Variational Methods in Image Processing for Inpainting and Shadow Removal

Simone Parisotto

University of Verona
Department of Computer Science

March 13th, 2014

Motivation of our work

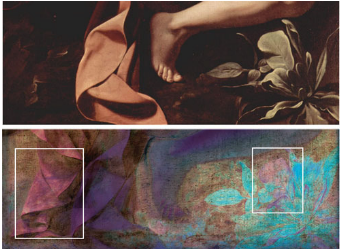
Describing and solving, in mathematical words, important and concrete applications.

Motivation of our work

Describing and solving, in mathematical words, important and concrete applications.

Many research fields have to treat with image processing:

- in cultural heritage;
  - *pentimenti;*



Caravaggio, *John the Baptist* (C. Daffara et al., 2011)

Motivation of our work

Describing and solving, in mathematical words, important and concrete applications.

Many research fields have to treat with image processing:

- in cultural heritage;
  - *pentimenti;*
  - *guide restoration;*

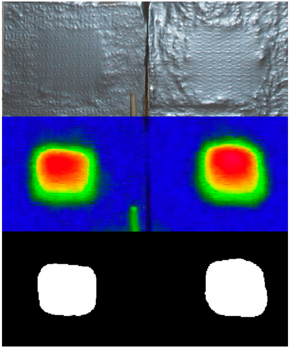
Neidhart von Reuental (C.B. Schönlieb, 2009)

Motivation of our work

Describing and solving, in mathematical words, important and concrete applications.

Many research fields have to treat with image processing:

- in cultural heritage;
  - *pentimenti;*
  - *guide restoration;*
  - *thermography;*



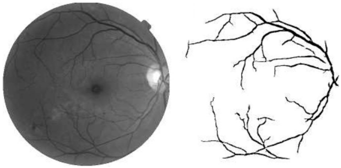
Detachments from wall (C. Daffara et al., 2010)

Motivation of our work

Describing and solving, in mathematical words, important and concrete applications.

Many research fields have to treat with image processing:

- in cultural heritage;
- in medical imaging;



VAMPIRE project (A. Giachetti et al., 2013)

Motivation of our work

Describing and solving, in mathematical words, important and concrete applications.

Many research fields have to treat with image processing:

- in cultural heritage;
- in medical imaging;
- in film restoration;

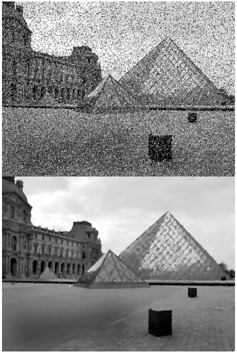
A. Buadès, S. Masnou et al. (2010)

Motivation of our work

Describing and solving, in mathematical words, important and concrete applications.

Many research fields have to treat with image processing:

- in cultural heritage;
- in medical imaging;
- in film restoration;
- in image denoising.



A. Chambolle, T. Pock (2010)

Motivation of our work

Describing and solving, in mathematical words, important and concrete applications.

Many research fields have to treat with image processing:

- in cultural heritage;
- in medical imaging;
- in film restoration;
- in image denoising.

In some of these projects I am still involved for further researches.

Plan of our work

This work aims to:

- Study relevant image processing tasks by variational and PDE methods:
  - to model inpainting;
  - to model shadow removal;
- See connection between the 2 problems;
- Show how to speed up the computational time for solving shadow removal;

Mathematical framework:

- Geometric Measure Theory;
- Functions of Bounded Variation;
- Sets of Finite Perimeters;
- Drift-Diffusion equation;

Inpainting - Criminisi et al. (2003)

Shadow Removal - Finlayson et al. (2006)

Basics on BV functions

Key idea: to describe images by gray level lines (all geometric information lie on edges)

- suitable setting: BV functions (natural for describing boundary discontinuities);

Distributional definition of BV function

Let $u :\in \Omega \subset \mathbb{R}^n \to \mathbb{R}$. Then $u \in \mathrm{BV}(\Omega)$ if Du is a vector Radon measure, i.e.

$$\langle Du, \vec{\varphi} \rangle = -\int_\Omega u \operatorname{div} \vec{\varphi} = \int_\Omega \vec{\varphi} \cdot \mathrm{d}Du \quad \forall \vec{\varphi} \in \left[C_0^\infty(\Omega)\right]^m$$

- Du is only concentrated on the boundaries;
- $Du = -\vec{\nu}|Du|$, with $|\nu| = 1$ and $|Du|$-a.e.

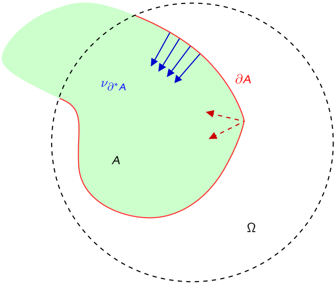
Total variation of u

$$|Du|(\Omega) = \sup \left\{ \int_\Omega u \operatorname{div} \vec{\varphi},\ \|\vec{\varphi}\|_\infty \leq 1,\ \vec{\varphi} \in C_c^\infty(\Omega)^n \right\}$$

- dealing with BV functions $\longrightarrow$ Sets of Finite Perimeters;

Sets of Finite Perimeters

A is of Finite Perimeter in Ω if and only if $\chi_A \in \mathrm{BV}(\Omega)$: $P(A;\Omega) \equiv |D\chi_A|(\Omega)$.



