Total Variation Methods: Overview and Applications

Thomas Pock
Introduction

• Computer Vision inherently deals with inverse problems:
  – One has to determine unknown quantities based on the observed data
  – Example: Denoising

**Observed data:** Noisy Image

**Unknown Quantity:** Clean Image
• Inverse problems in Computer Vision are mostly ill-posed
• Regularization of the solution is needed.
• But which regularization (prior) is the best?

• The Bayesian framework is often used to estimate the unknown quantities
• Bayesian inference is known to deliver solutions of minimal error

\[
\theta_{MAP} = \max_{\theta} \left\{ P(\theta \mid x) = \frac{p(x \mid \theta)P(\theta)}{p(x)} \right\}
\]
Example: Tikhonov Denoising [1]

\[
\max \left\{ P(u \mid u_0) = \prod_{\Omega} P(u) p(u_0 \mid u) \right\}
\]

\[
P(u) = \exp \left( -\frac{|\nabla u|^2}{2} \right), \quad p(u_0 \mid u) = \exp \left( -\frac{\lambda}{2} (u - u_0)^2 \right)
\]

\[
\min \left\{ E = \frac{1}{2} \int_{\Omega} |\nabla u|^2 \, dx + \frac{\lambda}{2} \int_{\Omega} (u - u_0)^2 \, dx \right\}
\]

\[
- \Delta u + \lambda (u - u_0) = 0
\]

Example: Tikhonov Denoising

- Works fine but does not preserve edges
- Quadratic image prior does not allow for sharp edges

\[
\min \left\{ \frac{1}{2} \int_{\Omega} |\nabla u|^2 \, dx + \frac{\lambda}{2} \int_{\Omega} (u - u_0)^2 \, dx \right\}
\]

\[
\int_{\Omega} |\nabla u| \, dx \quad \text{...Total Variation}
\]
Outline

• (I) History
• (II) Properties
• (III) Numerical Methods
• (IV) Applications
History

• Introduced by Rudin, Osher and Fatemi in 1992

Nonlinear total variation based noise removal algorithms*

Leonid I. Rudin¹, Stanley Osher and Emad Fatemi²
Cognitek Inc., 2800, 28th Street, Suite 101, Santa Monica, CA 90405, USA

A constrained optimization type of numerical algorithm for removing noise from images is presented. The total variation of the image is minimized subject to constraints involving the statistics of the noise. The constraints are imposed using Lagrange multipliers. The solution is obtained using the gradient-projection method. This amounts to solving a time dependent partial differential equation on a manifold determined by the constraints. As \( t \to \infty \), the solution converges to a steady state which is the denoised image. The numerical algorithm is simple and relatively fast. The results appear to be state-of-the-art for very noisy images. The method is noninvasive, yielding sharp edges in the image. The technique could be interpreted as a first step of moving each level set of the image normal to itself with velocity equal to the curvature of the level set divided by the magnitude of the gradient of the image, and a second step which projects the image back onto the constraint set.
History

- Original formulation: Constraint optimization problem

\[
\inf_{u} \left\{ \int_{\Omega} |\nabla u| \, dx \right\} \quad \text{s.t.} \quad \int_{\Omega} (u - u_0)^2 \, dx = \sigma^2 \quad (\text{non-convex})
\]

- Equivalent to the unconstraint optimization problem [2]

\[
\min_{u} \left\{ \int_{\Omega} |\nabla u| \, dx + \frac{\lambda}{2} \int_{\Omega} (u - u_0)^2 \, dx \right\} \quad (\text{convex})
\]

History

• Extensions
  – Color [3]
  – Deblurring [4]
  – Inpainting [5]
  – Image decomposition [6]
  – Active contours [7]

• Numerical schemes
  – Newton methods [8]
  – Duality-based methods [9]

Basic Properties

• The TV norm is the $L^1$ norm of the magnitude of the image gradient

$$TV(u) = \int_{\Omega} |\nabla u| \, dx = \int_{\Omega} \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2} \, dx$$

• Advantages:
  – disfavours oscillations such as noise
  – allowing for sharp discontinuities

