PDE Based Methods for Finding Global Minima

Selim Esedoglu

Based on joint works with:

Tony Chan, Mila Nikolova;

Xavier Bresson, Stan Osher, Pierre Vandergeynst, and J. P. Thiran.
Based mainly on the works:


1. Rudin-Osher-Fatemi with $L^1$ Fidelity.
2. Contrast invariance.
3. Exact solutions.
4. Application to Image Decomposition.
5. Application to Shape Denoising.
6. Application to Image Segmentation.
7. Convex duality based numerical optimization.
Standard Total Variation Model

- Rudin, Osher, and Fatemi (1992):
  \[ E_2(u) = \int |\nabla u| + \lambda (f - u)^2 \, dx \]

- Preserves sharp edges.

- Many advantages over Perona-Malik:
  - Continuous dependence on data (i.e. \( f \)), and parameters (i.e. \( \lambda \)).
  - Only one parameter to be chosen by user: \( \lambda \)

- Still some difficulties:
  - Non-differentiable.
  - Naïve numerical methods slow to converge.

- Intimately related to: Piecewise constant Mumford-Shah.
Total Variation: Basic Facts

- To define total variation for possibly discontinuous functions:
- Given a vector $x \in \mathbb{R}^n$, we can write:
  $$ |x| = \max_{|y| \leq 1} x \cdot y $$
- Apply this to
  $$ \int |\nabla u| dx = \int \max_{|g| \leq 1} g \cdot \nabla u \; dx $$
- When $u(x)$ is smooth, can take $g(x)$ to be smooth and compactly supported, and move “max” outside:
  $$ \int |\nabla u| dx = \sup_{|g(x)| \leq 1} \int g \cdot \nabla u \; dx $$
Total Variation: Basic Facts

- Integrate by parts:
  \[ \int |\nabla u| \, dx = \sup_{|g(x)| \leq 1} \int u \nabla \cdot g \, dx \]

- Right hand side can be finite even for discontinuous \( u \).

- It is taken to be the definition of total variation:
  \[ \int |\nabla u| = \sup_{|g(x)| \leq 1} \int u \nabla \cdot g \, dx \]
Total Variation: Basic Facts

- Example:
  \[ u(x) = 1_{\Sigma}(x) \]
  where \( \Sigma \) is a compact set with smooth boundary \( \partial \Sigma \).

- First of all,
  \[
  \int u \nabla \cdot g \, dx = \int_{\Sigma} \nabla \cdot g \, dx = \int_{\partial \Sigma} g \cdot n \, d\sigma \leq \int_{\partial \Sigma} d\sigma = \text{Length}(\partial \Sigma)
  \]

- Second: There exists a vector field \( \psi(x) \) s.t.
  1. \( \psi \) is smooth, compactly supported,
  2. \( |\psi(x)| \leq 1 \) for all \( x \),
  3. \( \psi(x) = n(x) \) for all \( x \in \partial \Sigma \).

- We have:
  \[
  \int u \nabla \cdot g \, dx = \int_{\partial \Sigma} n \cdot n \, d\sigma = \text{Length}(\partial \Sigma)
  \]
Total Variation: Basic Facts

- Hence, we see that

\[ \int |\nabla 1_\Sigma(x)| = \text{Length}(\partial \Sigma) := \text{Per}(\Sigma) \]

when \( \partial \Sigma \) is smooth.