- $D\chi_A$ encodes all geometric information on $\partial A \cap \Omega$;
- $\partial^* A$ (reduced boundary) is $\mathcal{H}^{n-1}$-rectifiable;
- $D\chi_A = \nu_{\partial^* A}\mathcal{H}^{n-1} \llcorner \partial^* A$ so it is concentrated only on the boundaries;
- Gauss-Green: $\int_A \operatorname{div} \varphi = \int_{\partial^* A} \varphi \cdot \nu \, \mathrm{d}\mathcal{H}^{n-1}$;
- measure-theoretic notion of tangent space;

Coarea Formula

Let $u : \mathbb{R}^n \to \mathbb{R}$ a Lipschitz function, $A \subset \mathbb{R}^n$ open.

$$\int_A |\nabla u| = \int_{\mathbb{R}} P(\{u > t\};\, A)\, \mathrm{d}t \quad \text{as elements of } [0, \infty].$$

The total variation of a function is the accumulated surfaces of all its level sets.

Functions of Bounded Variation

- C. Jordan (1881): *functions with control on the oscillations (Fourier series);*
- M. Miranda (1964): $V(u,\Omega) = \sup\left\{ \int_\Omega u \,\mathrm{div}\, \varphi \,\mathrm{d}x : \varphi \in [C_c^1(\Omega)]^n,\ \|\varphi(x)\|_\infty \le 1 \right\}$

The $\mathrm{BV}(\Omega)$ space is a Banach space with the norm $\|u\|_{\mathrm{BV}(\Omega)} = \int_\Omega |u| \,\mathrm{d}x + |Du|(\Omega)$.

- lower semicontinuity: $V(u,\Omega) \le \liminf_{h\to\infty} V(u_h,\Omega)$;
- Convexity: $V(tu_1 + (1-t)u_2,\Omega) \le tV(u_1,\Omega) + (1-t)V(u_2,\Omega)$;
- $u \in [\mathrm{BV}(\Omega)]^m \iff V(u,\Omega) < \infty$;

Theorem (Federer-Vol'pert)

Let $u \in [\mathrm{BV}(\Omega)]^m$. The discontinuity set is $\mathcal{H}^{n-1}$-rectifiable and $\mathcal{H}^{n-1}(S_u \setminus J_u) = 0$. Then

$$Du = \underbrace{\nabla u(x)\,\mathrm{d}x}_{D^a u} + \underbrace{(u_+(x) - u_-(x)) \otimes \nu_u(x)\,\mathrm{d}\mathcal{H}^{n-1} \llcorner J_u}_{D^j u} + \underbrace{D^s u \llcorner (\Omega \setminus S_u)}_{D^c u}.$$

- $V(u,\Omega) \equiv |Du|(\Omega),\ \forall u \in [\mathrm{BV}(\Omega)]^m$;

Functions of Special Bounded Variation

- introduced by E. De Giorgi, L. Ambrosio (1988);
- good candidate where both volume and surface energies are involved;
- relevant for images;

SBV space

Let $u \in \mathrm{BV}(\Omega)$, then $u \in \mathrm{SBV}(\Omega)$ if $D^c u = 0$:

$$Du = D^a u + D^j u = \nabla u \mathcal{L}^n + (u^+ - u^-)\nu_u \mathcal{H}^{n-1} \llcorner J_u, \quad \forall u \in \mathrm{SBV}(\Omega).$$

$$\mathrm{W}^{1,1}(\Omega) \subset \mathrm{SBV}(\Omega) \subset \mathrm{BV}(\Omega).$$

- if $u \in \mathrm{W}^{1,1}(\Omega)$, or $u \in C^1(\Omega)$, then $Du = D^a u$;
- if $u = \chi_A$ and $|A| < \infty$, then $Du = D^j u$ (not Sobolev because $Du = \nu_A \mathcal{H}^{n-1} \llcorner \partial^* A$);
- if u is the Cantor-Vitali function, then $Du = D^c u$;

Motivational Example

We are involved in several *cutting* and *pasting* domains. Let $u, v \in [\mathrm{BV}(\Omega)]^m$.

- Is $w = u\chi_A + v\chi_{\Omega\setminus A} \in [\mathrm{BV}(\Omega)]^m$?
- Can Dw be expressed?

Let $u, v \in [\mathrm{BV}(\Omega)]^m$, $A \subset \Omega$ a set of finite perimeter with $\partial^* A \cap \Omega$ oriented by ν_A. Let $u^+_{\partial^* A}$, $v^-_{\partial^* A}$ (interior and exterior trace of u and v) given for $\mathcal{H}^{n-1}$-a.e. $x \in \partial^* A \cap \Omega$. Then

$$w = u\chi_A + v\chi_{\Omega\setminus A} \in [\mathrm{BV}(\Omega)]^m \iff \int_{\partial^* A\cap\Omega} \left|u^+_{\partial^* A} - v^-_{\partial^* A}\right| \mathrm{d}\mathcal{H}^{n-1} < \infty,$$
$$Dw = Du \llcorner A^1 + (u^+_{\partial^* A} - v^-_{\partial^* A}) \otimes \nu_A \mathcal{H}^{n-1} \llcorner (\partial^* A \cap \Omega) + Dv \llcorner A^0.$$

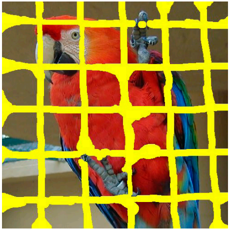
Let $u, v \in \mathrm{W}^{1,1}(\Omega) \cap \mathrm{L}^\infty(\Omega)$, $A \subset \Omega$ be a set of finite perimeter. Then

$$w = u\chi_A + v\chi_{\Omega\setminus A} \in \mathrm{SBV}(\Omega),$$
$$Dw = \left[\nabla u\chi_A + \nabla v\chi_{\Omega\setminus A}\right]\mathcal{L}^n + (\widetilde{u} - \widetilde{v})\nu_A \mathcal{H}^{n-1} \llcorner (\Omega \cap \partial^* A).$$

The inpainting problem

- very common in film restoration and image retouching;
- digital inpainting: retouching or recovering damaged ancient paintings (2001);
- we don't want to recover the true missing patch;
- we aim to create a new *natural* one;
- interpolation problem with unknown regularity degree (we are in BV space);
- geometric, sparse or exemplar-based approaches.

