A PROXIMAL BASED STRATEGY FOR SOLVING DISCRETE MUMFORD-SHAH AND AMBROSIO-TORTORELLI MODELS

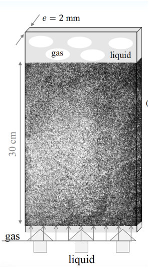
H. TrieuVy LE[1]   Marion Foare[2]   Nelly Pustelnik[1]

[1]Univ Lyon, Ens de Lyon, Univ Lyon 1, CNRS, Laboratoire de Physique, Lyon, 69342, France

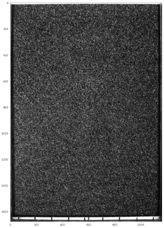
[2]Univ Lyon, Ens de Lyon, Univ Lyon 1, CNRS, INRIA, LIP, Lyon, 69342, France.

Conference on Digital Geometry and Discrete Variational Calculus
Marseille (France)

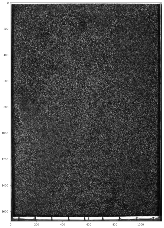
Background and motivation on edge detection



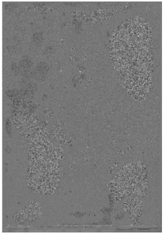
Multiphase flow of gas+liquid



at t_0 (liquid only)



at t_i (gas+liquid)



$\frac{(t_i)}{(t_0)}$

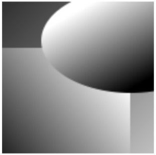
- Goal: estimate the interface between gas and liquid
- Datasize: large image (1677×1160 pixels), analysis need to be performed on a sequence of images.

Discrete Mumford-Shah model and related approaches

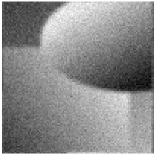
Degradation model:

$$\mathbf{z} = A\overline{\mathbf{u}} + n,$$

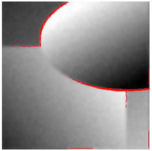
- $N = N_r N_c$: number of pixels
- $\mathbf{z} \in \mathbb{R}^M$: degraded image
- $\overline{\mathbf{u}} \in \mathbb{R}^N$: original image
- $A \in \mathbb{R}^{M \times N}$: linear degradation operator (e.g. a blur)
- $\mathbf{e} \in \mathbb{R}^{|\mathbb{E}|}$: edges detected, where $|\mathbb{E}| = N - N_r - N_c$
- $n \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_M)$: additive white gaussian noise



$\overline{\mathbf{u}}$



$\mathbf{z}$



$\widehat{\mathbf{u}}$
$\widehat{\mathbf{e}}$ (colored in red on edges between pixels)

→ Goal: Jointly recover $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{e}}$.

Related approaches, DMS-like model

- Estimate $\widehat{\mathbf{e}}$ from $\widehat{\mathbf{u}}$:

$$\widehat{\mathbf{u}} \in \underset{\mathbf{u}\in\mathbb{R}^N}{\operatorname{Argmin}} \;\; \frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2 + \lambda\mathcal{P}(\mathbf{u}).$$

  - Thresholded-ROF based on TV denoising [Cai, Steidl, 2013]: $\mathcal{P}(\mathbf{u}) = \|D_0\mathbf{u}\|_{1,2}$
  - Blake-Zisserman model: $\mathcal{P}(\mathbf{u}) = \sum_i \min(|(D\mathbf{u})_i|^p, \alpha^p)$,
    [Strekalovskiy, Cremers, 2014] [Hohm et al., 2015]
  - Potts model: $\mathcal{P}(\mathbf{u}) = \|\nabla\mathbf{u}\|_0$.
    [Storath, Weinmann, 2015]

- Jointly estimate $\widehat{\mathbf{e}}$ and $\widehat{\mathbf{u}}$ using Mumford-shah model (discrete formulation):

  [Foare, Lachaud, Talbot, 2016] [Foare, Pustelnik, Condat, 2019]

