Méthodes de décomposition de domaine multi-niveaux.

Présentation des travaux du stage

Omar Mokhtari

Encadrants Mia La Rochelle

Catherine Choquet
Michel Berthier

Traitement d'images numériques

- **Débruitage** / , **Réhaussement**,
- **Restauration** / **Inpainting**,
- Détection de contours / Segmentation,
- Reconnaissance d'objets / Extraction,
- Analyse/ Classification.

Débruitage

- Différentes origines du bruit :
    - Aquisition / Quantification,
    - Compression / Transmission.
- Différentes natures :
    - Bruit Gaussien / Anisotropique,
    - Bruit impultionnel / "En dent de scie".

On suppose le modèle de bruit suivant :

$$u_0 = B(u) + n.$$

-u, u_0 intensité en niveaux de gris (image noir et blanc)
-B lissage,
-n bruit additionnel.

Objectif :
Retrouver l'image débruitée u :

-supprimer le bruit,

-réhausser l'image.

Première approche

Équation de la chaleur

Idée :
Supprimer le bruit en lissant l'image $\longrightarrow$ Processus de diffusion.

Équation de diffusion (linéaire) :

$$\left\{ \begin{array}{rcl} \partial_t u & = & \Delta u, \\ u_{t=0} & = & u_0. \end{array} \right.$$

Problème : On perd les contours de l'image (impression de flou).

Diffusion anisotropique

Équation de Perona & Malik

Équation de diffusion (non linéaire) :

$$\begin{cases} \partial_t u & = & \operatorname{div}\left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right), \\ u_{t=0} & = & u_0. \end{cases}$$

avec un coefficient de diffusion :

$$\mathbf{c}(s) = \frac{1}{1 + \left(\frac{s}{k}\right)^2}.$$

- Zones homogènes : diffusion,
- Contours : arrêt de la diffusion,
- Choix du paramètre k...

Approche Variationnelle

Densité d'énergie

On définit une densité d'énergie $\Psi(|\nabla u|)$ sur l'image.
Le problème revient à minimiser l'énergie :

$$\mathbf{E}(u) = \int_\Omega \Psi(|\nabla u|)\; d\Omega.$$

Euler-Lagrange :

$$-\mathrm{div}(\mathbf{c}(|\nabla u|)\nabla u) = 0$$

où $\mathbf{c}(|\nabla u|) := \frac{\Psi'(|\nabla u|)}{|\nabla u|}$

Descente de Gradient :

$$\partial_t u = \mathbf{c}(|\nabla u|)\nabla u.$$

Potentiel à trois puits

- Ψ à 3 points d'attraction :
  - Gradient nulle : aucune energie,
  - Pénalisation des gradients faibles,
  - Faible énergie pour les gradients élevés,
  - Pénalisation des gradients élevés (pb "bien posé").
- Coefficient de diffusion **c** :
  - Positif pour les gradients faibles
  - Négatif pour les gradients moyens (réhaussement)
  - Nul pour les gradients élevés.

$$\mathbf{c}(\nabla u) = \frac{1 - \tanh(\mathbf{b}_1 \cdot (\|\nabla u\|) - \mathbf{k}_f)) - \mathbf{a}(1 - \tanh^2(\mathbf{b}_2 \cdot (\|\nabla u\|) - \mathbf{k}_b)))}{2}$$

Potentiel à trois puits

image gradient/c

Hyper-diffusion :

Processus de diffusion inverse ($\mathbf{c}<0$) $\longmapsto$ équation très instable.

On ajoute un terme d'ordre 4 :

$$-\nabla^4 u := \Delta^2(u) = \partial_{xxxx} u + 2\partial_{xxyy} u + \partial_{yyyy} u.$$

Double intérêt :

- Réduire les oscillations entre les 3 états puits d'énergie $\backsim$ terme de viscosité,
- Traiter le bruit à gradient élevé (bruit impultionnel...).

On considère finalement l'équation :

$$\begin{cases} \partial_t u &= \operatorname{div}\left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right) + \lambda(u_0 - u) - \epsilon\nabla^4 u \\ u_{t=0} &= u_0. \end{cases}$$

où $\lambda(u_0 - u)$: terme d'attache à la donnée initiale.