- If \( u(x) \) is piecewise constant:

\[ u(x) = \sum_{j=1}^{N-1} (c_j - c_{j-1}) 1_{\Sigma_j}(x) \]

with \( c_{j+1} > c_j > 0, \Sigma_j \subset \Sigma_{j+1} \), and \( \partial \Sigma_j \) smooth for all \( j \), then:

\[ \int |\nabla u| = \sum_{j=1}^{N-1} (c_{j+1} - c_j) \text{Per}(\Sigma_j) = \sum_{j=1}^{N-1} (c_{j+1} - c_j) \text{Per} \{x : u > c_j\} \]
Total Variation: Basic Facts

- Given any function $u \in L^1$, approximate by such piecewise constant functions.
- Use our formula.
- Pass to the limit.
- You get the co-area formula

$$\int |\nabla u| = \int \text{Per}(\{x : u(x) > \mu\}) \, d\mu$$
Standard ROF Model: Caveats

- **Contrast loss:** Consider the given image

\[
f(x) = 1_{BR(0)}(x)
\]

- **Exact solution:**

\[
u = c(\lambda, R)1_{BR}(x)
\]

where

\[
c(\lambda, R) = \max\left\{1 - \frac{1}{\lambda R}, 0\right\}
\]
Standard ROF Model: Caveats

- The image
  \[ f(x) = 1_{B_R(0)}(x) \]
  is noise free.
- Boundary (edges) \( \partial B_R(0) \) reconstructed exactly.
- However, there is contrast loss.
- **QUESTION:** What images are exactly preserved by the ROF model? i.e. which images are treated as noise free?

**ANSWER** (Y. Meyer): The only image that the ROF model reconstructs exactly is

\[ f(x) \equiv 0. \]
Standard ROF Model: Caveats

- **Relaxed QUESTION**: For which images of the form
  \[ f(x) = 1_\Omega(x) \]
  does the ROF model reconstruct the *shape boundary* exactly:
  \[ u(x) = c(\lambda)1_\Omega(x) \]?

- **Answer** (Bellettine, Caselles, Novaga): ROF preserves the boundary of only very special shapes: \( \Omega \) should be *bounded* and *convex*, with \( C^{1,1} \) boundary, and satisfy an inequality of the form:

\[
\sup_{x\in\partial\Omega} \kappa(x) \leq \frac{\text{Per}(\Omega)}{|\Omega|}
\]
ROF Model with $L^1$ Fidelity

$$E(u) = \int |\nabla u| + \lambda |f - u| \ dx$$

- Tiny perturbation of the original model.
- Useful advantages:
  - Contrast invariance.
  - Leaves many clean images invariant.
  - Leads to algorithms for finding global minima of
    - Segmentation, and
    - Denoising models.
- $E_1$ is convex, but not strictly so.
ROF Model with $L^1$ Fidelity

**Theorem** Let $D = \mathbb{R}^2$ and $f(x) = 1_\Omega(x)$, where $\Omega$ is a bounded domain with $C^2$ boundary $\partial \Omega$. There exists $\lambda_*$ such that if $\lambda > \lambda_*$, then the unique minimizer of $E_1(\cdot, \lambda)$ is given by $u(x) \equiv f(x)$.

**Proof:** Let $\phi(x)$ be a vector field such that:

1. $\phi(x) \in C^1_c(\mathbb{R}^n)$,
2. $|\phi(x)| \leq 1$ for all $x$,
3. $\phi(x) = n(x)$ for all $x \in \partial \Omega$.

(i.e. extend the normal off of $\partial \Omega$).
ROF Model with $L^1$ Fidelity

Then, we have:

$$\int |\nabla f| = \int f \text{div} \phi \ dx$$

and so:

$$E_1(u, \lambda) = \int |\nabla u| + \lambda \int |u - f| \ dx$$

$$\geq \int u \text{div} \phi \ dx + \lambda \int |u - f| \ dx$$

$$= \int f \text{div} \phi \ dx + \lambda \int |u - f| \ dx + \int (u - f) \text{div} \phi \ dx$$

$$\geq E_1(f, \lambda) + \left( \lambda - \max_{x \in \mathbb{R}^N} |\text{div} \phi(x)| \right) \int |u - f| \ dx.$$
ROF Model with $L^1$ Fidelity

- **UPSHOT:**
  - There is no contrast loss.
  - Many clean images are preserved exactly (treated as noise free).
Shape Denoising

Problem Statement: Given a binary observed image $f(x)$, to find a denoised (regularized) version. A binary image can be represented as:

$$f(x) = 1_\Omega(x)$$

where $\Omega$ is some domain in $D$, with possibly very rough boundary.