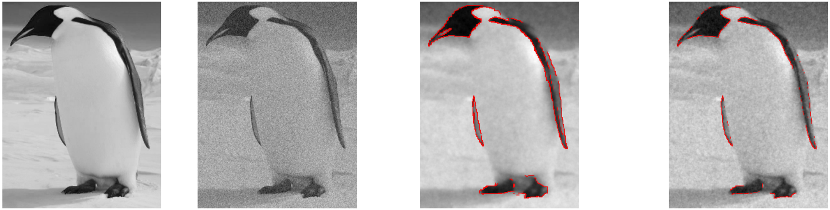
$$(\widehat{\mathbf{u}},\widehat{\mathbf{e}}) \in \operatorname{Argmin} \Phi(\mathbf{u},\mathbf{e}) := \frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2+\beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2+\lambda\mathcal{R}(\mathbf{e})$$

In the following we focus on this minimization problem

Proposed Discrete-Mumford-Shah model

$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \; \Phi(\mathbf{u},\mathbf{e}) := \frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2 + \beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2 + \lambda\mathcal{R}(\mathbf{e}).$$

- $\mathcal{R}$: edges penalization.
- $D_0 \in \mathbb{R}^{|\mathbb{E}|\times N}$: finite difference operator.
- $\mathbf{e} \in \mathbb{R}^{|\mathbb{E}|}$: edges between nodes whose value is 1 when a contour change is detected, and 0 otherwise.
- $\beta > 0$ and $\lambda > 0$: regularization parameters.

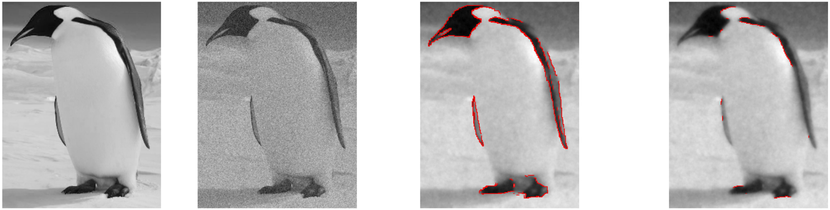


$\overline{\mathbf{u}}$ | $\mathbf{z}$ | $\widehat{\mathbf{u}}$, $\widehat{\mathbf{e}}$ $\beta=10, \lambda=0.01$ | $\widehat{\mathbf{u}}$, $\widehat{\mathbf{e}}$ $\beta=4, \lambda=0.01$

Proposed Discrete-Mumford-Shah model

$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \; \Phi(\mathbf{u},\mathbf{e}) := \frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2 + \beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2 + \lambda\mathcal{R}(\mathbf{e}),$$

- $\mathcal{R}$: edges penalization.
- $D_0 \in \mathbb{R}^{|\mathbb{E}|\times N}$: finite difference operator.
- $\mathbf{e} \in \mathbb{R}^{|\mathbb{E}|}$: edges between nodes whose value is 1 when a contour change is detected, and 0 otherwise.
- $\beta > 0$ and $\lambda > 0$: regularization parameters.



$\overline{\mathbf{u}}$ | $\mathbf{z}$ | $\widehat{\mathbf{u}}$, $\widehat{\mathbf{e}}$ $\beta=10, \lambda=0.01$ | $\widehat{\mathbf{u}}$, $\widehat{\mathbf{e}}$ $\beta=10, \lambda=0.05$

Choice of $\mathcal{R}(\mathbf{e})$

$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}}\ \Phi(\mathbf{u},\mathbf{e}) := \frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2 + \beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2 + \lambda\mathcal{R}(\mathbf{e}),$$

- $\mathcal{R}$: favors sparse solution on edges variable and convex.
  1. Ambrosio-Tortorelli approximation [Foare et al., 2016]:

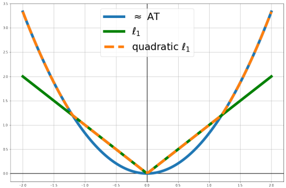
     $$\mathcal{R}(\mathbf{e}) = \frac{1}{4\varepsilon}\|\mathbf{e}\|_2^2 + \varepsilon\|D_1\mathbf{e}\|_2^2$$

  2. ℓ_1-norm: [Foare, Pustelnik, Condat, 2019] :

     $$\mathcal{R}(\mathbf{e}) = \|\mathbf{e}\|_1$$

  3. Quadratic-ℓ_1 [Foare, Pustelnik, Condat, 2019] :

     $$\mathcal{R}(\mathbf{e}) = \sum_{i=1}^{|\mathbb{E}|}\max\left\{|e_i|, \frac{e_i^2}{4\varepsilon}\right\}$$



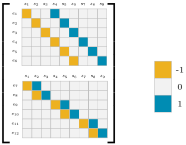
Design of D_0 and D_1

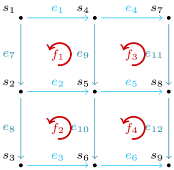
$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \ \Phi(\mathbf{u},\mathbf{e}) := \frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2 + \beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2 + \lambda\mathcal{R}(\mathbf{e}),$$

$$\text{with } \mathcal{R}(\mathbf{e}) = \frac{1}{4\varepsilon}\|\mathbf{e}\|_2^2 + \varepsilon\|D_1\mathbf{e}\|_2^2$$

- $D_0 : \mathbb{R}^N \to \mathbb{R}^{|\mathbb{E}|}$ maps $\mathbf{u}$ into a pair of $\mathbf{e} = (\mathbf{e}_h, \mathbf{e}_v)$ can be expressed by:

$$D_0 = \begin{bmatrix} H \\ V \end{bmatrix} =$$

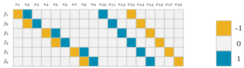




$(N = 9, |E| = 12)$

- $D_1 \in \mathbb{R}^{|\mathbb{E}|} \to \mathbb{R}^N$,

$$D_1 = \begin{bmatrix} V & -H \end{bmatrix} =$$



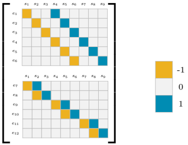
Design of D_0 and D_1

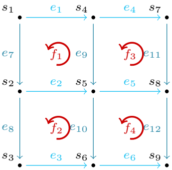
$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}}\ \Phi(\mathbf{u},\mathbf{e}) := \frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2 + \beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2 + \lambda\mathcal{R}(\mathbf{e}),$$

$$\text{with } \mathcal{R}(\mathbf{e}) = \frac{1}{4\varepsilon}\|\mathbf{e}\|_2^2 + \varepsilon\|D_1\mathbf{e}\|_2^2$$

- $D_0 : \mathbb{R}^N \to \mathbb{R}^{|\mathbb{E}|}$ maps $\mathbf{u}$ into a pair of $\mathbf{e} = (\mathbf{e}_h, \mathbf{e}_v)$ can be expressed by:

$$D_0 = \begin{bmatrix} H \\ V \end{bmatrix} =$$

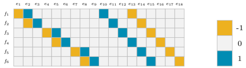




$(N = 9, |E| = 12)$

- $D_1 \in \mathbb{R}^{|\mathbb{E}|} \to \mathbb{R}^N$,

$$D_1 = \begin{bmatrix} V & -H \end{bmatrix} =$$



Proximal operator

Let $f : \mathcal{H} \to]-\infty, +\infty]$, convex, proper and lower semicontinuous, possibly non-smooth (i.e $\Gamma_0(\mathcal{H})$)

$$\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x} \in \mathcal{H}} f(\mathbf{x})$$

Proximal operator

Let $f : \mathcal{H} \to]-\infty, +\infty]$, convex, proper and lower semicontinuous, possibly non-smooth

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathcal{H}}{\text{Argmin}} \; f(\mathbf{x})$$

