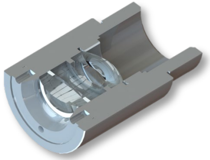
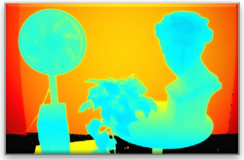








GAMES 204

Computational Imaging

Lecture 14: Image Deconvolution with the Half Quadratic Splitting (HQS) Method

Qilin Sun（孙启霖）

香港中文大学（深圳）

点昀技术（Point Spread Technology）

Today's Topic

- Image Deconvolution – Brief Review
- A Bayesian Perspective of Inverse Problems
- Image Priors/Regularization and Total Variation (TV)
- Half-Quadratic Splitting (HQS) Method
- Image Deconvolution with HQS
- Outlook on Unrolled Optimization

Image Deconvolution

Image Deconvolution – Brief Review

Given: blurry & noisy image

Desired: sharp & noise-free image

Image Deconvolution – Brief Review

- Image formation model:

$$b = c * x + \eta$$

2D measurements

known 2D convolution kernel

2D target image

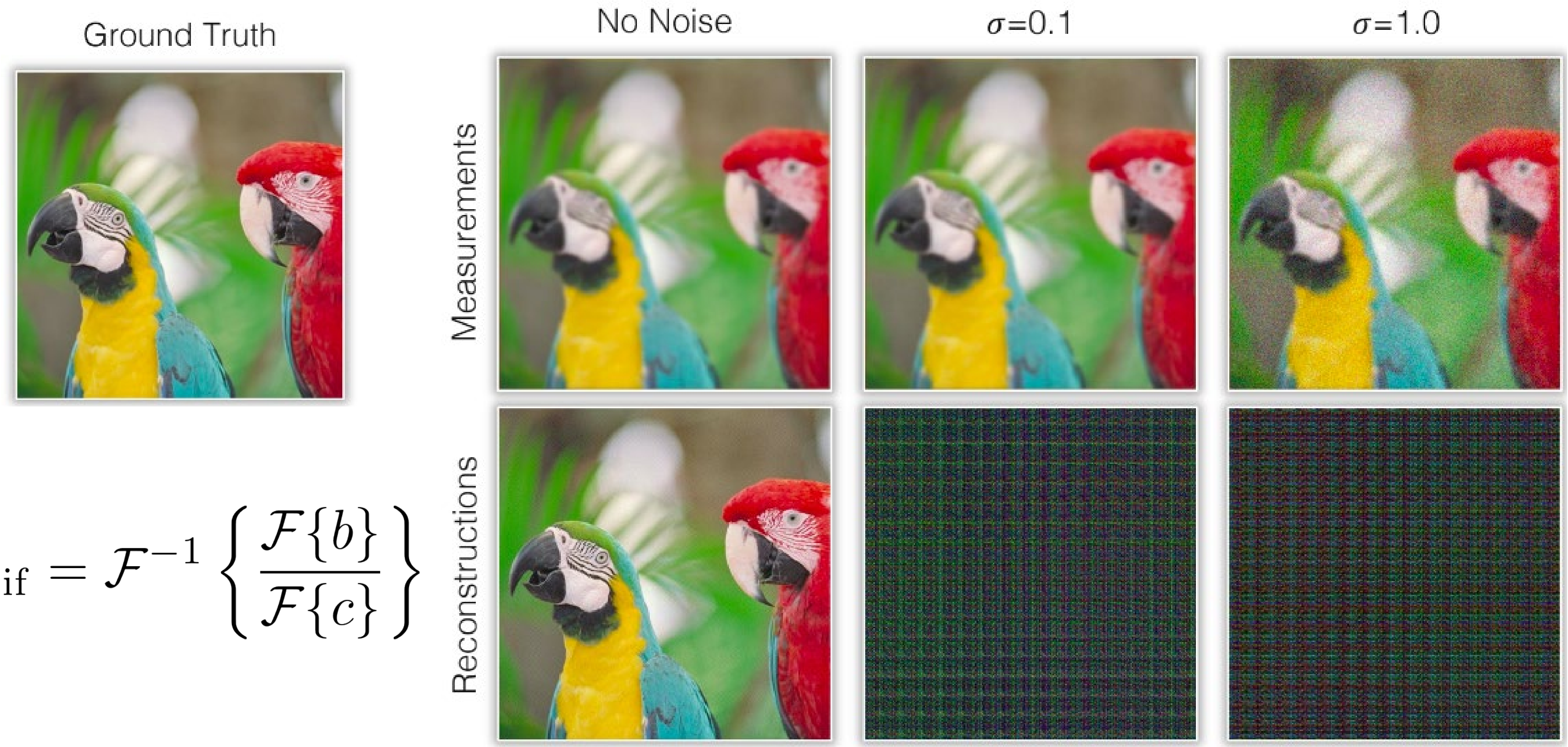
additive noise

Image Deconvolution – Brief Review

- Image formation model: $$b = c * x + \eta$$
- Convolution theorem: $$b = \mathcal{F}^{-1}\{\mathcal{F}\{c\} \cdot \mathcal{F}\{x\}\} + \eta$$
- Inverse filtering: $$\tilde{x}_{\text{if}} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}}\right\}$$
- Wiener filtering: $$\tilde{x}_{\text{wf}} = \mathcal{F}^{-1}\left\{\frac{|\mathcal{F}\{c\}|^2}{|\mathcal{F}\{c\}|^2 + 1/SNR} \cdot \frac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}}\right\}$$
- Duality of "signal processing" and "algebraic" interpretation:

$$b = c * x \quad \Longleftrightarrow \quad \boldsymbol{b} = \boldsymbol{C}\boldsymbol{x} \qquad \boldsymbol{C} \in \mathbb{R}^{N \times N}, \quad \boldsymbol{b}, \boldsymbol{x} \in \mathbb{R}^N$$

Image Deconvolution – Inverse Filtering



$$\tilde{x}_{\text{if}} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}}\right\}$$

Image Deconvolution – Wiener Filtering

Ground Truth