Applications: There are many (Osher):

- Denoising of fax documents ($D \subset \mathbb{R}^2$).
- Fairing of surfaces in computer graphics ($D \subset \mathbb{R}^3$).
- Understanding important segmentation models, which are closely related: Mumford-Shah and its variants.
Take \( f(x) = 1_{\Omega}(x) \) and restrict minimization to set of binary images:

\[
\min_{\begin{subarray}{l} \Sigma \subset D \\ u(x) = 1_{\Sigma}(x) \end{subarray}} \int_D |\nabla u| + \lambda \int_D \left( u(x) - 1_{\Omega}(x) \right)^2 \, dx
\]

It is of course equivalent to the following geometry problem: Given a set \( \Omega \subset D \),

\[
\min_{\Sigma \subset D} \text{Per}(\Sigma) + \lambda |\Sigma \Delta \Omega|
\]

where we use the following notation:

- \( S_1 \Delta S_2 \) denotes the symmetric difference of the sets \( S_1 \) and \( S_2 \):
  \[
  S_1 \Delta S_2 = (S_1 \setminus S_2) \cup (S_2 \setminus S_1).
  \]

- \( |\cdot| \) denotes area in 2D, volume in 3D.

Considered using level sets: Osher & Kang, Osher & Vese, etc.
Geometry Problem

\[ \min_{\Sigma \subset D} \text{Per}(\Sigma) + \lambda |\Sigma \Delta \Omega| \]

- This is a non-convex optimization problem, since the set of binary functions is non-convex.

- The global minimizer is not unique in general.

- In general, in addition to more than one global minimizer, there are also local minimizers.
Local Minima

Terzopoulos snakes, standard level sets (with reinitialization), etc.

Other types of local minima:

It is easy to verify rigorously that these scenarios do arise.
Solution via TVL$^1$:

**Claim:** To find a solution (i.e. a global minimizer) $u(x)$ of the non-convex variational problem

$$\min_{\Sigma \subset D} \text{Per}(\Sigma) + \lambda |\Sigma \Delta \Omega|$$

it is sufficient to carry out the following steps:

1. Find any minimizer of the **convex** energy

$$E_1(u) := \int_D |\nabla u| + \lambda \int_D |u(x) - 1_{\Omega}(x)| \, dx.$$  

Call the solution found $v(x)$.

2. Let $\Sigma = \{x \in D : v(x) > \mu\}$ for some $\mu \in (0, 1)$.

Then $\Sigma$ is a global minimizer of the original non-convex problem for almost every choice of $\mu$. 
Solution via TVL$^1$

**Proof** is based on the following formula for the energy $E_1$:

$$E_1(u) = \int_{-\infty}^{\infty} \text{Per} \left( \{ x : u(x) > \mu \} \right)$$

$$+ \lambda \left| \{ x : u(x) > \mu \} \triangle \{ x : f(x) > \mu \} \right| d\mu$$

This entails two ingredients:

- Coarea formula:
  $$\int_{\mathbb{R}^N} |\nabla u| = \int_{\mathbb{R}} \text{Per}(\{ x : u(x) > \mu \}) d\mu.$$

- “Layer Cake” theorem:
  $$\int_{\mathbb{R}^N} \left| u(x) - f(x) \right| dx = \int_{\mathbb{R}} \left| \{ x : u(x) > \mu \} \triangle \{ x : f(x) > \mu \} \right| d\mu.$$
Theorem: Let the observed image \( f(x) \) be the characteristic function of a bounded domain \( \Omega \subset \mathbb{R}^N \). Fix a \( \lambda \geq 0 \), and let \( u_\lambda(x) \in M(\lambda) \) be any minimizer of \( E_1(\cdot, \lambda) \). Then, for almost every \( \mu \in [0, 1] \) we have that the binary function

\[
1_{\{x: u_\lambda(x) > \mu\}}(x)
\]

is also a minimizer of \( E_1(\cdot, \lambda) \).