Definition

$f \in \Gamma_0(\mathcal{H})$, when $\mathcal{H}$ models a real Hilbert space, the proximity operator of f is defined by, for some $\tau > 0$:

$$(\forall \mathbf{x} \in \mathcal{H}) \qquad \text{prox}_{\tau f}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{H}}{\text{argmin}} \; f(\mathbf{y}) + \frac{1}{2\tau} \|\mathbf{y} - \mathbf{x}\|_2^2,$$

Proximal operator

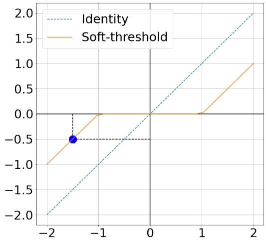
Definition

$f \in \Gamma_0(\mathcal{H})$, when $\mathcal{H}$ models a real Hilbert space, the proximity operator of f is defined by, for some $\tau > 0$:

$$(\forall \mathbf{x} \in \mathcal{H}) \qquad \mathrm{prox}_{\tau f}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{H}}{\mathrm{argmin}}\ f(\mathbf{y}) + \frac{1}{2\tau}\|\mathbf{y} - \mathbf{x}\|_2^2,$$

Example:

$f(x) = |x| \rightarrow \mathrm{prox}_{\tau f}(x) =$ soft-thresholding with a fixed threshold $\tau > 0$



Proximal algorithm: convex objective function

Let $f : \mathcal{H} \rightarrow]-\infty, +\infty]$, convex, proper and lower semicontinuous, possibly non-smooth

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathcal{H}}{\operatorname{Argmin}} \; f(\mathbf{x})$$

Definition

$f \in \Gamma_0(\mathcal{H})$, when $\mathcal{H}$ models a real Hilbert space, the proximity operator of f for some $\tau > 0$ is defined by:

$$(\forall \mathbf{x} \in \mathcal{H}) \qquad \operatorname{prox}_{\tau f}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{H}}{\operatorname{argmin}} \; f(\mathbf{y}) + \frac{1}{2\tau}\|\mathbf{y} - \mathbf{x}\|_2^2,$$

Proximal point iteration fo some $\tau > 0$:

$$\mathbf{x}^{[k+1]} = \operatorname{prox}_{\tau f}(\mathbf{x}^{[k]})$$

$\mathbf{x}^{[k]}$ converges to to a minimizer $\widehat{\mathbf{x}}$ of f [Bauschke, Combettes, 2011]

Difficulty: closed form of prox [cf. *proximity-operator.net*]

Proximal algorithm: convex objective function

f convex possibly non-smooth, g convex, differentiable and gradient L-Lipschitz

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathcal{H}}{\operatorname{Argmin}}\ f(\mathbf{x}) + \mathrm{g}(\mathbf{x})$$

- Forward-backward splitting algorithm:

$$\mathbf{x}^{[k+1]} := \mathbf{prox}_{\tau f}(\mathbf{x}^{[k]} - \tau \nabla\ \mathrm{g}(\mathbf{x}^{[k]}))$$

If $0 < \tau < 2/L$, this scheme converges to a minimizer of $f + g$.

Gauss-Seidel iterations: Non-convex objective function

f, h are convex, possibly non-smooth, g is convex w.r.t each variable and differentiable,

$$\underset{\mathbf{u}\in\mathbb{R}^N, \mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \ \Phi(\mathbf{u}, \mathbf{e}) := f(\mathbf{u}, \mathbf{z}) + g(\mathbf{u}, \mathbf{e}) + h(\mathbf{e})$$

- Gauss-seidel scheme = alternating minimization:

Set $(\mathbf{u}^{[0]}, \mathbf{e}^{[0]}) \in (\mathbb{R}^N, \mathbb{R}^{|\mathbb{E}|})$
for $k \in \mathbb{N}$ **do**
  $\mathbf{u}^{[k+1]} \in \underset{\mathbf{u}}{\text{Argmin}}\ \Phi(\mathbf{u}, \mathbf{e}^{[k]})$
  $\mathbf{e}^{[k+1]} \in \underset{\mathbf{e}}{\text{Argmin}}\ \Phi(\mathbf{u}^{[k+1]}, \mathbf{e})$