$$\tilde{x}_{\mathrm{wf}} = \mathcal{F}^{-1}\left\{ \frac{|\mathcal{F}\{c\}|^2}{|\mathcal{F}\{c\}|^2 + 1/SNR} \cdot \frac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}} \right\}$$

Image Deconvolution

- Problem: this is an **ill-posed** inverse problem, i.e., there are infinitely many solutions that satisfy the measurements

- Need some way to determine how “desirable” any one of these feasible solutions is -> need an image prior

A Bayesian Perspective of Inverse Problems

A Bayesian Perspective of Inverse Problems

- Image formation model: $$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta}, \quad \boldsymbol{b} \in \mathbb{R}^{M}, \boldsymbol{x} \in \mathbb{R}^{N}, \boldsymbol{A} \in \mathbb{R}^{M \times N}$$
- Interpret as random variables: $$\boldsymbol{x}_i \sim \mathcal{N}\left(\boldsymbol{x}_i, 0\right), \quad \boldsymbol{\eta}_i \sim \mathcal{N}\left(0, \sigma^2\right)$$ $$\boldsymbol{b}_i \sim \mathcal{N}\left((\boldsymbol{A}\boldsymbol{x})_i, \sigma^2\right)$$
- Probability of observation ***i***: $$p\left(\boldsymbol{b}_i \mid \boldsymbol{x}_i, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(\boldsymbol{b}_i - (\boldsymbol{A}\boldsymbol{x})_i\right)^2}{2\sigma^2}}$$
- Joint probability of all observations: $$p(\boldsymbol{b} \mid \boldsymbol{x}, \sigma) = \prod_{i=1}^{M} p\left(\boldsymbol{b}_i \mid \boldsymbol{x}_i, \sigma\right) \propto e^{-\frac{\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2}{2\sigma^2}}$$

A Bayesian Perspective of Inverse Problems

- Bayes' rule:

$$\underbrace{p(\boldsymbol{x}|\boldsymbol{b},\sigma)}_{\text{posterior}} = \frac{p(\boldsymbol{b}|\boldsymbol{x},\sigma)\boldsymbol{p}(\boldsymbol{x})}{p(\boldsymbol{b})} \propto \underbrace{p(\boldsymbol{b}|\boldsymbol{x},\sigma)}_{\text{image formation model}}\underbrace{p(\boldsymbol{x})}_{\text{prior}}$$

