Parallélisation d'algorithmes d'optimisation convexe

N. Pustelnik et C. Chaux

LIGM - Université Paris-Est Marne-la-Vallée

Journée logiciels, 19 janvier 2010

Quels besoins ?

- Méthode de restauration d'images pouvant traiter
  - différents types de bruit (gaussien, impulsionnel, poissonien,...)
  - différents types de dégradations linéaires (flou, projection,...)

Quels besoins ?

Observations: $z \in \mathbb{R}^M$ avec $z = \mathcal{D}_\alpha(Ty)$



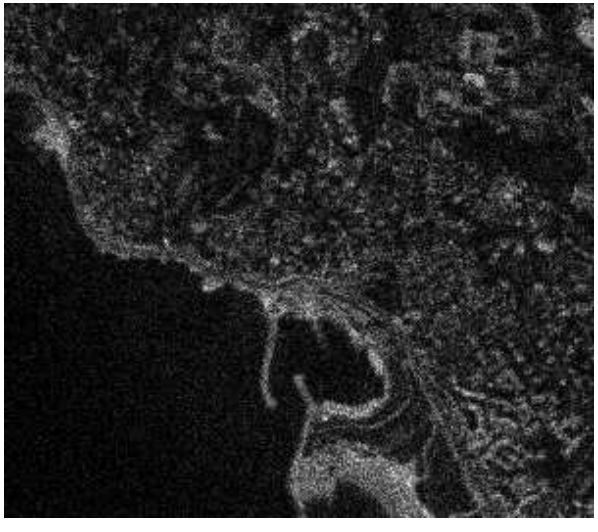
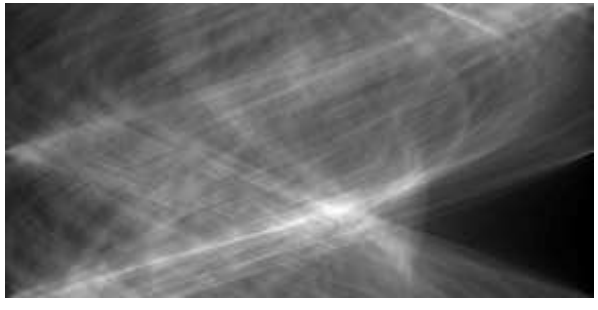