• Disadvantages:
  – highly nonlinear
  – Hard to optimize
Synthetic Example

- **TV-norm vs. TK-norm**

  \[
  \begin{align*}
  TV_1 &= 135.4 \\
  TK_1 &= 136 \\
  TV_2 &= 4408.9 \\
  TK_2 &= 2585.6
  \end{align*}
  \]

  Look at ratios:
  \[
  \begin{align*}
  TV_2 / TV_1 &= 32.6 \\
  TK_2 / TK_1 &= 19.0
  \end{align*}
  \]

- **Example: TV-Denoising**
  - Large image gradients are treated as outliers
  - Are less weighted in estimation the true pixel value
Theoretical Properties

- Minimizers of the ROF model live in the BV-space

\[ BV : \left\{ u \in L^1(\Omega) \ \middle| \ \int_{\Omega} |\nabla u| < \infty \right\} \]

- Used to prove existence and uniqueness of a solution

- Dual Formulation of the TV-norm

\[ \int_{\Omega} |\nabla u| dx = \sup \left\{ \int_{\Omega} u \nabla \cdot \vec{g} \ dx \ \middle| \ \| \vec{g} \| \leq 1 \right\} \]
• Co-area formula

\[ \int_{\Omega} |\nabla u| = \int_{-\infty}^{\infty} \left\{ \int_{\{u=\gamma\}} ds \right\} d\gamma \]

• The TV-norm can be decomposed by means of its level sets
• The TV-norm controls both the size of the jumps and the geometry
Numerical Methods

- Euler-Lagrange equations of the ROF model

\[
E = \int_\Omega |\nabla u| \, d\Omega + \frac{\lambda}{2} \int_\Omega (u - u_0)^2 \, d\Omega
\]

\[
\frac{\partial E}{\partial u} = -\nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) + \lambda (u - u_0) = 0 \quad \text{(Primal Formulation)}
\]

- Problem: EL equation is degenerated in flat regions

Replace \( |\nabla u| \) by \( |\nabla u|_\varepsilon = \sqrt{|\nabla u|^2 + \varepsilon^2} \)

- But decreases ability to preserve sharp edges
Primal-based Methods

- Artificial time marching
  \[
  \frac{\partial u}{\partial t} \equiv \frac{\partial E}{\partial u}, \quad u^{n+1} = u^n - \Delta t \frac{\partial u}{\partial t}
  \]
  - Slow convergence since \( \Delta t \leq c\Delta x^2|\nabla u| \) (CFL condition)

- Fixed point iteration scheme (quasi Newton method) [10]
  - No CFL condition

\[
\nabla \cdot \left( \frac{\nabla u^{i+1}}{|\nabla u^i|_\varepsilon} \right) - \lambda \left( u^{i+1} - f \right) = 0
\]

Duality-based Methods

• Dual Formulation of the TV norm

\[
\int_{\Omega} |\nabla u| dx = \sup \left\{ \int_{\Omega} u \nabla \cdot \tilde{g} \ dx \mid \|\tilde{g}\| \leq 1 \right\}
\]

• Plugging into the ROF model, one arrives at

\[
\sup_{\|\tilde{g}\| \leq 1} \left\{ \int_{\Omega} u \nabla \cdot \tilde{g} dx - \frac{1}{2\lambda} \int_{\Omega} (\nabla \cdot \tilde{g})^2 dx \right\}, \quad u = u_0 - \frac{1}{\lambda} \nabla \cdot \tilde{g}
\]

  – Continuously differentiable in g but has many constraints

• Primal-Dual Newton Method of Chan, Golub and Mulet [8]
• **Chambolle’s algorithm [9]**

\[
E = \int_{\Omega} |\nabla u| dx + \frac{1}{2\lambda} \int_{\Omega} (u - u_0)^2 dx
\]

\[
\frac{\partial E}{\partial u} = -\nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) + \frac{1}{\lambda} (u - u_0) = 0 \quad \text{(Primal Euler - Lagrange equation)}
\]

\[
\tilde{p} = \left( \frac{\nabla u}{|\nabla u|} \right) \quad \text{(dual variable)}
\]

(I) \( \tilde{p}|\nabla u| - \nabla u = 0, \quad \|\tilde{p}\| \leq 1 \)