- Maximum-a-posterior (MAP) solution:

$$\begin{aligned}\boldsymbol{x}_{MAP} &= \arg\min_x -\log(p(\boldsymbol{x} \mid \boldsymbol{b},\sigma)) \\ &= \arg\min_x -\log(p(\boldsymbol{b} \mid \boldsymbol{x},\sigma)) - \log(p(\boldsymbol{x})) \\ &= \arg\min_x \frac{1}{2\sigma^2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \Psi(\boldsymbol{x})\end{aligned}$$

A Bayesian Perspective of Inverse Problems

- Terminology:

regularizer prior

$$\Psi\ (\boldsymbol{x}) = -log(p(x))$$

data fidelity term regularization term

$$\boldsymbol{x}_{MAP} = \arg\min_{x} \frac{1}{2\sigma^2} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \Psi(\boldsymbol{x})$$

Image Priors/Regularization and Total Variation (TV)

Examples of Image Priors / Regularizers

blurry stuff

Promote smoothness!

$$\Psi(x) = \|\Delta x\|_2$$

Laplace operator

stars

Promote sparsity!

$$\Psi(x) = \|x\|_1$$

"natural" image

Promote sparse gradients!

$$\Psi(x) = \mathrm{TV}(x)$$

Total Variation (TV)

- Express (forward finite difference) gradient as convolution!

$$\boldsymbol{D}_x \boldsymbol{x} = d_x * x, d_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{D}_y \boldsymbol{x} = d_y * x, d_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$\boldsymbol{x}$

Total Variation (TV)

$\boldsymbol{x}$

better: isotropic

$$\sqrt{\boldsymbol{D}_x\boldsymbol{x})_i^2 + \boldsymbol{D}_y\boldsymbol{x})_i^2}$$

easier: anisotropic

$$\sqrt{\boldsymbol{D}_x\boldsymbol{x})_i^2 + \boldsymbol{D}_y\boldsymbol{x})_i^2}$$

Total Variation (TV)

- Examples are mostly black, indicating that gradient magnitudes are close to 0 -> natural images have sparse gradients!
- This intuition is well-captured by the TV pseudo-norm, either anisotropic or isotropic:

$$\mathrm{TV}_{\text{anisotropic}}(\boldsymbol{x}) = \|\boldsymbol{D}_x \boldsymbol{x}\|_1 + \|\boldsymbol{D}_y \boldsymbol{x}\|_1 = \sum_{i=1}^{N} |(\boldsymbol{D}_x \boldsymbol{x})_i| + |(\boldsymbol{D}_y \boldsymbol{x})_i| = \sum_{i=1}^{N} \sqrt{(\boldsymbol{D}_x \boldsymbol{x})_i^2} + \sqrt{(\boldsymbol{D}_y \boldsymbol{x})_i^2}$$

$$\mathrm{TV}_{\text{isotropic}}(\boldsymbol{x}) = \|\boldsymbol{D}\boldsymbol{x}\|_{2,1} = \sum_{i=1}^{N} \left\| \begin{bmatrix} (\boldsymbol{D}_x \boldsymbol{x})_i \\ (\boldsymbol{D}_y \boldsymbol{x})_i \end{bmatrix} \right\|_2 = \sum_{i=1}^{N} \sqrt{(\boldsymbol{D}_x \boldsymbol{x})_i^2 + (\boldsymbol{D}_y \boldsymbol{x})_i^2}$$

Total Variation (TV)

The TV pseudo-norm is one of the most popular regularization schemes for natural images!

Extensions to make it more general or applicable for other data:

- Hyper-Laplacian: Levin et al. 2009, Krishnan & Fergus 2009
- Total generalized variation: Bredies et al. 2009
- Frobenius norm of Hessian: Lefkimmiatis et al. 2003
- …

How to solve inverse problem that use these regularizers?