Original



Inpainting domain

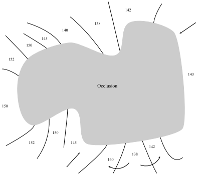
Bornemann (2007)

Geometric approach: The Euler's Elastica

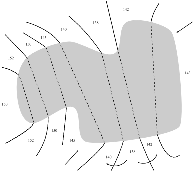
- image smoothness is expressed by *total variation* or *curvature* of level lines;
- the boundary data are propagated to predict the missing geometric structure;
- *local method* based on PDE but fails in presence of texture;
- Γ *Euler's elastica* if it is the equilibrium curve of the elasticity energy (1744):

$$E_2[\gamma] = \int_\gamma (a + b\kappa^2)\,\mathrm{d}s,$$

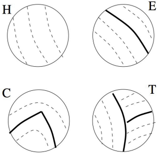
- from C.o.V., we obtain a fourth order equation: $2\kappa''(s) + \kappa^3(s) = \frac{a}{b}\kappa(s)$;



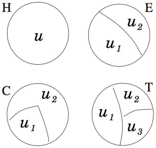
Occlusion



Possible connection



Situation



Approximation

- along any isophote $\gamma_\lambda : u \equiv \lambda$, the curvature of the oriented curve is given by

$$\kappa = \nabla \cdot \vec{n} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = \mathrm{div} \left(\frac{\nabla u}{|\nabla u|} \right) = \mathbf{H} = \kappa_1 + \kappa_2 \text{ (mean curvature)};$$

- $\mathrm{d}t$ is the length element along $\vec{n}$ so $\partial\lambda/\partial t = |\nabla u|$ or $\mathrm{d}\lambda = |\nabla u|\,\mathrm{d}t$;

$$J[u] = E[\mathcal{F}] = \int_0^1 \int_{\gamma_\lambda : u=\lambda} (a + b\kappa^2)\,\mathrm{d}s\,\mathrm{d}\lambda$$
$$\int_0^1 \int_{\gamma_\lambda : u=\lambda} \left(a + b\left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right)^2 \right) |\nabla u|\,\mathrm{d}t\,\mathrm{d}s$$
$$\int_D \left(a + b\left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right)^2 \right) |\nabla u|\,\mathrm{d}x, \text{ with } u \in \mathrm{BV}(\Omega).$$

- suitable boundary conditions;
- if $a/b = \infty$, then $TV(u) = \int_\Omega |\nabla u|$, with the condition $u|_{\Omega \setminus D} = u_0|_{\Omega \setminus D}$.

Theorem: The noise free TV inpainting model (Chan, Shen)

Suppose that $u_0 \in \mathrm{BV}(\Omega)$, $u_0 \subset [0, 1]$. Then the *noise free TV inpainting model* $TV(u)$, together with the gray value constraint $u \subset [0, 1]$, has one optimal inpainting at least.

Sparse approach based on (consistent) dictionaries

Input

Sparse approach based on (consistent) dictionaries

Hays - Efros

Exemplar-based approach: Variational Framework

- patches similarity within the image: correspondence maps $\varphi : O \to O^c$;
- the whole image is scanned (greedy algorithm but sensitive to the order);
- Demanet (2003): variational formulation searching $u(x) = \widehat{u}(\varphi(x))$, for $x \in O$:

$$E(\varphi) = \int_O \int_{\Omega_p} \left|\widehat{u}(\varphi(x+h)) - \widehat{u}(\varphi(x)+h)\right|^2 \mathrm{d}h\, \mathrm{d}x \quad \text{(non-convex);}$$

- Gilboa, Osher (2007): replace φ with weights $w(x,y)$, subject to $\int_{\widetilde{O}^c} w(x,y) = 1$;

Arias, Caselles, Facciolo (2011)

$$\min \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,y)\varepsilon\left(p_u(x) - p_{\widehat{u}}(y)\right) \mathrm{d}y\, \mathrm{d}x + T \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,y) \log(w(x,y))\, \mathrm{d}y\, \mathrm{d}x$$

- when $T \to 0$ the weights are the correspondence map: $w(x,\widehat{x}) = \delta(\widehat{x} - \varphi(x))$.
- NL Means: $\mathbb{P} \equiv \mathrm{L}^2(\Omega_p)$, $\varepsilon(p_u(x) - p_{\widehat{u}}(y)) = \|p_u(x) - p_{\widehat{u}}(y)\|_g^2$.
- NL Poisson: $\mathbb{P} \equiv \mathrm{W}^{1,2}(\Omega_p)$, $\varepsilon(p_u(x) - p_{\widehat{u}}(y)) = \|p_u(x) - p_{\widehat{u}}(y)\|_{\nabla,g}^2$.
- NL Gradient Medians: $\mathbb{P} \equiv \mathrm{BV}(\Omega_p)$, $\varepsilon(p_u(x) - p_{\widehat{u}}(y)) = \|p_u(x) - p_{\widehat{u}}(y)\|_{\nabla,g}$.

