PGMO Lecture: Vision, Learning and Optimization

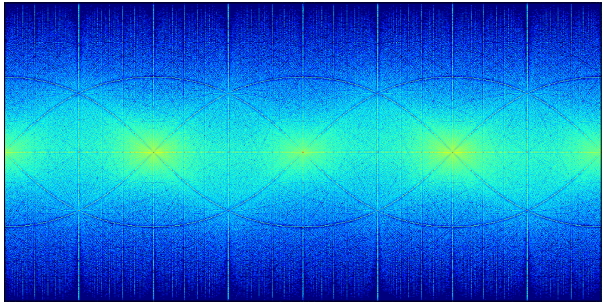
8. Learning

Thomas Pock

Institute of Computer Graphics and Vision

February 12, 2020

Introduction



- So far we have considereded mainly convex models based on the total variation, which however only served as a crude approximation to the true image statistics.
- In this chapter we will discuss methods to learn better variational models from data.
- We will start be learning just the regularization parameter but then will also learn filters and potential functions.
- Finally, we will also consider deep-learning inspired architectures that achieve state-of-the-art performance.

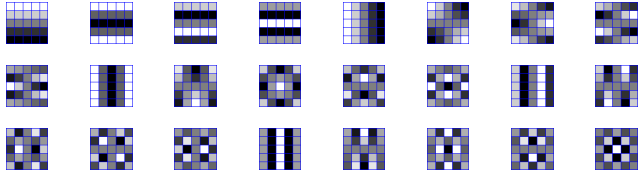
Overview

Learning regularization parameters

- In [Kunisch, P. '12] we considered a weighted sum of ℓ_1 regularizers:

$$\mathcal{R}(u) = \sum_{k=1}^{N_K} \vartheta_k \left\| K_k u \right\|_1 = \sum_{k=1}^{N_K} \sum_{i,j} \vartheta_k |(K_k u)_{i,j}|,$$

  where K_k are linear operators and $\vartheta_k \geq 0$ are the regularization weights.
- Can be see as a generalization of the total variation
- Usually, we restrict the linear operators to small convolution kernels f_k with the property that $K_k u \Leftrightarrow f_k * u$
- From JPEG compression, it is known that images have a sparse representation in terms of DCT basis functions.



The 24 DCT5 filters f_k

Bilevel optimization

- How can we choose optimal weights for the different operators?

Bilevel optimization

- How can we choose optimal weights for the different operators?
- In machine learning a popular approach is empirical risk minimization adopting a loss function.
- We assume we have given training data $(f_s, g_s)_{s=1}^S$ consisting of noisy observations f_s and ground truth reconstructions g_s.
- Applying this idea to our image reconstruction problems leads to a bilevel optimization problem [Kunisch, P. '12]

$$\begin{cases} \min\limits_{\vartheta \geq 0} \dfrac{1}{2} \displaystyle\sum_{s=1}^{S} \| u_s(\vartheta) - g_s \|_2^2 \\ \text{s.t. } u_s(\vartheta) = \arg\min\limits_{u} \displaystyle\sum_{k=1}^{N_K} \vartheta_k \| K_k u \|_1 + \dfrac{1}{2} \| u - f_s \|_2^2 \, . \end{cases}$$

Bilevel optimization

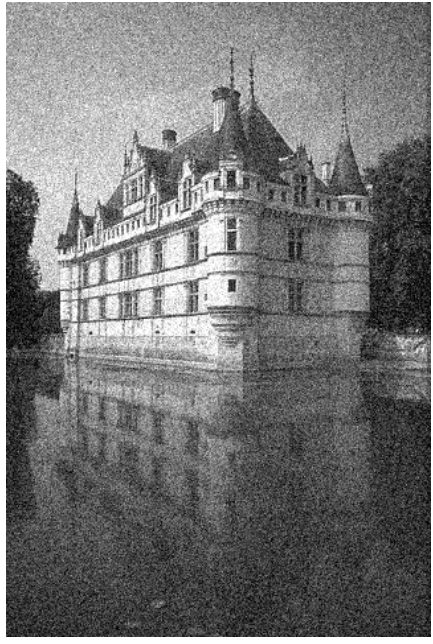
- How can we choose optimal weights for the different operators?
- In machine learning a popular approach is empirical risk minimization adopting a loss function.
- We assume we have given training data $(f_s, g_s)_{s=1}^S$ consisting of noisy observations f_s and ground truth reconstructions g_s.
- Applying this idea to our image reconstruction problems leads to a bilevel optimization problem [Kunisch, P. '12]

$$\begin{cases} \min\limits_{\vartheta \geq 0} \dfrac{1}{2} \displaystyle\sum_{s=1}^{S} \|u_s(\vartheta) - g_s\|_2^2 \\ \text{s.t. } u_s(\vartheta) = \arg\min\limits_{u} \displaystyle\sum_{k=1}^{N_K} \vartheta_k \|K_k u\|_1 + \dfrac{1}{2} \|u - f_s\|_2^2 \,. \end{cases}$$