Résolution Directe

On veut résoudre :

$$\begin{cases} \partial_t u & = & \operatorname{div}\left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right) + \lambda(u_0 - u) - \epsilon\nabla^4 u \\ u_{t=0} & = & u_0 . \end{cases}$$

Avec les conditions aux bords :

$$\begin{cases} \partial_n u & = & 0, \\ \partial_n \Delta u & := & 0. \end{cases}$$

- Pixelisation de l'image $\rightarrow$ Différences finies,
- Terme d'ordre 4 $\rightarrow$ ~~Différences finies~~ $\rightarrow$ Éléments finis.

Résolution directe (élements finis $P_2 \in C^1(\Omega)$) :

$$\int_\Omega \partial_t u = -\int_\Omega \left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right)\nabla\phi + \lambda\int_\Omega (u_0 - u)\phi - \epsilon\int_\Omega \Delta u \Delta\phi$$

- Éléments finis P_2 de *Morley*,
- Éléments finis P_3 avec non-flux ($\partial_n\phi = 0$) sur les arêtes du maillage $\rightarrow$ **Coûteux !**

Résolution Directe

On veut résoudre :

$$\left\{ \begin{array}{rcl} \partial_t u & = & \mathrm{div}\left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right) + \lambda(u_0 - u) - \epsilon \nabla^4 u \\ u_{t=0} & = & u_0. \end{array} \right.$$

Avec les conditions aux bords :

$$\left\{ \begin{array}{rcl} \partial_n u & = & 0, \\ \partial_n \Delta u & := & 0. \end{array} \right.$$

- Pixelisation de l'image $\rightarrow$ Différences finies,
- Terme d'ordre 4 $\rightarrow$ ~~Différences finies~~ $\rightarrow$ Éléments finis.

Résolution directe (élements finis $P_2 \in C^1(\Omega)$) :

$$\int_\Omega \partial_t u = -\int_\Omega \left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right)\nabla\phi + \lambda \int_\Omega (u_0 - u)\phi - \epsilon \int_\Omega \Delta u \Delta\phi$$

- Éléments finis P_2 de *Morley*,
- Éléments finis P_3 avec non-flux ($\partial_n \phi = 0$) sur les arêtes du maillage $\rightarrow$ **Coûteux !**

Résolution Directe

On veut résoudre :

$$\left\{\begin{array}{rcl} \partial_t u & = & \operatorname{div}\left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right) + \lambda(u_0 - u) - \epsilon\nabla^4 u \\ u_{t=0} & = & u_0 . \end{array}\right.$$

Avec les conditions aux bords :

$$\left\{\begin{array}{rcl} \partial_n u & = & 0, \\ \partial_n \Delta u & := & 0. \end{array}\right.$$

- Pixelisation de l'image $\rightarrow$ Différences finies,
- Terme d'ordre 4 $\rightarrow$ ~~Différences finies~~ $\rightarrow$ Éléments finis.

Résolution directe (élements finis $P_2 \in C^1(\Omega)$) :

$$\int_\Omega \partial_t u = -\int_\Omega \left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right)\nabla\phi + \lambda\int_\Omega (u_0 - u)\phi - \epsilon\int_\Omega \Delta u \Delta\phi$$

- Éléments finis P_2 de *Morley*,
- Éléments finis P_3 avec non-flux ($\partial_n \phi = 0$) sur les arêtes du maillage $\rightarrow$ **Coûteux !**

Résolution Directe

On veut résoudre :

$$\begin{cases} \partial_t u &= \operatorname{div}\left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right) + \lambda(u_0 - u) - \epsilon\nabla^4 u \\ u_{t=0} &= u_0. \end{cases}$$

Avec les conditions aux bords :

$$\begin{cases} \partial_n u &= 0, \\ \partial_n \Delta u &:= 0. \end{cases}$$

- Pixelisation de l'image $\rightarrow$ Différences finies,
- Terme d'ordre 4 $\rightarrow$ ~~Différences finies~~ $\rightarrow$ Éléments finis.