Proof: When \( f(x) = 1_\Omega(x) \), formula of the previous proposition becomes:

\[
E_1(u, \lambda) = \int_0^1 \text{Per} \left( \{x : u(x) > \mu\} \right) + \lambda \left| \{x : u(x) > \mu\} \Delta \Omega \right| d\mu
\]

Setting \( \Sigma(\mu) := \{x : u(x) > \mu\} \), this suggests that we consider for each level set of \( u(x) \) our geometry problem:

\[
\min_{\Sigma \subset \mathbb{R}^N} \text{Per}(\Sigma) + \lambda |\Sigma \Delta \Omega|
\]
Solution via TVL\(^1\)

Let \( \Sigma_* \subset \mathbb{R}^N \) be a minimizer of the geometry problem. Then,

\[
\text{Per}(\Sigma(\mu)) + \lambda |\Sigma(\mu) \triangle \Omega| \geq \text{Per}(\Sigma_*) + \lambda |\Sigma_* \triangle \Omega|
\]

for all \( \mu \). This implies:

\[
E_1(u_{\lambda}(x), \lambda) \geq E_1(1_{\Sigma_*}(x), \lambda)
\]

which means that \( 1_{\Sigma_*}(x) \) is also a minimizer of \( E(\cdot, \lambda) \). Conversely, since \( u_{\lambda}(x) \) is a minimizer, we must really have:

\[
E_1(u_{\lambda}(x), \lambda) = E_1(1_{\Sigma_*}(x), \lambda).
\]

That leads to the conclusion:

\[
\text{Per}(\Sigma(\mu)) + \lambda |\Sigma(\mu) \triangle \Omega| = \text{Per}(\Sigma_*) + \lambda |\Sigma_* \triangle \Omega| \text{ a.e. } \mu.
\]

\( \Rightarrow \Sigma(\mu) \) is a minimizer of the geometry problem for a.e. \( \mu \in (0, 1) \).

\( \Rightarrow 1_{\Sigma(\mu)}(x) \) is a minimizer of \( E_1(\cdot, \lambda) \).
Previous Work

This idea of writing total variation based optimization problems in terms of super-level sets goes back (at least) to the works of G. Strang for problems in plasticity:

\[
\min_{f} \int_{D} |\nabla u| \quad \text{subject to } \int_{D} u f dx = 1.
\]

It is shown that the minimizer is achieved at a characteristic function for the optimization problem above.


Example: Original binary image $f(x)$:
Minimizer of $E_1$ found:
Intermediates, showing the evolution:
Histograms of the intermediates:

**UPSHOT:** Need to go through **non-binary** images to avoid getting stuck in local minimizers.
Application: Multiscale Decomposition

As in: E. Tadmor, S. Nezzar, L. Vese.
Application: Multiscale Decomposition

- Much cleaner decompositions than what’s possible with standard ROF model.

- Many further results in this direction by Yin, Osher, Goldfarb, especially concerning cartoon – texture decomposition of images.
Generalization to P. C. Mumford-Shah

- Piecewise constant Mumford-Shah model of Chan & Vese:

\[
\min_{c_1, c_2 \in \mathbb{R}} \left\{ \operatorname{Per}(\Sigma) + \lambda \left( \int_{\Sigma} (c_1 - f)^2 \, dx + \int_{D \setminus \Sigma} (c_2 - f)^2 \, dx \right) \right\} = \int_D (u - f)^2 \, dx
\]

- Best approximation of the image \( f(x) \) among images that take only two values:

\[
u(x) = c_1 1_\Sigma(x) + c_2 1_{D \setminus \Sigma}(x)
\]
Generalization to P. C. Mumford-Shah

Our formulation:

\[
\min_{c_1, c_2 \in \mathbb{R}} \min_{0 \leq u(x) \leq 1} \int_D |\nabla u| + \lambda \int_D \left\{ (c_1 - f)^2 - (c_2 - f)^2 \right\} u(x) \, dx.
\]

Claim: If \((c_1, c_2, u(x))\) is a solution of above problem, then

\[
\left( c_1, c_2, 1_{\{x : u(x) > \mu\}}(x) \right)
\]

is a global minimizer of the P. C. Mumford-Shah model for a.e. \(\mu \in (0,1)\).

