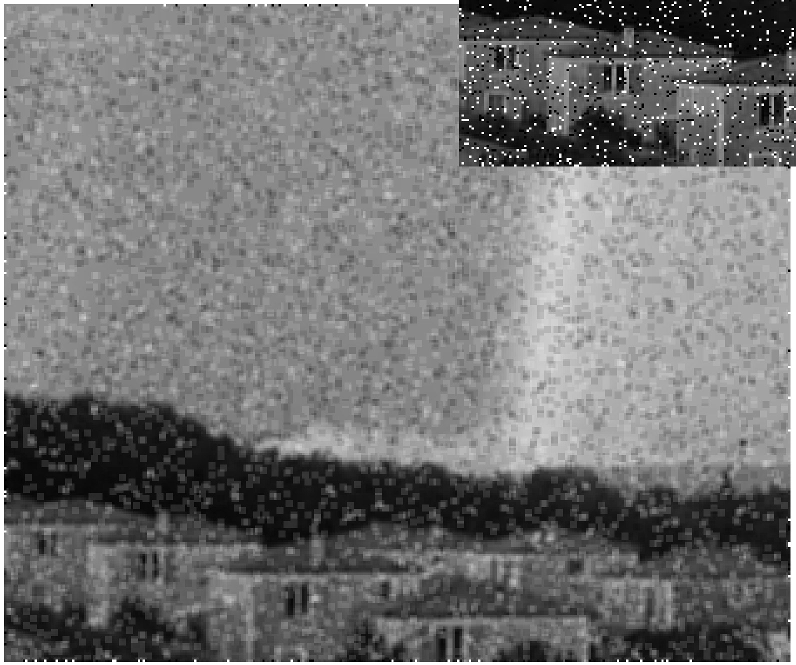
1/9	**1/9**	**1/9**
1/9	**1/9**	**1/9**
1/9	**1/9**	**1/9**

Exemples

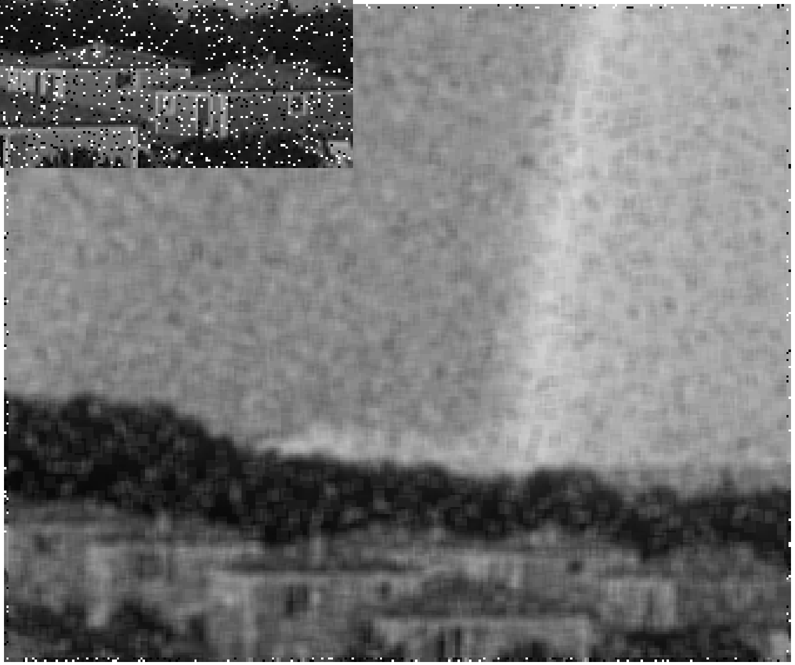
C’est un filtre passe-bas

Dégradation des contours (introduction de flou)

Lissage après bruit poivre et sel :



3x3



5x5

Lissage après bruit gaussien

3x3

5x5

Filtre gaussien (gaussian smoothing)

Gaussienne 2D :

moyenne nulle

variance σ^2

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Principe: convolution avec une gaussienne

En pratique : discrétisation de la gaussienne sur un noyau de taille (2p+1,2p+1)

Exemples de noyaux gaussiens

3x3

[1 2 1]
[2 4 2]
[1 2 1]