- solution's structure: rototranslation of patches;

NL-Poisson patch metric function

$$\mathcal{E}_{\nabla,T}(u,w) = \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,y) \| p_u(x) - p_{\widehat{u}} \|^2_{g,\nabla} \, \mathrm{d}y \, \mathrm{d}x + T \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,y) \log w(x,y) \, \mathrm{d}y,$$

Euler-Lagrange equations respect w and u ▸ Osmosis

$$w_{\varepsilon,T}(u)(x,y) = \frac{1}{Z_{\varepsilon,T}(u)(x)} \exp\left(-\frac{1}{T} \varepsilon\big(p_u(x) - p_{\widehat{u}}(y)\big) \right),$$

$$\begin{cases} \Delta u(z) = \operatorname{div} \mathbf{v}(w)(z), & z \in O, \\ u = \widehat{u}, & \text{in } \partial O, \end{cases} \implies \min \int_{\widetilde{O}} \| \nabla u(z) - \mathbf{v}(w)(z) \|_2^2 \, \mathrm{d}z.$$

Existence of minima for NL-Means and NL-Poisson

Existence of minima for NL-Means approach - Arias et al. (2011)

Assume $g \in C_c(\mathbb{R}^n)^+$, supp $g \in \Omega_p$, $\nabla g \in \mathrm{L}^1(\mathbb{R}^n)$ and $\widehat{u} \in \mathrm{BV}(O^c) \cap \mathrm{L}^\infty(O^c)$.

- If $(u_n, w_n) \in \mathcal{A}_2$ is a minimizing sequence for $\mathcal{E}_{2,T}$ such that u_n is uniformly bounded, then we may extract a subsequence converging to a minimum of $\mathcal{E}_{2,T}$.
- There exist a minimum $(u, w) \in \mathcal{A}_2$ of $\mathcal{E}_{2,T}$. For any minimum $(u, w) \in \mathcal{A}_2$ we have that $u \in W^{1,\infty}(O)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.

Existence of minima for NL-Poisson approach - Arias et al. (2011)

Assume $\widehat{u} \in \mathrm{W}^{2,2}(O^c) \cap \mathrm{L}^\infty(O^c)$, $g \in \mathrm{W}^{1,\infty}(\mathbb{R}^n)^+$, supp $g \in \Omega_p$ compact.

- There exists a solution of the variational problem $\min_{(u,w)\in\mathcal{A}_\nabla} \mathcal{E}_{\nabla,T}(u, w)$.
- For any solution $(u, w) \in \mathcal{A}_\nabla$ we have $u \in \mathrm{W}^{1,2}(O) \cap \mathrm{W}^{2,p}_{\mathrm{loc}}(O) \cap \mathrm{L}^\infty(O)$ for all $p \in [1, \infty]$ and $w \in \mathrm{W}^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.

Algorithms and Visual Results - Arias, Caselles, Facciolo (2011)

Some numerical details:

- Patchmatch (2009) for patch comparison: faster than kd-tree;
- because of high probability to fall in local minima: multiscale approach;

Alternating optimization for NL-means model

Input: u^0 with $\|u^0\|_\infty \le \|\widehat{u}\|_\infty$.
for each $k \in \mathbb{N}$ **do**
  $w^{k+1} = \arg\min_{w \in \mathcal{W}} \mathcal{E}_{2,T}(u^k, w)$,
  $u^{k+1} = \arg\min_{u} \mathcal{E}_{2,T}(u, w^{k+1})$.
end for

Alternating optimization for NL-Poisson model

Input: u^0 with $\|u^0\|_\infty \le \|\widehat{u}\|_\infty$.
for each $k \in \mathbb{N}$ **do**
  $w^{k+1} = \arg\min_{w \in \mathcal{W}} \mathcal{E}_{\nabla,T}(u^k, w)$,
  $u^{k+1} = \arg\min_{u \in W^{1,2},\, u|_{\partial O^c} = \hat{u}|_{\partial O^c}} \mathcal{E}_{\nabla,T}(u, w^{k+1})$.
end for

Original | KSY | NL-Means | NL-Poisson

Drift Diffusion PDE - Weickert (2013)

- Osmosis: omnipresent in nature (it transports water across membranes);
- diffusion (symmetric processes) leads to flat steady states;
- osmosis (nonsymmetric counterpart of diffusion) allows nonconstant steady states;
- a system is in a steady state for a property p if $\partial_t p = 0$.
- Fokker-Plank equation (time evolution of the p.d.f. of the velocity of a particle)

The continuous model

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta u - \operatorname{div}(\mathbf{d}u), & \text{on } \Omega \times (0,T] \\ u(\mathbf{x},0) = f(\mathbf{x}), & \text{on } \Omega \\ \langle \nabla u - \mathbf{d}u, \mathbf{n} \rangle = 0, & \text{on } \partial\Omega \times (0,T] \end{cases}$$

- preservation of the Average Grey Value;
- preservation of Positivity;
- convergence to Nontrivial Steady State when $\mathbf{d} = \nabla \log v$;

Associated minimization problem NL-Poisson

$$\min \int_\Omega v \left| \nabla \left(\frac{u}{v} \right) \right|^2 \mathrm{d}x, \text{ or } \min \int_\Omega |\nabla u - \mathbf{d}u|^2 \, \mathrm{d}x$$

Drift Diffusion PDE - Weickert (2013)

- Osmosis: omnipresent in nature (it transports water across membranes);
- diffusion (symmetric processes) leads to flat steady states;
- osmosis (nonsymmetric counterpart of diffusion) allows nonconstant steady states;
- a system is in a steady state for a property p if $\partial_t p = 0$.
- Fokker-Plank equation (time evolution of the p.d.f. of the velocity of a particle)