PAM iterations (Proximal Alternating Minimization)

f, h are convex, possibly non-smooth, g is convex w.r.t each variable and differentiable,

$$\underset{\mathbf{u}\in\mathbb{R}^N, \mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \ \Phi(\mathbf{u},\mathbf{e}) := f(\mathbf{u},\mathbf{z}) + g(\mathbf{u},\mathbf{e}) + h(\mathbf{e})$$

Algorithm 1: PAM [Attouch et al. 2010]

Set $(\mathbf{u}^{[0]}, \mathbf{e}^{[0]}) \in (\mathbb{R}^N, \mathbb{R}^{|\mathbb{E}|})$

for $k \in \mathbb{N}$ **do**

$$\mathbf{u}^{[k+1]} = \underset{\mathbf{u}}{\text{Argmin}} \ \Phi(\mathbf{u}, \mathbf{e}^{[k]}) + \frac{c_k}{2}\|\mathbf{u} - \mathbf{u}^{[k]}\|^2$$

$$\mathbf{e}^{[k+1]} = \underset{\mathbf{e}}{\text{Argmin}} \ \Phi(\mathbf{u}^{[k+1]}, \mathbf{e}) + \frac{d_k}{2}\|\mathbf{e} - \mathbf{e}^{[k]}\|^2$$

Convergence guarantees to a critical point

PAM iterations (Proximal Alternating Minimization)

f, h are convex, possibly non-smooth, g is convex w.r.t each variable and differentiable,

$$\underset{\mathbf{u}\in\mathbb{R}^N, \mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \; \Phi(\mathbf{u},\mathbf{e}) := f(\mathbf{u},\mathbf{z}) + g(\mathbf{u},\mathbf{e}) + h(\mathbf{e})$$

Algorithm 2: PAM [Attouch et al. 2010]

Set $(\mathbf{u}^{[0]}, \mathbf{e}^{[0]}) \in (\mathbb{R}^N, \mathbb{R}^{|\mathbb{E}|})$

for $k \in \mathbb{N}$ **do**

$$\mathbf{u}^{[k+1]} = \text{prox}_{\frac{1}{c_k}\Phi(\cdot,\mathbf{e}^{[k]})}(\mathbf{u}^{[k]})$$

$$\mathbf{e}^{[k+1]} = \text{prox}_{\frac{1}{d_k}\Phi(\mathbf{u}^{[k+1]},\cdot)}(\mathbf{e}^{[k]})$$

PAM iterations (Proximal Alternating Minimization)

$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \; \Phi(\mathbf{u},\mathbf{e}) := f(\mathbf{u},\mathbf{z}) + g(\mathbf{u},\mathbf{e}) + h(\mathbf{e})$$

Algorithm 3: PAM [Attouch et al. 2010]

Set $(\mathbf{u}^{[0]}, \mathbf{e}^{[0]}) \in (\mathbb{R}^N, \mathbb{R}^{|\mathbb{E}|})$

for $k \in \mathbb{N}$ **do**

$$\mathbf{u}^{[k+1]} = \text{prox}_{\frac{1}{c_k} f(\cdot;z) + \frac{1}{c_k} g(\cdot,\mathbf{e}^{[k]})}(\mathbf{u}^{[k]})$$

$$\mathbf{e}^{[k+1]} = \text{prox}_{\frac{1}{d_k} g(\mathbf{u}^{[k+1]},\cdot) + \frac{1}{d_k} h}(\mathbf{e}^{[k]})$$

Difficulty: Closed form expression for the prox of the sum of 2 functions.