- Interpretation: We try to find parameters ϑ such that the minimizers of the variational model minimizes the loss function
- Closely related approaches: [Haber and Tenorio, '02], [Samuel and Tappen '09], [Peyré and Fadili '11], [De Los Reyes and Schönlieb '12], ...
- We developed semi-smooth Newton algorithms to solve the bilevel optimization problem for the optimal parameter vector ϑ.

Example: Image denoising

Original image



Noisy image

Example: Image denoising

Original image

TV denoised

Example: Image denoising

Original image

DCT5

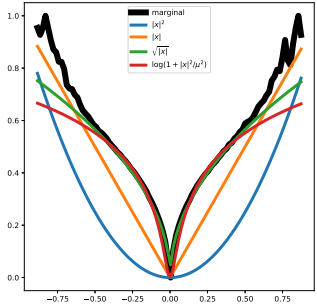
Overview

The Fields of Experts model

- Recall that the $|x|$ function does not provide a very accurate match to the marginal distributions of zero-mean filters
- A much better match is obtained by the negative log student's-t distribution $\log(1 + |x|^2/\mu^2)$ [Huang and Mumford '99].
- Let us consider the following nonconvex model [Roth, Black '09], [Samuel, Tappen '09], called the "Fields of Experts" model:

$$\mathcal{R}(u) = \sum_{k=1}^{N_K} \sum_{i,j} \rho_k((K_k u)_{i,j}),$$

  where K_k are again linear operators implementing 2D convolutions with small filters f_k, that is $f_k * u \Leftrightarrow K_k u$, and $\rho_k(t) = \alpha_k \log(1 + |t|^2)$.
- In contrast to the previous model, also the filters are learned.



Bilevel optimization

- We again consider training data consisting of clean and noisy images $(f_s, g_s)_{s=1}^S$
- We again used a bilevel optimization approach to learn the filters and functions $\vartheta = (f_k, \alpha_k)_{k=1}^{N_K}$

$$\begin{cases} \min\limits_{\vartheta} L(\vartheta) = \dfrac{1}{2}\displaystyle\sum_{s=1}^{S} \|u_s(\vartheta) - g_s\|^2 + R(\vartheta) \\ \text{s.t. } u_s(\vartheta) = \arg\min\limits_{u} \displaystyle\sum_{k=1}^{N_K}\sum_{i,j} \rho_k((K_k u)_{i,j}) + \dfrac{1}{2}\|u - f_s\|_2^2, \end{cases}$$

where $R(\vartheta)$ is a regularization term for the learned parameters, for example one could consider the constraints

$$\mathbf{1}^T f_k = 0, \quad \alpha_k \geq 0, \quad k = 1, ..., K$$

Lagrangian

- In order to compute gradients of the loss function with respect to ϑ, we replace the lower-level optimization problem by its first-order optimality condition (assuming $s = 1$ and dropping the index):

$$\sum_{k=1}^{N_K} K_k^* \phi_k(K_k u) + u - f = 0, \quad \phi_k(y) = \mathrm{diag}(\rho_k'(y_1), ..., \rho_k'(y_n)),$$

  where K_k^* denotes the adjoint filter and consider the Lagrangian functional

$$\mathcal{L}(u, \vartheta, \lambda) = \|u - g\|^2 + R(\vartheta) + (\sum_{k=1}^{N_K} K_k^* \phi_k(K_k u) + u - f)^T p,$$

  where p is a vector of Lagrange multipliers.