Solving Regularized Inverse Problem

- Objective or "loss" function of general inverse problem:

$$\text{minimize}_x \ \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\Psi(\boldsymbol{x})$$

weight of regularizer

- Practical #1 go-to solution: Adam solver implemented in PyTorch

- 3 simple steps, will explore in problem session & homework:
  1. Implement evaluation of loss function
  2. Set hyperparameters, including learning rate
  3. Run

- The "fine print": convenient but doesn't always converge well

Half-quadratic Splitting (HQS) Method

The Half-quadratic Splitting (HQS) Method

- Objective or "loss" function of general inverse problem:

$$\text{minimize}_x \ \frac{1}{2}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\underset{\text{weight of regularizer}}{\lambda}\Psi(\boldsymbol{x})$$

- Reformulate as:

$$\text{minimize}_{\{x,z\}} \ \underbrace{\frac{1}{2}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2}_{f(x)}+\underbrace{\lambda\Psi(\mathbf{z})}_{g(z)}$$

$$subject\ to\ \boldsymbol{Dx}-\boldsymbol{z}=0$$

- Remove constraints using penalty term (equivalent for large ρ):

$$L_\rho(\boldsymbol{x},\boldsymbol{z})=f(\boldsymbol{x})+g(\boldsymbol{z})+\underbrace{\frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}\|_2^2}_{\text{penalty term}}$$

The Half-quadratic Splitting (HQS) Method

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x}) + g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

- Alternating gradient descent approach to solving penalty formulation leads to following iterative algorithm:

while not converged:

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{f,\rho}(\mathbf{z}) = \arg\min_x L_\rho(\boldsymbol{x}, \mathbf{z}) = \arg\min_x f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \mathbf{z}\|_2^2$$

$$\mathbf{z} \leftarrow \mathrm{prox}_{g,\rho}(\boldsymbol{D}\boldsymbol{x}) = \arg\min_z L_\rho(\boldsymbol{x}, \mathbf{z}) = \arg\min_z g(\mathbf{z}) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \mathbf{z}\|_2^2$$

HQS for Image Deconvolution with TV

- Generic: $$L_\rho(\boldsymbol{x},\boldsymbol{z}) = f(\boldsymbol{x}) + g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$
- Deconv: $$L_\rho(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

$\mathbf{x} \in \mathbb{R}^{\mathrm{N}}$	unknown sharp image
$\boldsymbol{C} \in \mathbb{R}^{\mathrm{NxN}}$	circulant convolution matrix for known kernel ***c***
$\boldsymbol{z} \in \mathbb{R}^{2\mathrm{N}}$	slack variable, twice the size of ***x!***
$\boldsymbol{D} = \begin{bmatrix}\boldsymbol{D}_x \\ \boldsymbol{D}_y\end{bmatrix} \in \mathbb{R}^{2\mathrm{NxN}}$	finite difference gradients, horizontal & vertical

HQS for Image Deconvolution with TV

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

while not converged:

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\mathbf{z}) = \arg\min_x \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

$$\mathbf{z} \leftarrow \mathrm{prox}_{\|\cdot\|_1,\rho}(\boldsymbol{D}\boldsymbol{x}) = \arg\min_z \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \mathbf{z}\|_2^2$$

HQS for Image Deconvolution with TV

x **- update:**

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

reformulate

$$= \frac{1}{2}(\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b})^T(\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}) + \frac{\rho}{2}(\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z})^T(\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z})$$

$$= \frac{1}{2}\left(\boldsymbol{x}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{C}^T\boldsymbol{b} + \boldsymbol{b}^T\boldsymbol{b}\right) + \frac{\rho}{2}\left(\boldsymbol{x}^T\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{D}^T\boldsymbol{z} + \boldsymbol{z}^T\boldsymbol{z}\right)$$

find solution by setting gradient to 0

$$0 = \nabla_x f(\boldsymbol{x}) = \boldsymbol{C}^T\boldsymbol{C}\boldsymbol{x} - \boldsymbol{C}^T\boldsymbol{b} + \rho\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{x} - \rho\boldsymbol{D}^T\boldsymbol{Z}$$