Image originale: $y \in \mathbb{R}^N$

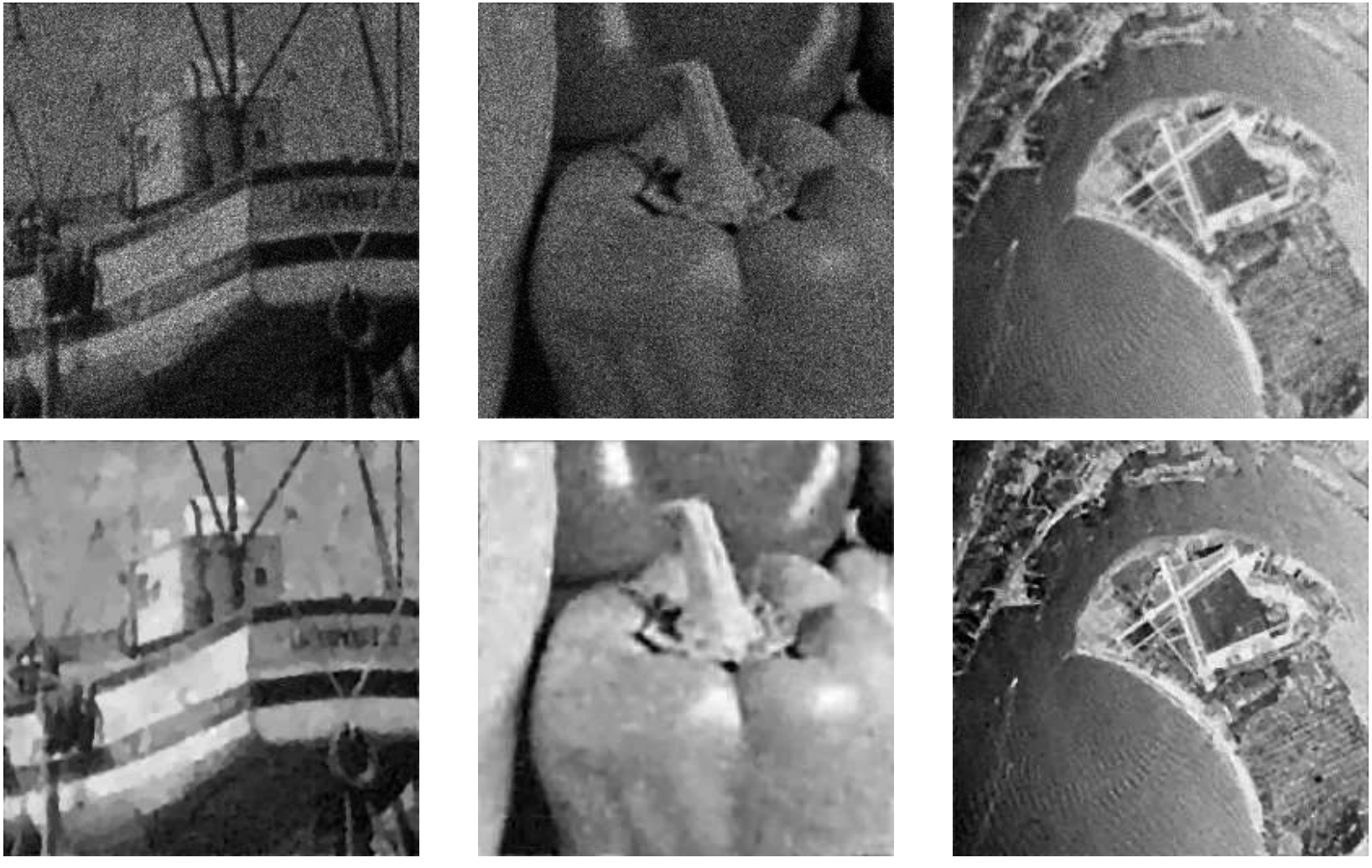



Quels besoins ?

- Méthode de restauration d'images pouvant traiter
  - différents types de bruit (gaussien, impulsionnel, poissonien,...)
  - différents types de dégradations linéaires (flou, projection,...)

- Méthode donnant de bons résultats visuels !

Quels besoins ?



Quels besoins ?

- Méthode de restauration d'images pouvant traiter
  - différents types de bruit (gaussien, impulsionnel, poissonien,...)
  - différents types de dégradations linéaires (flou, projection,...)
- Méthode donnant de bons résultats visuels !
- Méthode rapide : faible temps de calcul même pour des images de grandes tailles

Quels besoins ?

- Méthode de restauration d'images pouvant traiter
  - différents types de bruit (gaussien, impulsionnel, poissonien,...)
  - différents types de dégradations linéaires (flou, projection,...)
- Méthode donnant de bons résultats visuels !
- Méthode rapide : faible temps de calcul même pour des images de grandes tailles $\Rightarrow$ Implémention en parallèle de l'algorithme.

Quelle méthodologie ? ⇒ Optimisation convexe & ondelettes

Notre objectif

$\min_{y \in \mathbb{R}^N} \quad d(Ty, z) + \kappa \, \mathrm{tv}(y) + \vartheta \, f(Fy) + \iota_C(y)$
où $\kappa > 0$, $\vartheta > 0$.

- $z \in \mathbb{R}^M$: Observations,
- $y \in \mathbb{R}^N$: Image composée de N pixels,
- $T \in \mathbb{R}^{M \times N}$: Matrice associée à l'opérateur de dég. lin.,
- $F \in \mathbb{R}^{K \times N}$: Opérateur d'analyse de trame,
- $d(\cdot, z)$: Terme d'attache aux données (quadratique, divergence de Kullback-Leibler,...),
- tv : Variation totale,
- f : Terme de rég. sur les coeff. de trame (par ex. norme ℓ_1),
- ι_C : Fonction indicatrice d'un ens. convexe fermé $C = [0, 255]^N$.

Quelle méthodologie ? ⇒ Optimisation convexe & ondelettes

Notre objectif

$\min_{x \in \mathbb{R}^K} \quad d(TF^\top x, z) + \kappa\, \mathrm{tv}(F^\top x) + \vartheta\, f(x) + \iota_C(F^\top x)$
où $\kappa > 0$, $\vartheta > 0$.

- $z \in \mathbb{R}^M$: Observations,
- $x \in \mathbb{R}^K$: coefficients de trame,
- $T \in \mathbb{R}^{M \times N}$: Matrice associée à l'opérateur de dég. lin.,
- $F^\top \in \mathbb{R}^{N \times K}$: opérateur de synthèse de trame,
- $d(\cdot, z)$: Terme d'attache aux données (quadratique, divergence de Kullback-Leibler,...),
- tv : Variation totale,
- f : Terme de rég. sur les coeff. de trame (par ex. norme ℓ_1),
- ι_C : Fonction indicatrice d'un ens. convexe fermé $C = [0, 255]^N$.

Quelle méthodologie ? ⇒ Optimisation convexe & ondelettes

Notre objectif

$\min_{x \in \mathbb{R}^K} \quad d(TF^\top x, z) + \kappa\, \mathrm{tv}(F^\top x) + \vartheta\, f(x) + \iota_C(F^\top x)$
où $\kappa > 0$, $\vartheta > 0$.

- $z \in \mathbb{R}^M$: Observations,
- $x \in \mathbb{R}^K$: coefficients de trame,
- $T \in \mathbb{R}^{M \times N}$: Matrice associée à l'opérateur de dég. lin.,
- $F^\top \in \mathbb{R}^{N \times K}$: opérateur de synthèse de trame,
- $d(\cdot, z)$: Terme d'attache aux données (quadratique, divergence de Kullback-Leibler,...),
- tv : Variation totale,
- f : Terme de rég. sur les coef. de trame (par ex. norme ℓ_1),
- ι_C : Fonction indicatrice d'un ens. convexe fermé $C = [0, 255]^N$.