- Assuming the existence of a regular local minimum in (u, ϑ), we can invoke the classical Lagrange multiplier theorem, which guarantees the existence of multipliers p such that:

$$\begin{pmatrix} (\sum_{k=1}^{N_K} K_k^* D\phi_k(K_k u) K_k + I)p + u - g \\ D_\vartheta R(\vartheta) + (D_\vartheta \sum_{k=1}^{N_K} K_k^* \phi_k(K_k u))p \\ \sum_{k=1}^{N_K} K_k^* \phi_k(K_k u) + u - f \end{pmatrix} = 0.$$

Implicit differentiation

- For fixed ϑ, the system can be reduced by first solving the lower level problem (last equation) for u^*, that is

$$\sum_{k=1}^{N_K} K_k^* \phi_k(K_k u^*) + u^* - f = 0,$$

then one can solve for p^* by solving the linear system

$$p^* = (\sum_{k=1}^{N_K} K_k^* D\phi_k(K_k u^*) K_k + I)^{-1}(g - u^*),$$

and finally the gradient of the loss function with respext to ϑ is given by

$$\partial_\vartheta L(\vartheta) = D_\vartheta R(\vartheta) + (D_\vartheta \sum_{k=1}^{N_K} K_k^* \phi_k(K_k u^*))(\sum_{k=1}^{N_K} K_k^* D\phi_k(K_k u^*) K_k + I)^{-1}(g - u^*),$$

which is nothing else then implicit differentiation.

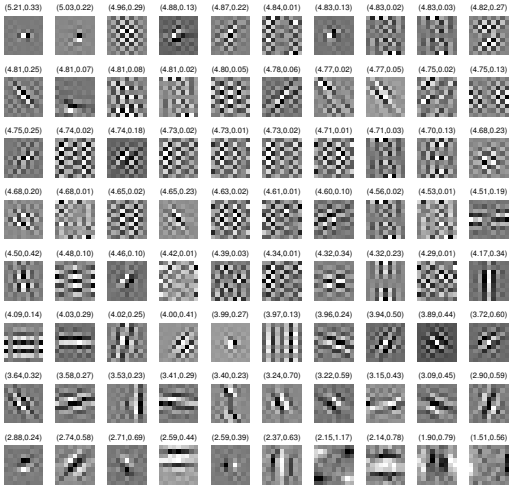
- The loss function can then be minimized using any gradient-based optimization algorithm.

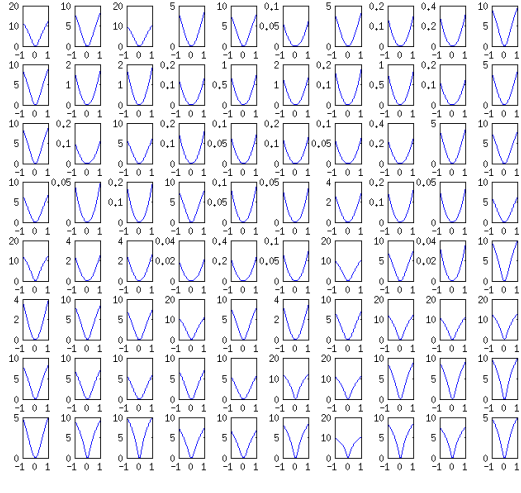
The learned filters and functions

- In [Chen, Ranftl, P. '14] we learned 80 filters of size 9×9 plus function parameters $\rightarrow$ 6480 parameters on a database of $\sim$ 200 images using bilevel optimization
- ... two weeks later ...

The learned filters and functions

- In [Chen, Ranftl, P. '14] we learned 80 filters of size 9×9 plus function parameters $\rightarrow$ 6480 parameters on a database of $\sim$ 200 images using bilevel optimization
- ... two weeks later ...





Evaluation

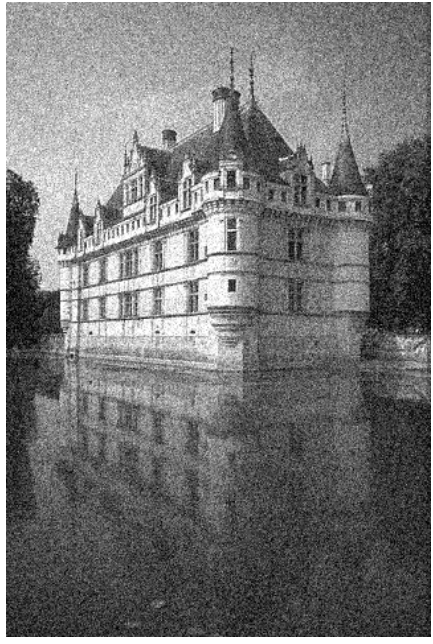
- Comparison with five state-of-the-art approaches: K-SVD [Elad and Aharon '06], FoE [Q. Gao and Roth '12], BM3D [Dabov et al. '07], GMM [D. Zoran et al. '12], LSSC [Mairal et al. '09]
- We report the average PSNR on 68 images of the Berkeley image data base [Chen, P. 14]