Résolution directe (élements finis $P_2 \in C^1(\Omega)$) :

$$\int_\Omega \partial_t u = -\int_\Omega \left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right)\nabla\phi + \lambda\int_\Omega (u_0 - u)\phi - \epsilon\int_\Omega \Delta u \Delta\phi$$

- Éléments finis P_2 de *Morley*,
- Éléments finis P_3 avec non-flux ($\partial_n\phi = 0$) sur les arêtes du maillage $\rightarrow$ **Coûteux !**

On introduit la variable intermédiaire : $v = \Delta u$,

On obtient un problème couplé :

$$\left\{ \begin{array}{rcl} \partial_t u & = & \operatorname{div}\left(\mathbf{c}\left(|\nabla u|\right)\nabla u\right) + \lambda(u_0 - u) - \epsilon\Delta v \\ v & = & \Delta u. \end{array} \right.$$

Avec les conditions aux bords (Neumann) :

$$\left\{ \begin{array}{rcll} \partial_n u & = & 0 & \text{sur } \partial\Omega, \\ \partial_n v & = & 0 & \text{sur } \partial\Omega. \end{array} \right.$$

- Équation d'ordre 2 $\rightarrow$ Éléments finis P_1.

Discétisation temporelle :

- On utilise un schéma d'Euler :
  - Les termes linéaires sont traités de manière **implicite**,
  - Cas du terme non-linéaire : **semi-explicite**.
- L'équation devient (avec $u^0 = u_0$ donnée) :

$$\left\{\begin{array}{rcl} \frac{u^{n+1}-u^n}{\Delta t} & = & \operatorname{div}\left(\mathbf{c}\left(|\nabla u^n|\right)\nabla u^{n+1}\right) - \lambda(u^{n+1} - u_0) - \epsilon \Delta v^{n+1}. \\ v^{n+1} & = & \Delta u^{n+1}. \end{array}\right.$$

Formulation faible :

Trouver $(u, v) \in H^1(\Omega) \times H^1(\Omega)$ tel que : $\forall(\phi, \psi) \in H^1(\Omega) \times H^1(\Omega)$ on a :

$$\left\{\begin{array}{rcl} \int_\Omega \frac{u^{n+1}-u^n}{\Delta t}\phi & = & \int_\Omega \mathbf{c}\left(|\nabla u^n|\right)\nabla u^{n+1}\nabla\phi - \lambda\int_\Omega (u^{n+1} - u_0)\phi - \epsilon\int_\Omega \nabla v^{n+1}\nabla\phi. \\ \int_\Omega v^{n+1}\psi & = & \int_\Omega \nabla u^{n+1}\nabla\psi. \end{array}\right.$$

Résolution

Algorithme

Algorithme

- $u_{old} = u_0$;

Faire :

- $\underbrace{\langle \frac{1}{\Delta t} u + \lambda u, \phi\rangle + \langle \mathbf{c}\left(|\nabla u_{old}|\right)\nabla u - \varepsilon \nabla v, \nabla\phi\rangle}_{\text{Terme bilinéaire}} - \underbrace{\langle \frac{1}{\Delta t} u_{old} + \lambda u_0, \phi\rangle}_{\text{Terme linéaire}} = 0;$
- $u_{old} = u$;

Tant que : $\|u - u_{old}\| > \epsilon_0$.

Implementation

```
solve MonEDP([u,v],[p,q])
= int2d(Th)(u*p/deltat+c(uold)*grad(u)'*grad(p)+lambda*u*p-eps*grad(v)'*grad(p)-
v*q-grad(u)'*grad(q))
- int2d(Th)(uold*p/deltat+lambda*u0*p) ;
```

Freefem++ s'occupe du reste ! ...

Algorithme

- $u_{old} = u_0$;

Faire :

- $$\underbrace{\langle \frac{1}{\Delta t} u + \lambda u, \phi \rangle + \langle \mathbf{c}\left(|\nabla u_{old}|\right) \nabla u - \varepsilon \nabla v, \nabla \phi \rangle}_{\text{Terme bilinéaire}} - \underbrace{\langle \frac{1}{\Delta t} u_{old} + \lambda u_0, \phi \rangle}_{\text{Terme linéaire}} = 0;$$
- $u_{old} = u$;

Tant que : $\|u - u_{old}\| > \epsilon_0$.