⇒ Permet de traiter différents types de dégradations.

Quelle méthodologie ? ⇒ Optimisation convexe & ondelettes

Régularisation ondelettes

⇓

Manque de régularité

Variation totale

⇓

Effet de "staircasing"

Quel algorithme ? ⇒ Algorithme Parallèle Proximal (PPXA) : $\min_{x\in\mathcal{H}} \sum_{j=1}^{J} f_j$

Initialisation

Choisir $\gamma \in]0, +\infty[$.
Choisir $(\omega_j)_{1\leq j\leq J} \in]0,1]^J$ telle que $\sum_{j=1}^{J} \omega_j = 1$.
Choisir $(u_{j,0})_{1\leq j\leq J} \in (\mathbb{R}^K)^J$ et $\xi_0 = \sum_{j=1}^{J} \omega_j u_{j,0}$.

Itérations [Combettes and Pesquet, 2008]

For $\ell = 0, 1, \ldots$
  Pour $j = 1, \ldots, J$
    $\boxed{p_{j,\ell} = \mathrm{prox}_{\gamma f_j/\omega_j} u_{j,\ell}} + a_{j,\ell}$ ← Étape parallélisable
  $p_\ell = \sum_{j=1}^{J} \omega_j p_{j,\ell}$
  Choisir $\lambda_\ell \in]0, 2[$
  Pour $j = 1, \ldots, J$
    $\boxed{u_{j,\ell+1} = u_{j,\ell} + \lambda_\ell \left(2\, p_\ell - \xi_\ell - p_{j,\ell}\right)}$ ← Étape parallélisable
  $\xi_{\ell+1} = \xi_\ell + \lambda_\ell \left(p_\ell - \xi_\ell\right)$

Quels outils ?

- Programmation : langage C.
- Bibliothèques :
  - transformée de Fourier : GSL (GNU Scientific Library) / FFTW (Fatest Fourier Transform in the West),
  - statistiques (génération du bruit, est. des paramètres) : GSL,
  - ondelettes : Libmri (toolbox développée au sein de l'équipe) ⇒ Toolbox développée par Alexandru MARIN (Stage M2),
  - Programmation parallèle : Open MP / Pthreads ⇒ Toolbox développée par Adji Néné DIAO & Baptiste MAZIN (Stage M1).

Quelles performances ?

Image de taille 128×128

Librairie	Flou	FFTW	GSL	FFTW //	GSL //
Open MP	1x1	9	10	6 (1.46)	8 (1.27)
	3x3	22	36	10 (2.18)	12 (3.05)
	7x7	61	121	18 (3.25)	23 (5.22)
	9x9	222	461	45 (4.85)	69 (6.67)
	31x31	861	1809	153 (5.61)	241 (7.48)
Pthread	3x3	22	36		16 (2.26)
	7x7	61	121		39 (3.06)
	9x9	222	461		106 (4.34)
	31x31	861	1809		392 (4.61)

Quelles performances ?

Image de taille 256×256

Librairie	Flou	FFTW	GSL	FFTW //	GSL //
Open MP	1x1	32	40	22 (1.40)	28 (1.40)
	3x3	83	146	34 (2.43)	41 (3.52)
	7x7	246	490	72 (3.41)	89 (5.49)
	9x9	884	1849	231 (3.82)	267 (6.92)
	31x31	3441	7308	860 (4.00)	977 (7.48)
Pthread	3x3	3	146		53 (2.05)
	7x7	246	490		120 (4.06)
	9x9	884	1849		401 (4.60)
	31x31	3441	7308		1536 (4.76)

Quelles performances ?

Image de taille 512×512

Librairie	Flou	FFTW	GSL	FFTW //	GSL //
Open MP	1x1	143	150	94 (1.52)	91 (1.65)
	3x3	457	697	169 (2.70)	169 (4.11)
	7x7	1448	2426	487 (2.97)	461 (5.25)
	9x9	5365	9056	1649 (3.25)	1508 (6.00)
	31x31	20691	35418	6306 (3.28)	5542 (6.39)
Pthread	3x3	457	697		224 (3.11)
	7x7	1448	2426		621 (3.90)
	9x9	5365	9056		2125 (4.26)
	31x31	20691	35418		8022 (4.41)

Conclusions

Conclusion sur la parallélisation

- Parallélisation avec openMP simple à mettre en œuvre et performante.
- Tests avec implémentation sur carte graphique ⇒ à optimiser.
  ⇒ Soumission PEPS Interaction Maths-Informatique-Ingénierie (Équipes Signal et A3SI + MIP Toulouse)

Conclusion sur le logiciel

- Logiciel visant à restaurer/reconstruire des images en utilisant des méthodes efficaces d'optimisation convexe.
- Logiciel visant à être adapté aux architectures multicœurs.
- Reste à implémenter d'autres types de bruit (Gaussien et impulsionnel) et opérateur de projection afin d'avoir un logiciel plus complet.