The discrete model

$$
\begin{aligned}
\frac{\partial u_{i,j}}{\partial t} = &\left(\frac{1}{h^2} - \frac{d_{1,i+\frac{1}{2},j}}{2h}\right) u_{i+1,j} + \left(\frac{1}{h^2} + \frac{d_{1,i-\frac{1}{2},j}}{2h}\right) u_{i-1,j} \\
&+ \left(\frac{1}{h^2} - \frac{d_{2,i,j+\frac{1}{2}}}{2h}\right) u_{i,j+1} + \left(\frac{1}{h^2} + \frac{d_{2,i,j-\frac{1}{2}}}{2h}\right) u_{i,j-1} \\
&+ \left(-\frac{4}{h^2} - \frac{d_{1,i+\frac{1}{2},j}}{2h} + \frac{d_{1,i-\frac{1}{2},j}}{2h} - \frac{d_{2,i,j+\frac{1}{2}}}{2h} + \frac{d_{2,i,j-\frac{1}{2}}}{2h}\right) u_{i,j} = P[u_{i,j}].
\end{aligned}
$$

- suppose to know shadow boundaries for applications we have in mind;

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \operatorname{div}(\mathbf{d}u) \implies \partial_t u = Pu \text{ (only } P \text{ needed)}$$

- Exponential Integrators;

$$\begin{cases} \mathbf{y}'(t) = A\mathbf{y}(t) + \mathbf{b}(t, \mathbf{y}(t)), & t > t_0 \\ \mathbf{y}(t_0) = \mathbf{y}_0, \end{cases}$$

whose analytical solution is, with $\varphi_1(z) = (e^z - 1)/z$

$$\mathbf{y}(t) = \exp((t - t_0)A)\mathbf{y}_0 + \int_{t_0}^{t} \exp((t-\tau)A)\mathbf{b}(\tau, \mathbf{y}(\tau))\, d\tau,$$

$$\mathbf{y}(t) = \exp((t - t_0)A)\mathbf{y}_0 + (t - t_0)\varphi_1((t - t_0)A)\mathbf{b} = \mathbf{y}_0 + (t - t_0)\varphi_1((t - t_0)A)(A\mathbf{y}_0 + \mathbf{b}).$$

- no need to compute $\exp(P)$ but $\exp(P)u$ (Krylov methods for P big and sparse)

$$A = V_m H_m V_m^T \implies \exp(A)V_m \approx V_m \exp(H_m) \implies \exp(A)v \approx V_m \exp(H_m)e_1$$

- Euler exponential method is exact if $b(\mathbf{y}(t)) = b(\mathbf{y}_0) \equiv \mathbf{b}$ or of order 1 otherwise.
- scripts from *Al-Mohy and Higham(2011)* and *Sidje(1998)* have been tested;

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

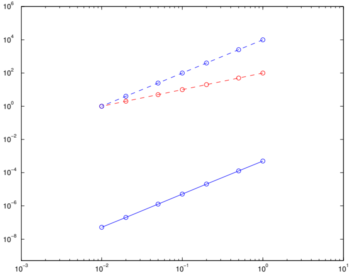
- Exponential Integrators;
- θ-method with direct method: LUpq factorization

```
[L,U,p,q]=lu(I-dt*theta*A,'vector');
B = (I+dt*(1-theta)*A);
for t = (dt:dt:T)
  C = B*y;
  y(q) = U\(L\(noto(p)));
end
```

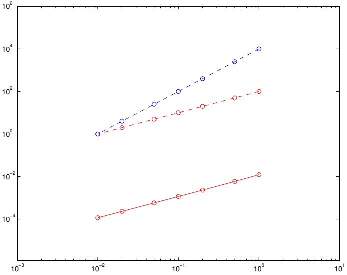
Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization



LUpq, $\theta = 0.5$



LUpq, $\theta = 1$

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \operatorname{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
  - BiCGStab: standard, fixed timestep dt until T fixed is reached;

```
y=bicgstab(I-dt*theta*A,(I+dt*(1-theta)*A)*y,tol,maxit);
y=bicgstab(I-dt*theta*A,(I+dt*(1-theta)*A)*y,tol,maxit,[],[],y);
y=bicgstab(I-dt*theta*A,(I+dt*(1-theta)*A)*y,tol,maxit,L,U,y);
```

It is not satisfactory at all (first steps are the most important ones - far away from steady state).

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
  - BiCGStab: standard, fixed timestep dt until T fixed is reached;
  - A-BiCGStab: adaptative, variable timestep dt until T fixed is reached;

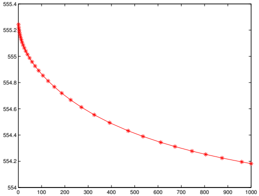
```
% k = number of iterations in BiCGStab
averit = 35*maxit/50;
safe_zone = [0.8*averit,1.2*averit];
```

$$\texttt{dt(t+1)} = \begin{cases} \texttt{1.2*dt(t)} & \texttt{if k<min(safe\_zone) steps} \\ \texttt{1.0*dt(t)} & \texttt{if min(safe\_zone)<k<max(safe\_zone) steps} \\ \texttt{0.8*dt(t)} & \texttt{if k>max(safe\_zone) steps} \\ \texttt{0.5*dt(t)} & \texttt{otherwise (don't increase t).} \end{cases}$$

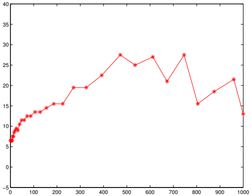
Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

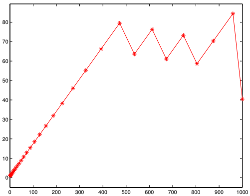
- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
  - BiCGStab: standard, fixed timestep dt until T fixed is reached;
  - A-BiCGStab: adaptative, variable timestep dt until T fixed is reached;



A-BiCGStab: `norm(y)`



A-BiCGStab: Iters

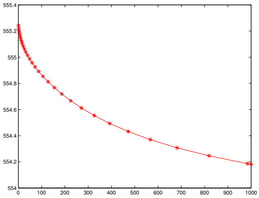


A-BiCGStab: dt

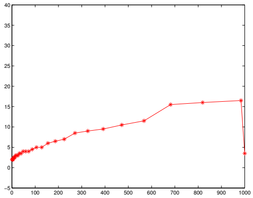
Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{d}t\theta P)^{-1}(I + \mathrm{d}t(1-\theta)P)u^t$$