$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \; \Phi(\mathbf{u},\mathbf{e}) := \underbrace{\frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2}_{f(\mathbf{u},\mathbf{z})} + \underbrace{\beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2}_{g(\mathbf{u},\mathbf{e})} + \underbrace{\lambda\mathcal{R}(\mathbf{e})}_{h(\mathbf{e})}$$

PALM iterations (Proximal Linearized Alternating Minimization)

$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \ \Phi(\mathbf{u},\mathbf{e}) := f(\mathbf{u},\mathbf{z}) + g(\mathbf{u},\mathbf{e}) + h(\mathbf{e})$$

Algorithm 4: PALM [Bolte, Sabach, Teboulle, 2013]

Initialization: $\mathbf{u}^{[0]} \in \mathbb{R}^N, \mathbf{e}^{[0]} \in \mathbb{R}^{|\mathbb{E}|}$

while $\frac{\Phi^{[k+1]}-\Phi^{[k]}}{\Phi^{[k]}} < 1\text{e}{-6}$ *and* $k \in \mathbb{N}$ **do**

- Set $\gamma > 1$ and $c_k = \gamma L_1(\mathbf{e}^{[k]})$;
- $\mathbf{u}^{[k+1]} = \text{prox}_{\frac{1}{2c_k}f(\cdot,\mathbf{z})}(\mathbf{u}^{[k]} - \frac{1}{c_k}\nabla_{\mathbf{u}} g(\mathbf{u}^{[k]},\mathbf{e}^{[k]}))$;
- Set $\eta > 1$ and $d_k = \eta L_2(\mathbf{u}^{[k+1]})$;
- $\mathbf{e}^{[k+1]} = \text{prox}_{\frac{1}{d_k}h}(\mathbf{e}^{[k]} - \frac{1}{d_k}\nabla_{\mathbf{e}} g(\mathbf{u}^{[k+1]},\mathbf{e}^{[k]}))$

Difficulty: closed form of prox in the update of the edge variable.

PALM iterations (Proximal Linearized Alternating Minimization)

The update of the edge variable in Algorithm 2 takes a closed form that is:

Proposition

- $\mathcal{R}(\mathbf{e}) = \|\mathbf{e}\|_1$:

$$\mathbf{e}^{[k+1]} = soft_{\frac{\lambda}{d_k}}(\mathbf{e}^{[k]})$$

- $\mathcal{R}(\mathbf{e}) = \frac{1}{4\varepsilon}\|\mathbf{e}\|_2^2 + \varepsilon\|D_1\mathbf{e}\|_2^2$:

$$\mathbf{e}^{[k+1]} = \left[\frac{2\lambda\varepsilon}{d_k} {D_1}^* D_1 + \left(1 + \frac{2\lambda}{4\varepsilon d_k}\right)\mathrm{Id}\right]^{-1} \mathbf{e}^{[k]}.$$

  - *Need to modify D_1 into periodic-D_1 to favor the computation of inverse operator above.*
  - *Efficiently inversion in Fourier if D_1 is circulant*

SL-PAM iterations

$$\underset{\mathbf{u}\in\mathbb{R}^N,\mathbf{e}\in\mathbb{R}^{|\mathbb{E}|}}{\text{minimize}} \; \Phi(\mathbf{u},\mathbf{e}) := \underbrace{\frac{1}{2}\|A\mathbf{u}-\mathbf{z}\|_2^2}_{f(\mathbf{u},\mathbf{z})} + \underbrace{\beta\|(1-\mathbf{e})\odot D_0\mathbf{u}\|_2^2}_{g(\mathbf{u},\mathbf{e})} + \underbrace{\lambda\mathcal{R}(\mathbf{e})}_{h(\mathbf{e})}$$

Algorithm 5: SL-PAM [Foare, Pustelnik, Condat, 2017]

Initialization: $\mathbf{u}^{[0]} \in \mathbb{R}^N, \mathbf{e}^{[0]} \in \mathbb{R}^{|\mathbb{E}|}$

while $\frac{\Phi^{[k+1]}-\Phi^{[k]}}{\Phi^{[k]}} < 1\mathrm{e}{-6}$ *and* $k \in \mathbb{N}$ **do**