5x5

[1 1 2 1 1]
[1 2 4 2 1]
[2 4 8 4 2]
[1 2 4 2 1]
[1 1 2 1 1]

7x7

[1 1 2 2 2 1 1]
[1 2 2 4 2 2 1]
[2 2 4 8 4 2 2]
[2 4 8 16 8 4 2]
[2 2 4 8 4 2 2]
[1 2 2 4 2 2 1]
[1 1 2 2 2 1 1]

Comment calculer un noyau gaussien ? (1)

Masque ? ((2p+1) x (2p+1))

- coefficients entiers (calculs + rapides) puis normalisation en divisant par la somme des coefficients
- encore + efficace : coefficients sous formes de puissances de 2
- σ dépend de la taille du masque (p)

Comment calculer un noyau gaussien ? (2)

- Le dernier coefficient vaut 1
- Sa vraie valeur est obtenue par division avec la somme des coefficients

...	...	
...	2^q	... $2^q.e^{-p^2/(2\sigma^2)}$
	...	...
	$2^q.e^{-p^2/(2\sigma^2)}$	$1 = 2^q.e^{-p^2/\sigma^2}$

Autre exemple de noyau gaussien

$$\frac{1}{273}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline \end{array}$$

Filtrage gaussien après bruit poivre et sel

3x3

7x7

Filtrage gaussien après bruit gaussien ($\sigma = 16$)

3x3

7x7

Filtrage conservatif

Principe : on ne conserve la valeur d'un pixel que si celle-ci est dans l'intervalle de valeurs défini par les voisins. Sinon, on prend la frontière la plus proche.

- si supérieur à la borne sup.: borne sup.
- si inférieur à la borne inf.: borne inf.

Filtrage conservatif après poivre et sel

original bruité

après filtrage conservatif

Filtrage conservatif après bruit gaussien

original bruité

après filtrage conservatif

Filtrage conservatif

63	62	67
88	0	93
140	110	255

$0 \notin [62 ; 255]$

63	62	67
88	62	93
255	110	255

63	0	67
88	0	93
140	110	255

$0 \in [0 ; 255]$

63	0	67
88	0	93
255	110	255

Filtres médians (*median filter*)

- Principe : un pixel est remplacé par la valeur médiane de ses voisins
- élément médian = élément du milieu
  - nécessité de trier les éléments
- permet d'éliminer les éléments aberrants (*outliers)*
- bonne performance pour du bruit sel et poivre

Filtrage médian

63	0	67
88	0	93
255	110	255

63	0	67
88	88	93
255	110	255

63	0	67	88	0	93	255	110	255

tri

0	0	63	67	88	93	110	255	255

valeur médiane

Quelques masques pour les filtres médians

carrés (7x7)

[1 1 1 1 1 1 1]
[1 1 1 1 1 1 1]
[1 1 1 1 1 1 1]
[1 1 1 1 1 1 1]
[1 1 1 1 1 1 1]
[1 1 1 1 1 1 1]
[1 1 1 1 1 1 1]

diamand (7x7)

[0 0 0 1 0 0 0]
[0 0 1 1 1 0 0]
[0 1 1 1 1 1 0]
[1 1 1 1 1 1 1]
[0 1 1 1 1 1 0]
[0 0 1 1 1 0 0]
[0 0 0 1 0 0 0]

carrés (3x3)

[1 1 1]
[1 1 1]
[1 1 1]

en croix (7x7)

[0 0 0 1 0 0 0]
[0 0 0 1 0 0 0]
[0 0 0 1 0 0 0]
[1 1 1 1 1 1 1]
[0 0 0 1 0 0 0]
[0 0 0 1 0 0 0]
[0 0 0 1 0 0 0]

en croix (3x3)

[0 1 0]
[1 1 1]
[0 1 0]

Filtre médian après poivre et sel

masque 3x3, 1 fois

masque 3x3, 2 fois

Filtre médian après bruit gaussien

original bruité

après filtre médian

Rehaussement de contours
Sharpening

filtrage passe-haut (*edge crispening)*

pour contrecarrer les effets du lissage

noyau de convolution

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 10 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ ou } \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix} \text{ ou } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

augmente le bruit

on augmente l'effet en diminuant le coefficient central (essayer 5 au lieu de 10)