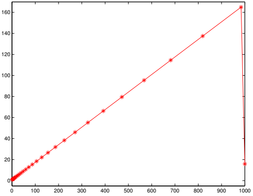
- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
  - BiCGStab: standard, fixed timestep $\mathrm{d}t$ until T fixed is reached;
  - A-BiCGStab: adaptative, variable timestep $\mathrm{d}t$ until T fixed is reached;



A-BiCGStab+ilu: `norm(y)`



A-BiCGStab+ilu: Iters



A-BiCGStab+ilu: $\mathrm{d}t$

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
  - BiCGStab: standard, fixed timestep dt until T fixed is reached;
  - A-BiCGStab: adaptative, variable timestep dt until T fixed is reached;
  - F-BiCGStab: standard, fixed timestep dt until exit condition is true;

```
norm(y_new-y)/norm(y_new) < dt * tol_exit;
```

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
  - BiCGStab: standard, fixed timestep dt until T fixed is reached;
  - A-BiCGStab: adaptative, variable timestep dt until T fixed is reached;
  - F-BiCGStab: standard, fixed timestep dt until exit condition is true;
  - FA-BiCGStab: adaptative, variable timestep dt until exit condition is true;

```
norm(y_new-y)/norm(y_new) < dt(t) * tol_exit;
```

Solving the Drift-Diffusion PDE for Shadow Removal

$$u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t;$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
- semi-Fourier collocation, full-Fourier collocation;
  - mirror the image to guess the periodic boundary condition;
  - large timestep: error from reference is upper bounded by Gibbs phenomenon on high jumps of colours;
  - **d** can be modified when/where necessary (e.g. $\mathbf{d} = \mathbf{d}. * u_{\text{mask}}$);

Solving the Drift-Diffusion PDE for Shadow Removal

$$u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t;$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
- semi-Fourier collocation, full-Fourier collocation;

$$(I - \mathrm{dt}\theta_1\Delta - \mathrm{dt}\theta_2 D)u^{t+1} = (I + \mathrm{dt}(1-\theta_1)\Delta + \mathrm{dt}(1-\theta_2)D)u^t;$$

Algorithm 1: Semi-implicit solver with bridge Fourier collocation

Input: u^0 (original image, $2\mathrm{D}$ matrix of N rows and M columns), k pixel-indexes.
Output: u^T at time $T = t^{\mathrm{end}}$.
1: coeff $= (1 + 4k\pi^2\mathrm{dt})$;
2: $A = (I + \mathrm{dt}D)$;
3: **for** $t = \mathrm{dt} : \mathrm{dt} : T$ **do**
4: $u = \mathrm{reshape}(Au(:), N, M)$; $\hat{v} = \mathrm{fft2}(u)./\mathrm{coeff}$;
5: $u = \mathrm{ifft2}(\hat{v})$;
6: **end for**

Solving the Drift-Diffusion PDE for Shadow Removal

$$u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t;$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
- semi-Fourier collocation, full-Fourier collocation;

$$u^{t+1} = (I - \mathrm{dt}\Delta)^{-1}(I + \mathrm{dt}D)u^t;$$

Algorithm 1: Semi-implicit solver with bridge Fourier collocation

Input: u^0 (original image, $2\mathrm{D}$ matrix of N rows and M columns), k pixel-indexes.
Output: u^T at time $T = t^{\mathrm{end}}$.

1: coeff $= (1 + 4k\pi^2\mathrm{dt})$;
2: $A = (I + \mathrm{dt}D)$;
3: **for** $t = \mathrm{dt} : \mathrm{dt} : T$ **do**
4: $\quad u = \mathrm{reshape}(Au(:), N, M)$; $\hat{v} = \mathrm{fft2}(u)./\mathrm{coeff}$;
5: $\quad u = \mathrm{ifft2}(\hat{v})$;
6: **end for**

Solving the Drift-Diffusion PDE for Shadow Removal

$$u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t;$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
- semi-Fourier collocation, full-Fourier collocation;

$$u = \sum_{|k|_\infty \le N} u_k \mathrm{e}^{2\mathrm{i}\pi x\cdot k} \quad \text{and} \quad \Delta u = -\sum_{|k|_\infty \le N} u_k 4\pi^2 |k|^2 \mathrm{e}^{2i\pi x\cdot k};$$

Algorithm 1: Semi-implicit solver with bridge Fourier collocation

Input: u^0 (original image, $2\mathrm{D}$ matrix of N rows and M columns), k pixel-indexes.
Output: u^T at time $T = t^{\text{end}}$.

1: coeff $= (1 + 4k\pi^2\mathrm{dt})$;
2: $A = (I + \mathrm{dt}D)$;
3: **for** $t = \mathrm{dt} : \mathrm{dt} : T$ **do**
4: $\quad u = \text{reshape}(Au(:), N, M)$; $\hat{v} = \text{fft2}(u)./\text{coeff}$;
5: $\quad u = \text{ifft2}(\hat{v})$;
6: **end for**

Solving the Drift-Diffusion PDE for Shadow Removal

$$u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t;$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
- semi-Fourier collocation, full-Fourier collocation;

$$\sum_{|k|_\infty \leq N} v_k \mathrm{e}^{2\mathrm{i}\pi x\cdot k} = v = (I - \mathrm{dt}\Delta)u = \sum_{|k|_\infty \leq N} (1 + \mathrm{dt}4\pi^2|k|^2)u_k \mathrm{e}^{2\mathrm{i}\pi x\cdot k};$$

Algorithm 1: Semi-implicit solver with bridge Fourier collocation

Input: u^0 (original image, $2\mathrm{D}$ matrix of N rows and M columns), k pixel-indexes.
Output: u^T at time $T = t^{\mathrm{end}}$.