σ	KSVD	FoE	BM3D	GMM	LSSC	BL7x7	BL9x9
15	30.87	30.99	31.08	31.19	**31.27**	31.18	31.22
25	28.28	28.40	28.56	28.68	**28.70**	28.66	**28.70**
50	25.17	25.35	25.62	25.67	25.72	25.70	**25.76**

- Performs as well as state-of-the-art

Denoising results for $\sigma = 25$

Original image



Noisy image

Denoising results for $\sigma = 25$

Original image

TV denoised

Denoising results for $\sigma = 25$

Original image

FoE prior

`foe.ipynb`

Computing gradients

- Bilevel optimization is heavily time consuming since for implicit differentiation we need to:
  - Solve the lower problems exactly
  - Invert the Hessian of the lower level problem
- Performance strongly depends on the error of the stationary point u^*

Computing gradients

- Bilevel optimization is heavily time consuming since for implicit differentiation we need to:
  - Solve the lower problems exactly
  - Invert the Hessian of the lower level problem
- Performance strongly depends on the error of the stationary point u^*

- Alternative: Unroll the steps of an iterative algorithm
- The bilevel optimization problem becomes

$$\begin{cases} \min_{\vartheta} \frac{1}{2} \sum_{s=1}^{S} \left\| u_s^T(\vartheta) - g_s \right\|^2 \\ \text{s.t. } u_s^{t+1} = u_s^t - \tau^t \left(\sum_{k=1}^{N_K} K_k^* \phi_k(K_k u_s) + (u_s^t - f) \right), \\ t = 0 \dots T-1 \end{cases}$$

- We can compute the exact gradient with respect to the model parameters using the backpropagation algorithm.
- It turns out that taking only a finite number of steps works even better....

Overview

Motivation

As a motivating example, let us step back to a smooth approximation of the TV–L^2 (ROF) model

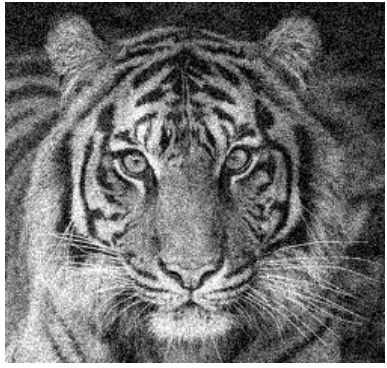
$$E_\epsilon[u,\nu] = \nu \sum_{i,j} \sqrt{|(\mathrm{D}u)_{i,j}|^2 + \epsilon^2} + \frac{1}{2}\,\|u-g\|_2^2\,,$$

and for various weighting parameters ν the gradient flow with step size τ

$$u_{s+1} = u_s - \tau\left(\nu \mathrm{D}^*\left(\frac{\mathrm{D}u_s}{\sqrt{|(\mathrm{D}u_s)|^2+\epsilon^2}}\right) + u_s - g\right)$$