Exemple 1:

lissage gaussien (3x3) puis rehausseur (3a)

Exemple 2:

lissage gaussien (7x7) puis rehausseur (3c)

Bilan provisoire

Bruit de poivre et sel:

filtre médian efficace

Bruit gaussien :

filtre gaussien assez efficaces

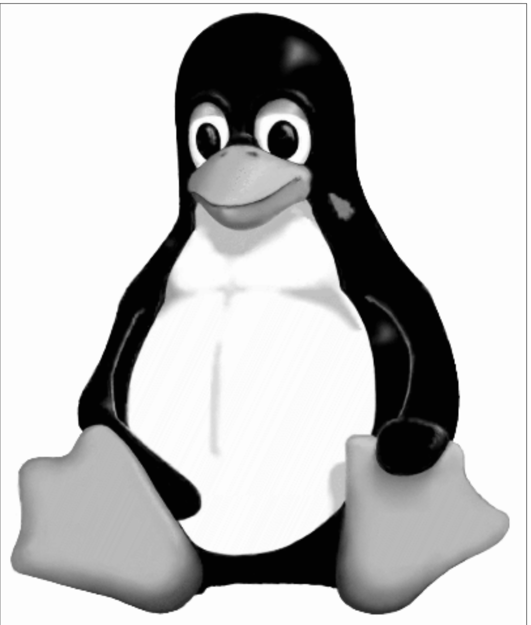
Inconvénient pour les 2 cas :

lissage des contours

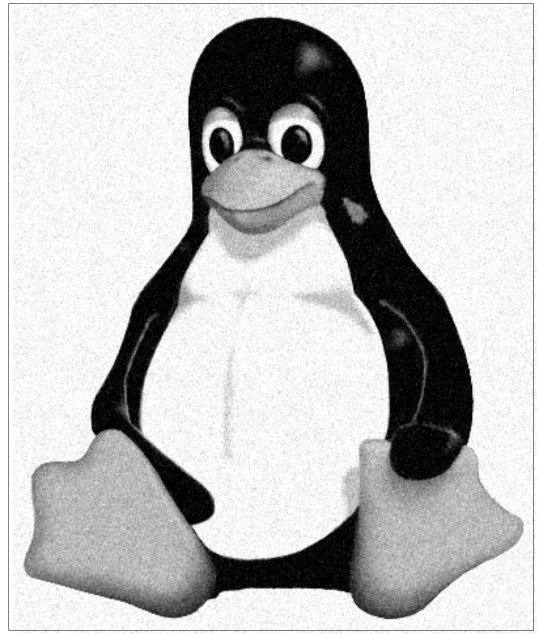
Idée :