close-form solution

$$x \leftarrow \left(C^TC + \rho D^TD\right)^{-1}\left(C^Tb + \rho D^Tz\right)$$

HQS for Image Deconvolution with TV

x **- update:**

$$\boldsymbol{x} \leftarrow \operatorname{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

$$x \leftarrow \underline{\left(C^T C + \rho D^T D\right)}^{-1} \underline{\left(C^T b + \rho D^T z\right)}$$

exploit duality of algebraic & signal processing interpretation

$$\boldsymbol{C}^T\boldsymbol{C} \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\}\} \quad \boldsymbol{D}^T\boldsymbol{z} = \boldsymbol{D}_x^T z_1 + \boldsymbol{D}_y^T z_2 \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{d_x\} * \cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\} * \cdot \mathcal{F}\{z_2\}\}$$

$$\boldsymbol{D}^T\boldsymbol{D} \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\}\} \quad \boldsymbol{C}^T\boldsymbol{b} \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\} * \cdot \mathcal{F}\{b\}\}$$

$$\underline{\boldsymbol{C}^T\boldsymbol{C} + \rho \boldsymbol{D}^T\boldsymbol{D}} \Leftrightarrow \mathcal{F}^{-1}\left\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho\left(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\}\right)\right\}$$

$$\underline{\boldsymbol{C}^T\boldsymbol{b} + \rho \boldsymbol{D}^T\boldsymbol{z}} \Leftrightarrow \mathcal{F}^{-1}\left\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\left(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{z_2\}\right)\right\}$$

HQS for Image Deconvolution with TV

x - update:

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

$$x \leftarrow \left(C^T C + \rho D^T D\right)^{-1} \left(C^T b + \rho D^T z\right)$$

- Efficient ***x***–update operates purely on 2D images with FFTs and element-wise multiplications and divisions:

$$\mathrm{prox}_{\|\cdot\|_2,\rho}(\mathbf{z}) = \mathcal{F}^{-1}\left\{ \frac{\boxed{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\}} + \rho\left(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{z_2\}\right)}{\boxed{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho\left(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\}\right)}} \right\}$$

can pre-compute most parts

$$z_1 = \boldsymbol{z}(1:N),\, z_2 = \boldsymbol{z}(N+1:2N)$$

HQS for Image Deconvolution with TV

x **- update:**

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_1,\rho}(\boldsymbol{D}\boldsymbol{x}) = \arg\min_{z} \lambda \|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

- Efficient ***z***–update operates purely on 2D images with FFTs and element-wise multiplications and divisions:

$$\mathrm{prox}_{\|\cdot\|_1,\rho}(\boldsymbol{v}) = \mathcal{S}_\kappa(\boldsymbol{v}) = \begin{cases} v - \kappa v > \kappa \\ \quad 0 \quad |v| \le \kappa = (v-\kappa)_+ - (-v-\kappa)_+ \\ v + \kappa v < -\kappa \end{cases}$$

$$k = {}^{\lambda}/_{\rho}$$

This element-wise soft thresholding is the proximal operator for anisotropic TV, see course notes on block soft thresholding for isotropic TV.

$$\boldsymbol{v} = \boldsymbol{D}\boldsymbol{x}$$

HQS for Image Deconvolution with Denoiser

x **- update:**

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2 \qquad z \epsilon \mathbb{R}^N$$

$$x \leftarrow \left(C^T C + \rho I\right)^{-1}\left(C^T b + \rho z\right)$$

no matrix ***D***!

- Efficient *x*–update operates purely on 2D images with FFTs and element-wise multiplications and divisions:

$$\mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

HQS for Image Deconvolution with Denoiser

***x* - update:**

$$z \leftarrow \operatorname{prox}_{\mathcal{D},\rho}(\boldsymbol{x}) = \arg\min_{z} \lambda\Psi(z) + \frac{\rho}{2}\|x - z\|_2^2$$
$$= \arg\min_{z} \Psi(z) + \frac{\rho}{2\lambda}\|x - z\|_2^2$$