- Set $\gamma > 1$ and $c_k = \gamma L_1(\mathbf{e}^{[k]})$
- $\mathbf{u}^{[k+1]} = \mathrm{prox}_{\frac{1}{c_k}f(\cdot,\mathbf{z})}\left(\mathbf{u}^{[k]} - \frac{1}{c_k}\nabla_{\mathbf{u}} g(\mathbf{e}^{[k]},\mathbf{u}^{[k]})\right)$
- Set $d_k > 0$
- $\mathbf{e}^{[k+1]} = \mathrm{prox}_{\frac{1}{d_k}(g(\cdot,\mathbf{u}^{[k+1]})+h)}\left(\mathbf{e}^{[k]}\right)$

→ Difficulty: computation of $\mathrm{prox}_{\frac{1}{d_k}(\beta\mathcal{S}(\cdot,\mathbf{u}^{[k+1]})+\lambda\mathcal{R})}\left(\mathbf{e}^{[k]}\right)$

SL-PAM iterations

→ Difficulty: computation of $\text{prox}_{\frac{1}{d_k}(\beta\mathcal{S}(\cdot,\mathbf{u}^{[k+1]})+\lambda\mathcal{R})}\left(\mathbf{e}^{[k]}\right)$

Proposition (Foare, Pustelnek, Condat, 2019)

We assume that $\mathcal{S}$ is separable, i.e:

$$(\forall \mathbf{e} = (\mathbf{e}_i)_{1\leq i\leq|\mathbb{E}|}) \qquad \mathcal{R}(\mathbf{e}) = \sum_{i=1}^{|\mathbb{E}|} \sigma_i(\mathbf{e}_i),$$

- $\sigma_i : \mathcal{R}^{|\mathbb{E}|} \to]-\infty, +\infty]$ *and whose proximity operator has a closed form expression*
- *Let $d_k > 0$, then at iteration $k \in \mathbb{N}$, the updating step on $\mathbf{e}^{[k+1]}$ in Algorithm 3 is:*

$$\mathbf{e}^{[k+1]} = \left(prox_{\frac{\lambda\sigma_i}{2\beta(D_0\mathbf{u}^{[k+1]})_i^2+d_k}} \frac{\beta(D_0\mathbf{u}^{[k+1]})_i^2 + \frac{d_k \mathbf{e}_i^{[k]}}{2}}{\beta(D_0\mathbf{u}^{[k+1]})_i^2 + \frac{d_k}{2}} \right)_{i\in\mathbb{E}}$$

SL-PAM iterations

Proposition (Le, Foare, Nelly, 2021)

When $\mathcal{R}(\mathbf{e}) = \frac{1}{4\varepsilon}\|\mathbf{e}\|_2^2 + \varepsilon\|D_1\mathbf{e}\|_2^2$, then the update of the edge variable in Algorithm 3 takes a closed form that is:

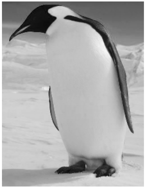
$$\mathbf{e}^{[k+1]} = \left[\frac{2\lambda\varepsilon}{d_k}D_1^*D_1 + \frac{2\beta}{d_k}\mathrm{diag}\left((D_0\mathbf{u}^{[k+1]})^2\right) + \dots \right.$$

$$\left. \left(\frac{\lambda}{2\varepsilon d_k} + 1\right)\mathrm{Id}\right]^{-1}\left(\frac{2\beta}{d_k}(D_0\mathbf{u}^{[k+1]})^2 + \mathbf{e}^{[k+1]}\right).$$

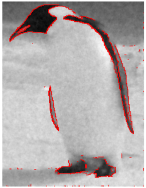
Difficulty: $D_1^*D_1$ and $\mathrm{diag}(D_0\mathbf{u}^{[k]})$ not-diagonalisable in the same basis.