lissage uniquement des zones de faibles gradients

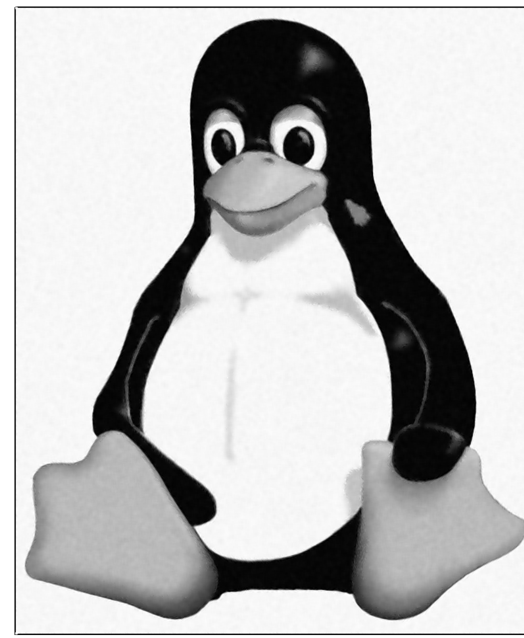
Lissage avec préservation des contours



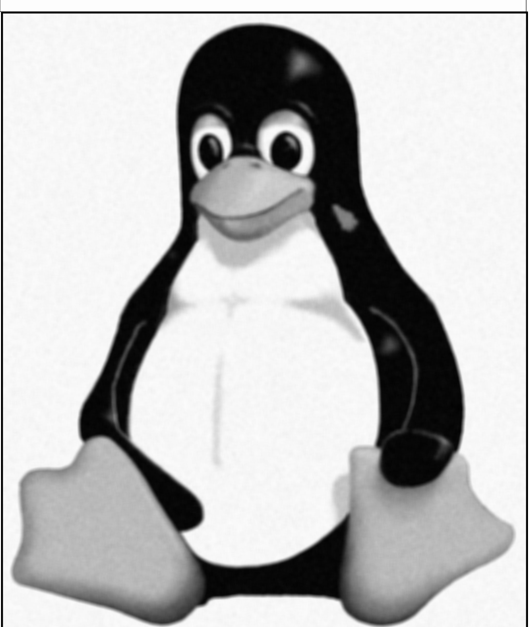
original



bruit gaussien σ =16



avec préservation des contours



moyenneur 7x7

Préservation des contours: détails

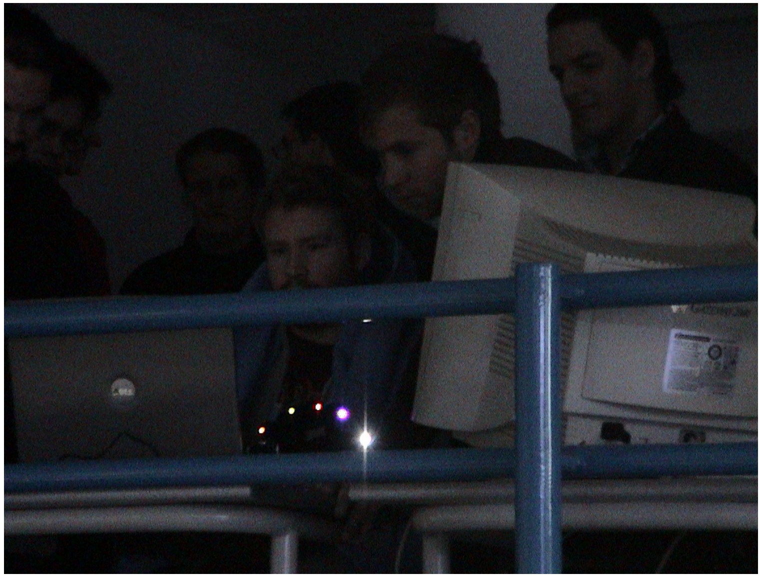
Calcul d'un masque:

- lissage 7x7
- Sobel
- seuillage

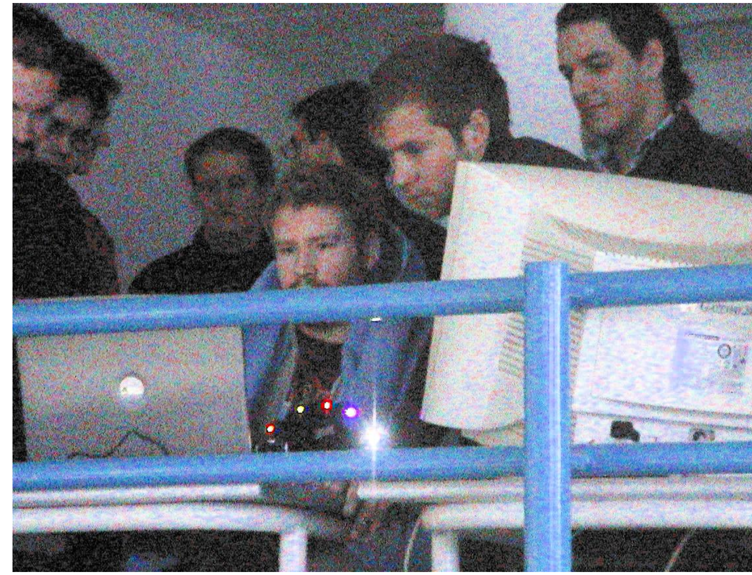
Selon le masque

- lissage 7x7 ou dimensions 5x5 ou 3x3 si en frontière de masque
- calcul de la direction du contour atan(Iy/Ix) et lissage dans cette direction