(II) \(-\nabla \cdot \tilde{p} + \frac{1}{\lambda} (u - u_0) = 0 \Rightarrow u = \lambda \nabla \cdot \tilde{p} + u_0 \)

\[
\tilde{p}|\nabla (\lambda \nabla \cdot \tilde{p} + u_0)| - \nabla (\lambda \nabla \cdot \tilde{p} + u_0) = 0
\]

\[
\bar{p}^{n+1} = \bar{p}^n + \tau \nabla \left( \frac{\nabla \cdot \bar{p}^n + u_0}{\lambda} \right) \frac{1}{1 + \tau |\nabla \left( \frac{\nabla \cdot \bar{p}^n + u_0}{\lambda} \right)|}
\]

**Fixed point iteration scheme**
Applications

• Denoising using the ROF model

Computational time: 30ms on Nvidia 8800 GTX / CUDA for 1000x1000 images
• Denoising of color images using the ROF model
• TV inpainting
Optical Flow

- Optical Flow is a major task of biological and artificial visual systems
- Relates the motion of pixels between different image frames
- Optical Flow Constraint:
  \[
  \begin{pmatrix}
  u_1(x) \\
  u_2(x)
  \end{pmatrix} \cdot \nabla I(x,t) + I_t(x,t) = 0
  \]
- Gives only the normal flow
- No OFC in untextured areas
TV-L¹ Optical Flow

\[ E_{TV-L^1} = \int_{\Omega} |\nabla u_1| + |\nabla u_2| \, dx + \lambda \int_{\Omega} |I_1(\vec{x} + \vec{u}(\vec{x})) - I_0(\vec{x})| \, dx \]

- **Advantages:**
  - Allows for discontinuities in the flow and outliers in the data term

- **Disadvantages:**
  - Sophisticated optimization techniques are needed
2D Results
3D Results

• We have also applied the same algorithm for non-rigid medical image registration
• For $256^3$ volumes the algorithm takes about 15 min. on a single CPU
Fusion of Depth Images
TV-L¹ Depth Image Fusion

\[ E_{TV-L^1} = \int_{\Omega} |\nabla u| \, dx + \lambda \int_{\Omega} \sum_{i \in I(x)} w_i(x)|u - f_i| \, dx \]

- The TV term penalizes the perimeter of the level sets of \( u \) which is (in 3D) exactly the surface area of the 3D model.
- The data term measures the distance of \( u \) to all \( f_i \) by means of the robust \( L^1 \) norm.
Results
Results
Links between TV regularization, minimal surfaces and shape optimization problems

- Geodesic active contours introduced by Casselles et al. 1997
- Closed curves (surfaces) which evolve to minimize the weighted length (area)

\[
E_{gac}[S] = \int_S g(S) ds
\]

\[
g = \frac{1}{1 + \alpha |\nabla G_\sigma * I|^p} + \varepsilon
\]

\[
\frac{\partial S}{\partial \tau} = -(g\kappa - \nabla g \cdot \vec{N})\vec{N}
\]

- Can be implemented e.g. using Level Sets
- Problem: Gradient descent usually converge to local minima
A TV-based Formulation

- Geodesic Active Contours can also be formulated as a weighted TV – inpainting functional

\[ E_{TV-gac} = \int_D g|\nabla u|dx + \int_{\Omega} \lambda_D(x)(u - f)^2 dx \]

\( u, f \in [0,1] \)

\[ \lambda_D(x) = \begin{cases} 
0 & \text{in } D \\
\infty & \text{in } \Omega \setminus D 
\end{cases} \]

- Functional is convex
- Computes the globally optimal solution
- Can also be computed using Chambolle’s algorithm
- The final contour can be extracted by simple thresholding \( u \)
Results
Results
Fast Implementation Using Graphics Hardware

- Performance of graphics cards is steadily increasing
- Particularly well suited to compute variational models
Some Features

- 32 bit floating point arithmetic, 768 MB main memory
- Rich set of general purpose arithmetic operations and conditional branches
- All features can be accessed via C-like languages (CUDA)
- Flagship of Nvidia: Nvidia 8800 GTX

(Supercomputer in Pocket Format)
Thank You!