Experiments

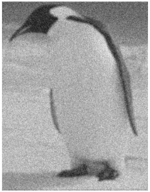
$N = 572 \times 558,\ \beta = 10,\ \lambda = 0.01$



$\bar{\mathbf{u}}$



SL-PAM ℓ_1



$\mathbf{z}$

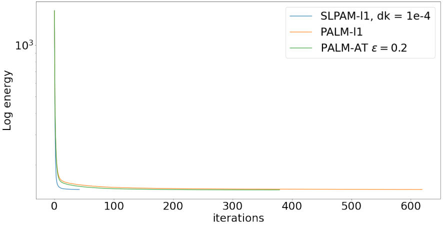
PALM-ℓ_1

PALM-AT$_{\varepsilon=0.2}$

Which scheme gives the best result?

Experiments

$N = 572 \times 558$, $\beta = 10$, $\lambda = 0.01$

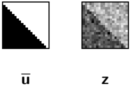


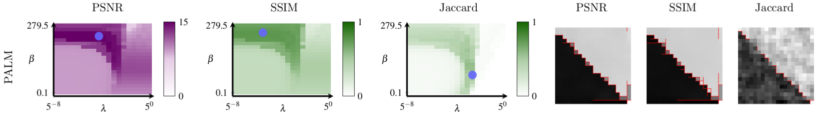
	CT	Number of iterations
SLPAM-l1	10s	43
PALM-l1	4min 01s	620
SLPAM-AT eps=0.2	NA	NA
PALM-AT eps=0.2	2 mins 27 s	380

Table: Computation time and number of iterations

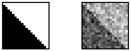
SL-PAM ℓ_1 is the fastest scheme. However, which scheme gives the best result in terms of contour detection?

Experiments

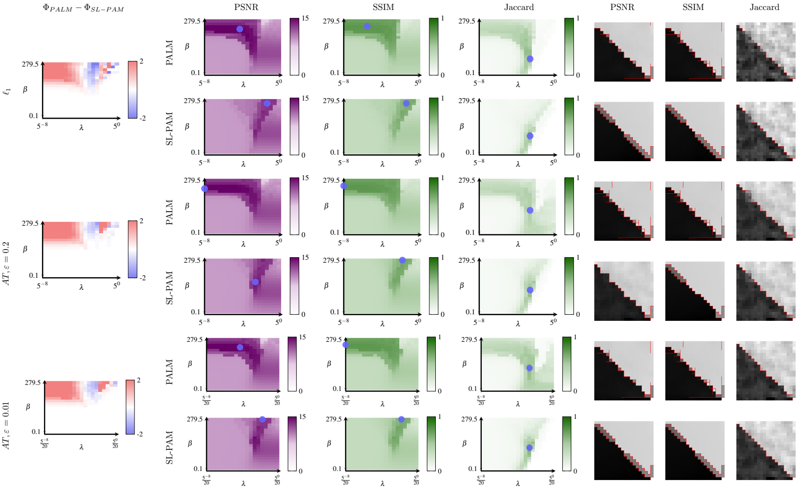




Experiments



$\bar{\mathbf{u}}$ $\mathbf{z}$



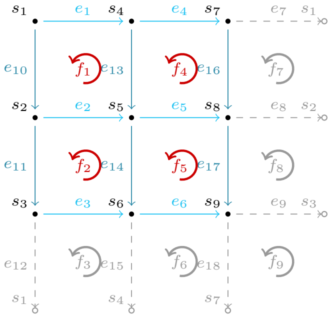
Conclusions and future works

- Discrete-Mumford-Shah and Ambrosio-Tortorelli models
- SL-PAM and PALM for solving both models
- SL-PAM fastest for ℓ_1 but not applicable for AT for large data set
- SL-PAM AT, SL-PAM-l1, PALM-AT, PALM-l1 very close performance in terms of contour detection
- Importance of β and λ choice

- Deeper convergence analysis of PALM and SL-PAM
- Application to multiphasic flow experiment leading to modifications in the data-term.

THANK YOU FOR YOUR ATTENTION!!!

PALM iterations



$H_{per} =$

$V_{per} =$