Un exemple pratique



Égalisation d'histo-gramme



Lissage avec préservation des contours

Approche variationnelle

Retour au filtrage gaussien

Rappel: convolution avec une gaussienne :

$$u_\sigma(\mathbf{x}) = [G_\sigma * u](\mathbf{x}) = \int\int G_\sigma(\mathbf{x}-\tau)u(\tau)d\tau$$

avec

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

Evolution d'une image

On va ajouter un paramètre temporel t d'évolution de l'image :

$$u(\mathbf{x}) \Rightarrow u(\mathbf{x},t)$$

Evolution de l'image :

initialisation : $u(\mathbf{x},t_0) = u_0(\mathbf{x})$

état à un instant t : $u(\mathbf{x},t)$ = image modifiée

évolution : vitesse ?

$$\frac{\partial u}{\partial t}$$

Exemple : équation de la chaleur (1) *The heat equation*

décrit la propagation de la chaleur :

$$\frac{\partial u}{\partial t} = \Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}$$

équation linéaire, parabolique

propriété intéressante : il existe une solution explicite qui est de forme gaussienne :

$$u(\mathbf{x}, t) = \int\int G_{\sqrt{2t}}(\mathbf{x} - \tau) u_0(\tau) d\tau = (G_{\sqrt{2t}} * u_0)(\mathbf{x})$$

équation de la chaleur (2)

Exercice :

vérifier en remplaçant que ça marche

Conclusion :

correspondance entre lissage gaussien de variance $\sigma^2=2t$ et évolution selon l'EDP de l'équation de la chaleur

un filtre gaussien classique peut s'écrire comme solution d'une EDP (*in english : PDE*)

Les **E**quations aux **D**érivées **P**artielles
Partial Differential Equation

- Méthodes anciennes, beaucoup de choses de faites, ...
- Comment appliquer cette technique en imagerie ?
- En général, on n'a pas de solution explicite
  - implémentation de l'évolution

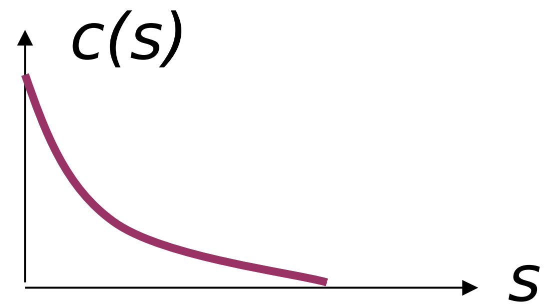
Application des EDP à la restauration

Inconvénient des filtres vus précédemment: pas ou peu de préservation des contours

Equation de la chaleur :

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

$c(|\nabla u|^2)$



on veut :

- lisser les parties de l'image de faible gradient
- en présence de gradient : lisser dans une direction parallèle au contour

idée : pondérer par une fonction qui dépend du gradient [Perona-Malik]

Restauration avec préservation des contours

$$\frac{\partial u}{\partial t} = \operatorname{div}(c(|\nabla u|^2)\nabla u)$$

idée intuitive :

- $|\nabla u|$ faible : c proche de 1 : équation de la chaleur
- $|\nabla u|$ fort : c proche de 0 : vitesse nulle, lissage arrêté

Préservation des contours (suite)

plus précisément : décomposons en 2 termes : tangentiel et normal

$$\begin{aligned}\operatorname{div}(c(|\nabla u|^2)\nabla u) = &\ 2(u_x^2 u_{xx} + u_y^2 u_{yy} + 2u_x u_y u_{xy})c'(|\nabla u|^2) \\ + &\ (u_{xx} + u_{yy})c(|\nabla u|^2)\end{aligned}$$

En prenant : $b(s) = c(s) + 2sc'(s)$

on obtient :

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) = c(|\nabla u|^2)u_{TT} + b(|\nabla u|^2)u_{NN}$$

Fonction de pondération

Contraintes : $$\lim_{s \to +\infty} \frac{sc'(s)}{c(s)} = -\frac{1}{2}$$

$$\begin{cases} c : [0, \infty[\to]0, +\infty[\text{ décroissante} \\ c(0) = 1 \\ c(s) \text{ de comportement asymptotique à l'}\infty : \frac{1}{\sqrt{s}} \\ b(s) = c(s) + 2sc'(s) > 0 \end{cases}$$

Exemple :

$c(s) = 1 / \sqrt{1 + s}$