⇒ UPSHOT: For fixed \(c_1, c_2\), the inner minimization is convex.
⇒ Local minima due to geometry are eliminated.
Generalization to P. C. Mumford-Shah
Generalization to P. C. Mumford-Shah
Generalization to P. C. Mumford-Shah

Stationary state reached
Alternative Methods

- **Graph cuts:**
  - Anisotropy in the perimeter term.
  - Have to increase connectivity (# of edges per vertex) to converge to isotropic perimeter:
Alternative Methods

- On the other hand, Lucier et. al. show the following:
- Limit of the discrete energies:
  \[ \sum_{i,j} h^2 \sqrt{(D_x^+ u_{i,j})^2 + (D_y^+ u_{i,j})^2} \]
  as grid size $h \to 0$ is
  \[ \int |\nabla u| \]
  i.e. the isotropic perimeter.
- **UPSHOT:** With the PDE approach, no need to increase “connectivity” to converge to isotropic shape optimization problem, even though solutions are characteristic functions.
Geodesic Active Contours

- **Gradient descent** for:

\[
\int_0^1 \frac{1}{1 + |(\nabla G_\sigma * f)(y)|^2} |y'(s)| \, ds
\]

- Let

\[
g(x) = \frac{1}{1 + |(\nabla G_\sigma * f)(y)|^2}
\]

- **Cohen & Kimmel:**
  - Pick two points on the boundary: \( p \) & \( q \).
  - Construct the distance function \( d_p(x) \) to \( p \).
    \[
    \left| \nabla d_p \right| = g(x).
    \]
  - Gradient descent on \( d_p(x) \), starting from \( q \).
Geodesic Active Contours

- **Cohen & Kimmel**: Unfortunately, no generalization to higher dims.

- **Our formulation**:
  \[
  \min_{0 \leq u \leq 1} \int g(x)|\nabla u| + \lambda \int \{(f - c_2)^2 - (f - c_1)^2\} u \, dx
  \]
  or
  \[
  \min_{0 \leq u \leq 1} \int g(x)|\nabla u| \quad u(x) = 1 \ \forall x \in D
  \]

- Directly generalizes to any dimension.
Extensions to Multiphase Setting

- An extension due to Chambolle:

\[ \min_{u(x) \in \{c_1, \ldots, c_k\}} \int |\nabla u| + (f - u)^2 \, dx \]

- Segmentation into \( k \) regions.

- Boundaries weighted very unequally:
  - Let \( w_{i,j} \) = Weight of interface between \( \Sigma_i \) and \( \Sigma_j \).
  - Then:
    \[ w_{i,j} = w_{i,\ell} + w_{\ell,j} \]

where \( i > \ell > j \).
Fast TV Iteration

- Chambolle’s iteration:

\[
p^{n+1} = \frac{p^n + (\delta t) \left\{ \nabla (\nabla \cdot p^n) - \frac{1}{\lambda} \nabla f \right\}}{1 + (\delta t) \left| \nabla (\nabla \cdot p^n) - \frac{1}{\lambda} \nabla f \right|}.
\]

- Converges for \( \delta t \leq \frac{1}{3} \). In practice, can take \( \delta t \leq \frac{1}{4} \).
- No dependence on a regularization parameter \( \epsilon \).
- Complexity still \( O(N^2) \), where \( N \) = Number of pixels (2D).
Fast TV Iteration