Implementation

```
solve MonEDP([u,v],[p,q])
= int2d(Th)(u*p/deltat+c(uold)*grad(u)'*grad(p)+lambda*u*p-eps*grad(v)'*grad(p)-
v*q-grad(u)'*grad(q))
- int2d(Th)(uold*p/deltat+lambda*u0*p) ;
```

Freefem++ s'occupe du reste ! ...

Améliorations

Critère sur le gradient

Le gradient ∇u de l'image varie au cours des itérations :

$$\Longrightarrow [k_f, k_b] = [1, 2] \cdot k(t)$$

Plusieurs choix possibles :

- $k(t) = \frac{90}{100} \cdot \int_\Omega |\nabla u(t)| d\Omega$:

```
Du=normGrad(u);
k = .9*int2d(Th)(Du);
```

- $k(t) = 1.4826 \cdot (\mathbf{median}(\|\nabla u(t)\| - \mathbf{median}(\|\nabla u(t)\|)))$:

```
Du=normGrad(u);
int n=Du.n;
Du[].sort;
k = (n%2==0)?(Du[][n/2-1]+Du[][n/2])/2 : Du[][(n-1)/2];
Du=abs(Du-kf);
Du[].sort;
k = (n%2==0)? (Du[][n/2-1]+Du[][n/2])/2 : Du[][(n-1)/2];
k=1.4826*k;
```

Améliorations

Produit de convolution

Coefficient de diffusion calculé sur une version lissée :

$$\partial_t u = \operatorname{div}\left(\mathbf{c}\left(|\nabla u_\sigma|\right)\nabla u\right) + \cdots$$

où $u_\sigma := G_\sigma * u$ (produit de convolution) avec :

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

Plusieurs possibilités :

- Calcul direct (coûteux)
- En passant par Fourier :

$$\begin{aligned} \Im(G_\sigma * u) &= \Im(G_\sigma)\Im(u), \\ \Rightarrow \qquad G_\sigma * u &= \Im^{-1}(\Im(G_\sigma)\Im(u)). \end{aligned}$$

- On résolvant l'équation de la chaleur à $\mathbf{t} = \sigma^2/2$:

$$\left\{\begin{array}{lcll} \partial_t u & = & \Delta u & \text{dans } \Omega, \\ \partial_n u & = & 0 & \text{sur } \partial\Omega. \end{array}\right. \Longrightarrow$$

```
solve EDP([Gu],[p])=
 int2d(Th)(Gu*p/dt+grad(Gu)'*grad(p))
- int2d(Th)(Guold*p/dt);
```

Débruitage

Bruit Gaussien
Bruit impultionnel

Applications

Réhaussement de contours

Bien choisir les coefficients $\mathbf{k}_f$ et $\mathbf{k}_b$.

Inpainting

image...

Objectif : Remplir une région endommagée D de l'image à partir de l'information présente autour.

Idée : Choisir un coefficient d'attache $\lambda(x) = \lambda_0 \cdot \mathbb{1}_{\Omega \setminus D}(x)$:

$$\begin{cases} \lambda = 0 & \text{dans la zone à inpainter,} \\ \lambda = \lambda_0 & \text{en dehors.} \qquad (\text{avec } \lambda_0 \gg 1) \end{cases}$$

Images...

Inpainting

Choix de λ_0

λ_0 petit $\rightarrow$ Image floutée

λ_0 grand $\rightarrow$ Problèmes d'interface

Nouveau choix de λ :

Bilan

Traitement d'image :

- Intérêt des EDP,
- Plusieurs applications d'une même équation (débruitage, réhaussement, inpainting...),
- Difficulté du choix des paramètres...

Freefem :

- Mise en œuvre rapide et simple (une fois l'équation établie...)
- Faibles temps de calculs
- Idéal pour jouer avec les paramètres...