1: coeff $= (1 + 4k\pi^2\mathrm{dt})$;
2: $A = (I + \mathrm{dt}D)$;
3: **for** $t = \mathrm{dt} : \mathrm{dt} : T$ **do**
4: u = reshape$(Au(:), N, M)$; $\hat{v}$ = fft2(u)./coeff;
5: u = ifft2$(\hat{v})$;
6: **end for**

Solving the Drift-Diffusion PDE for Shadow Removal

$$u^{t+1} = (I - \mathrm{dt}\theta P)^{-1}(I + \mathrm{dt}(1-\theta)P)u^t;$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
- semi-Fourier collocation, full-Fourier collocation;

$$u_k = \frac{v_k}{(1+\mathrm{dt}4\pi^2|k|^2)}.$$

Algorithm 1: Semi-implicit solver with bridge Fourier collocation

Input: u^0 (original image, $2\mathrm{D}$ matrix of N rows and M columns), k pixel-indexes.
Output: u^T at time $T = t^{\mathrm{end}}$.

1: coeff $= (1 + 4k\pi^2\mathrm{dt})$;
2: $A = (I + \mathrm{dt}D)$;
3: **for** $t = \mathrm{dt} : \mathrm{dt} : T$ **do**
4: $\quad u = \mathrm{reshape}(Au(:), N, M)$; $\hat{v} = \mathrm{fft2}(u)./\mathrm{coeff}$;
5: $\quad u = \mathrm{ifft2}(\hat{v})$;
6: **end for**

Solving the Drift-Diffusion PDE for Shadow Removal

$$\partial_t u = \Delta u - \mathrm{div}(\mathbf{d}u) \implies u^{t+1} - \mathrm{dt}\Delta u^{t+1} = u^t - \mathrm{dt}\,\mathrm{div}((\nabla \log u)\, u^t);$$

- Exponential Integrators;
- θ-method with direct method: LUpq factorization;
- θ-method with iterative method: BiCGStab and variants;
- semi-Fourier collocation, full-Fourier collocation;

Algorithm 2: Semi-implicit solver with fully Fourier collocation

Input: u^0 (original image, $2\mathrm{D}$ matrix of N rows and M columns), k pixel-indexes.
Output: u^T at time $T = t^{\text{end}}$.

define the `flag_log` $= \{0, 1\}$ variable, useful to change the computation of $\mathbf{d}$.
if `flag_log` **then**
  $\mathbf{d} = \nabla \log u = \mathrm{ifft2}(\mathrm{fft2}(\log u).*(2\pi \mathrm{i} k))$
else
  $\mathbf{d} = \nabla u./u = (\mathrm{ifft2}(\mathrm{fft2}(u).*(2\pi \mathrm{i} k)))./u;$
end if
$\mathrm{coeff} = (1 - 4k\pi^2 \mathrm{dt});$
for $t = \mathrm{dt} : \mathrm{dt} : T$ **do**
  $\mathrm{div}(\mathbf{d}u) = \mathrm{ifft2}((\mathrm{fft2}(\mathbf{d}u).*(2\pi \mathrm{i} k)));\quad u = \mathrm{ifft2}(\mathrm{fft2}(u - \mathrm{dt}\,\mathrm{div}(\mathbf{d}u))./\mathrm{coeff})$
end for

Numerical Results

Parameters: dt = 1 (*for Fourier (F), dt = 100), `tol_bicgstab`=10^{-06}, `tol_exit`=10^{-06} and maxit = 30.

θ = 0.5	**LUpq**	**BiCGStab**				**BiCGStab + ilu**			
		-	A	F	FA	-	A	F	FA
T	1000	1000	1000	5963	7177.26	1000	1000	6933	7579.16
I	-	2968.5	1359	15394	4082	1585	516	9979.5	1485
R	-	0	0	0	4	0	0	0	2
E	1.47e-08	1.68e-04	1.19e-03	-	-	1.03e-04	1.21e-03	-	-
C	86.80	138.12	24.43	804.32	73.34	192.73	19.37	1308.37	51.36

θ = 1	**LUpq**	**BiCGStab**				**BiCGStab + ilu**			
		-	A	F	FA	-	A	F	FA
T	1000	1000	1000	5618	7798.39	1000	1000	6935	7856.53
I	-	3306.5	1717	15897.5	4529.5	1785	753	10180.5	2298.5
R	-	0	1	0	13	0	0	0	2
E	4.59e-05	1.47e-04	4.53e-03	-	-	2.04e-04	3.94e-03	-	-
C	84.27	125.36	30.04	679.08	83.84	181.92	25.63	1203.16	75.19

	Ref. `expmv`	**LUpq**	**BiCGStab**	**BiCGStab + ilu**	`expv`	**F. Alg. 1**	**F. Alg. 2**
T	1000	1000	1000	1000	1000	1000*	1000*
θ	-	1	0.5	0.5	-	-	-
I	-	-	1359	516	-	-	-
R	-	-	0	0	-	-	-
E	-	4.59e-05	1.19e-03	1.21e-03	1.73e-04	0.1275	0.1080
C	206.26	84.27	24.43	19.37	25.07	4.80	10.94

Numerical Results

Parameters: $\mathrm{dt} = 1$ (*for Fourier (F), $\mathrm{dt} = 100$), `tol_bicgstab`=10^{-07}, `tol_exit`=10^{-07} and $\mathrm{maxit} = 30$.