- Observe: Chambolle’s iteration is equivalent to

\[ \frac{p^{n+1} - p^n}{\delta t} = \nabla (\nabla \cdot p^n) - \frac{1}{\lambda} \nabla f - \left| \nabla (\nabla \cdot p^n) - \frac{1}{\lambda} \nabla f \right| p^{n+1} \]

- Explicit time stepping, except for diagonal terms.
- This is the reason for \( O(N^2) \) scaling.
Fast TV Iteration

- **Our approach:** Implicit schemes for Chambolle’s Euler-Lagrange equation.
- First, consider:
  \[
  \frac{p^{n+1} - p^n}{\delta t} = \nabla (\nabla \cdot p^{n+1}) - \frac{1}{\lambda} \nabla f - \left| \nabla (\nabla \cdot p^n) - \frac{1}{\lambda} \nabla f \right| p^{n+1}.
  \]
- This is a semi-implicit scheme (implicit off-diagonal terms).
- Unconditionally stable (proof in progress).
Fast TV Iteration

Let's write out the components:

\[
\frac{p_1^{n+1} - p_1^n}{\delta t} = D_{xx} p_1^{n+1} + D_{xy} p_2^{n+1} - \frac{1}{\lambda} f_x - |\nabla (\nabla \cdot p^n)| - \frac{1}{\lambda} \nabla f |p_1^{n+1}|
\]

\[
\frac{p_2^{n+1} - p_2^n}{\delta t} = D_{yy} p_2^{n+1} + D_{xy} p_1^{n+1} - \frac{1}{\lambda} f_y - |\nabla (\nabla \cdot p^n)| - \frac{1}{\lambda} \nabla f |p_2^{n+1}|
\]

Observe:

- Only \( x \) derivatives on \( p_1 \) in the first equation, and
- Only \( y \) derivatives on \( p_2 \) in the second equation.
Fast TV Iteration

- Consider: A slightly less implicit scheme:

\[
\frac{p_1^{n+1} - p_1^n}{\delta t} = D_{xx} p_1^{n+1} + D_{xy} p_2^n - \frac{1}{\lambda} f_x - |\nabla (\nabla \cdot p^n) - \frac{1}{\lambda} \nabla f| p_1^{n+1},
\]

\[
\frac{p_2^{n+1} - p_2^n}{\delta t} = D_{yy} p_2^{n+1} + D_{xy} p_1^n - \frac{1}{\lambda} f_y - |\nabla (\nabla \cdot p^n) - \frac{1}{\lambda} \nabla f| p_2^{n+1}.
\]

- A great discretization: Solve independent tridiagonal systems.
- Unfortunately: Not unconditionally stable.
Fast TV Iteration

- **Stabilization:**

\[
\frac{p_1^{n+1} - p_1^n}{\delta t} = D_{xx} p_1^{n+1} + D_{xy} p_2^n - \frac{1}{\lambda} f_x - |\nabla(\nabla \cdot p^n)| - \frac{1}{\lambda} \nabla f |p_1^{n+1} + D_{yy} p_1^{n+1} - D_{yy} p_1^n.
\]

\[
\frac{p_2^{n+1} - p_2^n}{\delta t} = D_{yy} p_2^{n+1} + D_{xy} p_1^n - \frac{1}{\lambda} f_y - |\nabla(\nabla \cdot p^n)| - \frac{1}{\lambda} \nabla f |p_2^{n+1} + D_{xx} p_2^{n+1} - D_{xx} p_2^n.
\]

- **Empirical evidence:** Unconditionally stable (proof in progress).
Fast TV Iteration

- At every iteration, need to solve

\[ \Delta p_1^{n+1} - a(x) p_1^{n+1} = RHS_1, \]
\[ \Delta p_2^{n+1} - b(x) p_2^{n+1} = RHS_2. \]

- Independent Poisson equations for \( p_1 \) and \( p_2 \).
- We solve them by
  - Alternating Direction Implicit (ADI) method,
  - Chebychev acceleration.
Fast TV Iteration