- Efficient ***z***–update uses arbitrary denoiser ***D(·)*** , such as DnCNN and non-local means, using noise variance

$$\sigma^2 = \frac{\lambda}{\rho}$$

$$\operatorname{prox}_{\mathcal{D},\rho}(\boldsymbol{x}) = \mathcal{D}\left(\boldsymbol{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

Image Deconvolution with HQS

Image Deconvolution with HQS


Target Image


Measurements, σ=0.1


Wiener Deconv., PSNR 19.5 dB


Adam+TV, PSNR 26.1 dB


HQS+TV, PSNR 26.3 dB


HQS+DnCNN, PSNR 26.7 dB

Image Deconvolution with HQS

香港中文大學(深圳) The Chinese University of Hong Kong, Shenzhen 点昀 POINT SPREAD

HQS for deconvolution with denoiser

1: initialize ρ and λ
2: $x = zeros\,(W, H)\,;$
3: $z = zeros\,(W, H)\,;$
4: **for** $k = 1$ **to** max_iters **do**
5: $\quad x = \mathbf{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$
6: $\quad z = \mathbf{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$
7: **end for**

HQS for deconvolution with TV

1: initialize ρ and λ
2: $x = zeros\,(W, H)\,;$
3: $z = zeros\,(W, H)\,;$
4: **for** $k = 1$ **to** max_iters **do**
5: $\quad x = \mathbf{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{z_2\})}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\})}\right\}$
6: $\quad z = \mathbf{prox}_{\|\cdot\|_1,\rho}(\mathbf{Dx}) = \mathcal{S}_{\lambda/\rho}(\mathbf{Dx})$
7: **end for**

HQS - Convergence Criterion

- Run or "unroll" HQS for K iterations
- Run until change in residual between iterations is < threshold

$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

$$\vdots$$

Outlook on Unrolled Optimization

Outlook on Unrolled Optimization

- Run or “unroll” HQS for K iterations
- Interpret as unrolled feedforward network:

$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

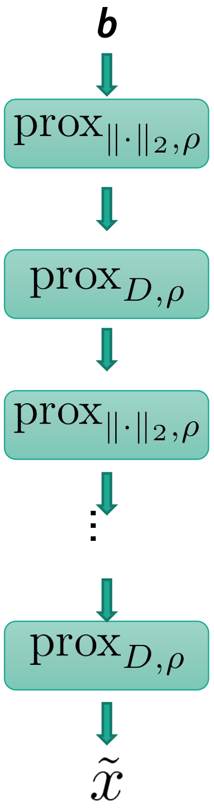
$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

$$x = \mathrm{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

$$z = \mathrm{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

$$\vdots$$



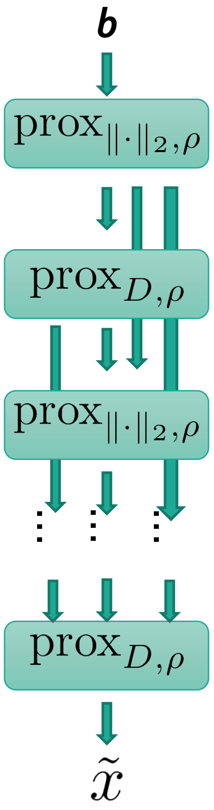
Credit: Diamond et al. 2017

Outlook on Unrolled Optimization

- Run or "unroll" HQS for K iterations
- Interpret as unrolled feedforward network:

Benefits over unrolled optimization

- Learnable parameters: $\lambda^{(k)}$, $\rho^{(k)}$ denoiser
- DenseNet-like skip connections $D^{(k)}$
- Denoiser/regularizer can adapt to matrix ***C***
- Can train with advanced loss functions (perceptual, adversarial, other network, …)



Credit: Diamond et al. 2017

Today's Topic

- Image Deconvolution – Brief Review
- A Bayesian Perspective of Inverse Problems
- Image Priors/Regularization and Total Variation (TV)
- Half-Quadratic Splitting (HQS) Method
- Image Deconvolution with HQS
- Outlook on Unrolled Optimization

GAMES 204

Thank You!

Qilin Sun（孙启霖）

香港中文大学（深圳）

点昀技术（Point Spread Technology）