$\theta = 0.5$	**LUpq**	**BiCGStab**				**BiCGStab + ilu**			
		-	A	F	FA	-	A	F	FA
T	1000	1000	1000	16619	23061.54	1000	1000	20430	24228.54
I	-	3398.5	1872	38248	11660	3066	680.5	27837.5	3859
R	-	0	4	0	24	0	0	0	7
E	1.47e-08	2.24e-05	4.09e-03	-	-	9.58e-06	1.20e-03	-	-
C	86.80	147.61	35.2	1996	215.72	233.46	24.33	3428.48	130.35

$\theta = 1$	**LUpq**	**BiCGStab**				**BiCGStab + ilu**			
		-	A	F	FA	-	A	F	FA
T	1000	1000	1000	15358	22299.95	1000	1000	20433	21338.26
I	-	4032.5	2407.5	39377.5	10806	3332.5	1036	31189	5774.5
R	-	0	4	0	47	0	0	0	6
E	4.59e-05	6.59e-05	4.80e-03	-	-	7.21e-05	3.22e-03	-	-
C	84.27	137.13	43.44	1748.14	209.73	225.12	34.25	3240.42	187.02

	Ref. `expmv`	**LUpq**	**BiCGStab**	**BiCGStab + ilu**	`expv`	**F. Alg. 1**	**F. Alg. 2**
T	1000	1000	1000	1000	1000	1000*	1000*
θ	-	1	0.5	0.5	-	-	-
I	-	-	1872	680.5	-	-	-
R	-	-	4	0	-	-	-
E	-	4.59e-05	4.09e-03	1.20e-03	1.73e-04	0.1275	0.1080
C	206.26	84.27	35.2	24.33	25.07	4.80	10.94

Input

Reference with `expmv.m`

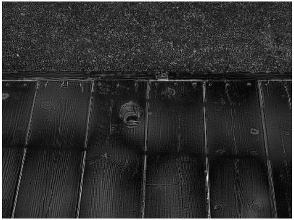
LUpq, $\theta = 1$

A-BiCGStab + ilu, $\theta = 0.5$

Input

Reference with `expmv.m`

Fourier Alg. 1

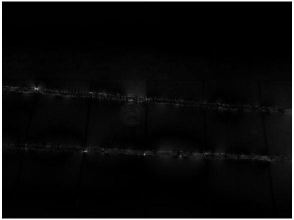


Error Alg. 1

Input

Reference with `expmv.m`

Fourier Alg. 2 with $\mathbf{d} = \nabla u/u$

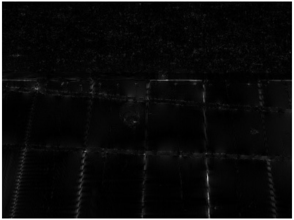


Error Alg. 2 with $\mathbf{d} = \nabla u/u$

Input

Reference with `expmv.m`

Fourier Alg. 2 with $\mathbf{d} = \nabla \log u$



Error Alg. 2 with $\mathbf{d} = \nabla \log u$

Input

F-BiCGStab, $\theta = 0.5$

F-BiCGStab + ilu, $\theta = 0.5$

Input

FA-BiCGStab, $\theta = 0.5$

FA-BiCGStab + ilu, $\theta = 0.5$

Other application: seamless image cloning

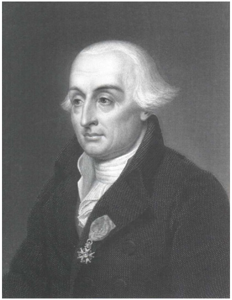
- fuse incompatible information - *Poisson Image Editing*, Perez (2003);
- interpolant f_2 of f_1 over Γ is the solution of

$$\text{(Euler - Lagrange)}\ \Delta f_2 = 0,\ \text{on } \Gamma,\ \text{with } f_2 = f_1,\ \text{on } \partial\Gamma \implies \text{blurred};$$

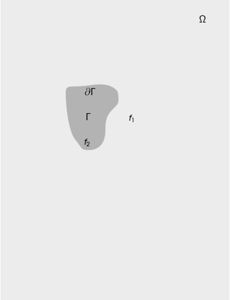
- *guidance vector field* **p**:

$$\text{(Euler - Lagrange)}\ \Delta f_2 = \operatorname{div}\mathbf{p},\ \text{on } \Gamma,\ \text{with } f_2 = f_1,\ \text{on } \partial\Gamma \implies p = \nabla f_2;$$

Euler



Lagrange



Notation

Input

Other application: seamless image cloning

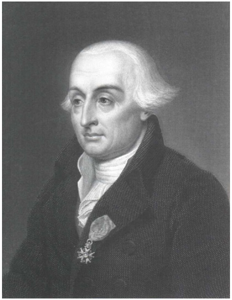
- fuse incompatible information - *Poisson Image Editing*, Perez (2003);
- interpolant f_2 of f_1 over Γ is the solution of

$$\text{(Euler - Lagrange) } \Delta f_2 = 0, \text{ on } \Gamma, \text{ with } f_2 = f_1, \text{ on } \partial\Gamma \implies \text{blurred};$$

- *guidance vector field* **p**:

$$\text{(Euler - Lagrange) } \Delta f_2 = \operatorname{div} \mathbf{p}, \text{ on } \Gamma, \text{ with } f_2 = f_1, \text{ on } \partial\Gamma \implies p = \nabla f_2;$$

Euler



Lagrange

Pérez (2003)

Osmosis: mean on $\partial\Gamma$

Conclusions and Future Works

- Fourier is the fastest way tested despite of a visually negligible Gibbs phenomenon;
- FA-BiCGStab or Exponential Integrators are alternative approaches;
- connection between *NL-Poisson inpainting* and *shadow removal* problems;
- better control on stopping criterion for BiCGStab;
- to model non-constant shadow areas: variational model to *inpaint* the light?
- simple old equations are still useful to model new computer vision problems;

Thank you for your attention.