- Slightly faster version:

\[
\frac{p_1^{n+1} - p_1^n}{\delta t} = D_{xx} p_1^{n+1} + D_{xy} p_2^n - \frac{1}{\lambda} f_x - |\nabla (\nabla \cdot p^n)| - \frac{1}{\lambda} \nabla f |p_1^{n+1} + \theta D_{yy} p_1^{n+1} - \theta D_{yy} p_1^n,
\]

\[
\frac{p_2^{n+1} - p_2^n}{\delta t} = D_{yy} p_2^{n+1} + D_{xy} p_1^n - \frac{1}{\lambda} f_y - |\nabla (\nabla \cdot p^n)| - \frac{1}{\lambda} \nabla f |p_2^{n+1} + \theta D_{xx} p_2^{n+1} - \theta D_{xx} p_2^n.
\]

- Empirical observation: Unconditionally stable for \( \theta \geq \frac{1}{4} \).

- Every iteration involves:

\[
D_{xx} p_1^{n+1} + \theta D_{yy} p_1^{n+1} - a(x) p_1^{n+1} = RHS_1,
\]

\[
D_{yy} p_2^{n+1} + \theta D_{xx} p_2^{n+1} - b(x) p_2^{n+1} = RHS_2.
\]
Fast TV Iteration

Test images:

- Synthetic Image
- Cameraman Image
Fast TV Iteration

Performance on the **synthetic** image:

<table>
<thead>
<tr>
<th>Method</th>
<th>$128^2$, $\lambda = 0.5$</th>
<th>$256^2$, $\lambda = 0.5$</th>
<th>$512^2$, $\lambda = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chambolle</td>
<td>1700; CPU=1</td>
<td>6500; CPU=17</td>
<td>23K; CPU=302</td>
</tr>
<tr>
<td>New</td>
<td>8; CPU=0.11</td>
<td>8; CPU=0.5</td>
<td>8; CPU=2.13</td>
</tr>
</tbody>
</table>

Performance on the **synthetic** image, bigger $\lambda$:

<table>
<thead>
<tr>
<th>Method</th>
<th>$128^2$, $\lambda = 1$</th>
<th>$256^2$, $\lambda = 1$</th>
<th>$512^2$, $\lambda = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chambolle</td>
<td>3K; CPU=1.83</td>
<td>12K; CPU=32.8</td>
<td>37K; CPU=458</td>
</tr>
<tr>
<td>New</td>
<td>15; CPU=0.22</td>
<td>16; CPU=0.97</td>
<td>15; CPU=4</td>
</tr>
</tbody>
</table>
Fast TV Iteration

- Performance on the *cameraman* image:

<table>
<thead>
<tr>
<th>Method</th>
<th>$256^2$, $\lambda = 0.5$</th>
<th>$256^2$, $\lambda = 1$</th>
<th>$256^2$, $\lambda = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chambolle</td>
<td>5800; CPU=14</td>
<td>17000; CPU=40.7</td>
<td>40000; CPU=96</td>
</tr>
<tr>
<td>New</td>
<td>19; CPU=0.74</td>
<td>35; CPU=1.41</td>
<td>70; CPU=2.73</td>
</tr>
</tbody>
</table>
Fast TV Iteration

Figure: Original cameraman image, and its processed versions at $\lambda = 0.5$, $\lambda = 1$, and $\lambda = 2$. 
Fast TV Iteration

Comparisons with Yin & Zhang's method: Synthetic image at $256^2$ resolution, with $\lambda = 0.5$

<table>
<thead>
<tr>
<th>Residual threshold</th>
<th>Iterations</th>
<th>Error</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>71</td>
<td>11%</td>
<td>2.35 sec</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>379</td>
<td>5%</td>
<td>11.6 sec</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>4269</td>
<td>1.7%</td>
<td>129 sec</td>
</tr>
<tr>
<td>$3 \times 10^{-6}$</td>
<td>10425</td>
<td>1%</td>
<td>316 sec</td>
</tr>
</tbody>
</table>

New algorithm takes about 0.5 sec.