TV–L^2 classical

u_0

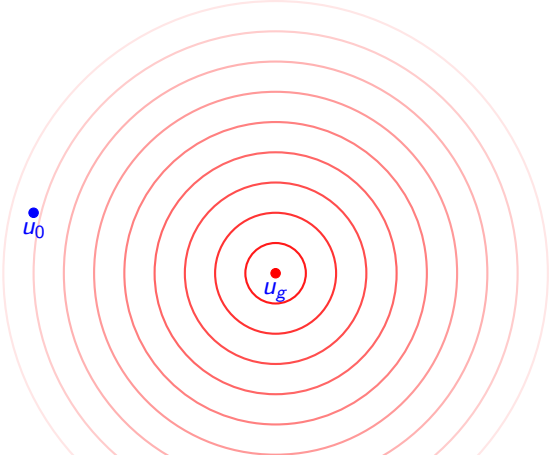


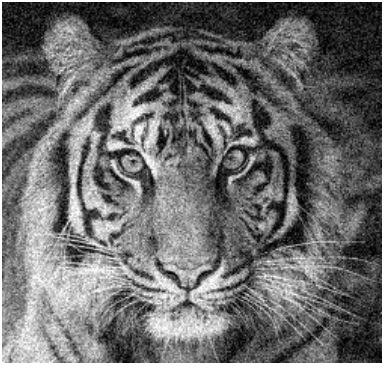
TV–L^2 classical

u_0

u_g

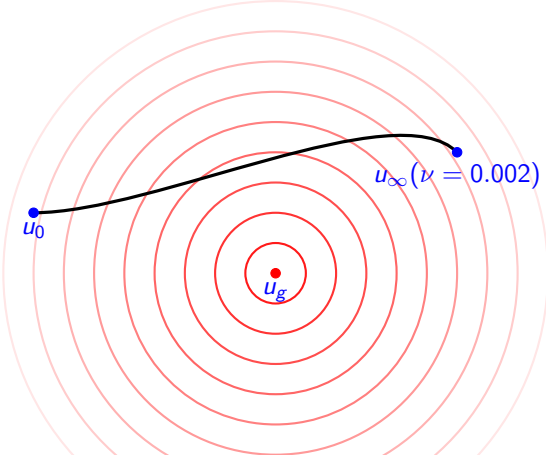
TV–L^2 classical





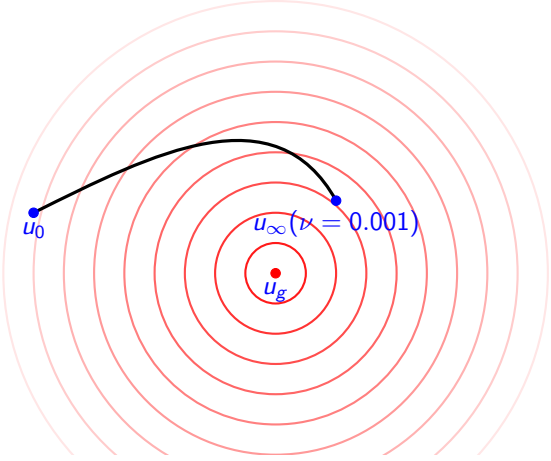
PSNR = 20.53

TV–L^2 classical



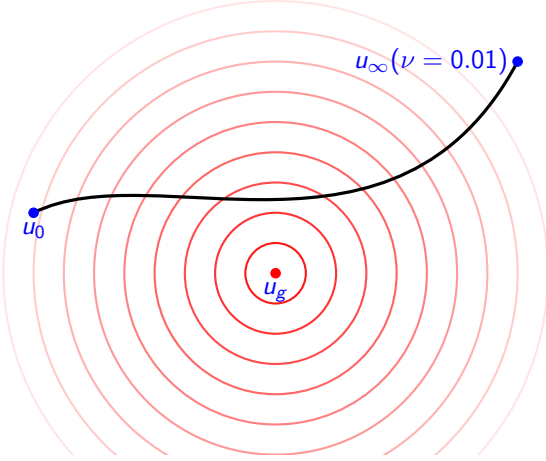
PSNR = 24.26

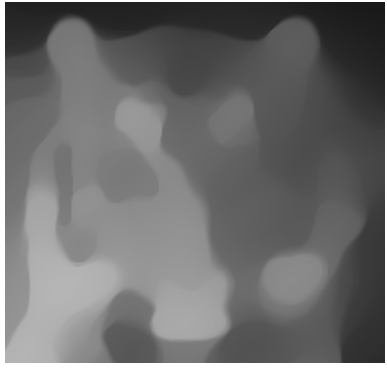
TV–L^2 classical



PSNR = 26.59

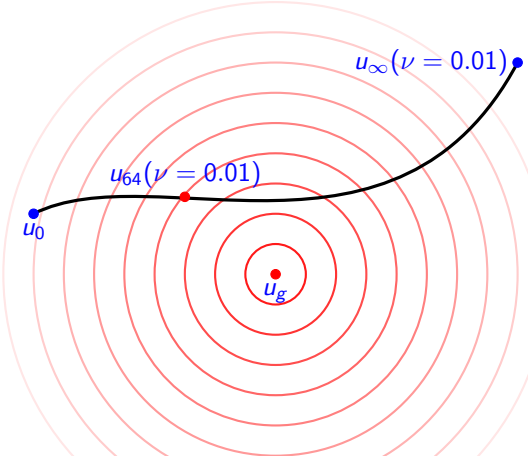
TV–L^2 classical

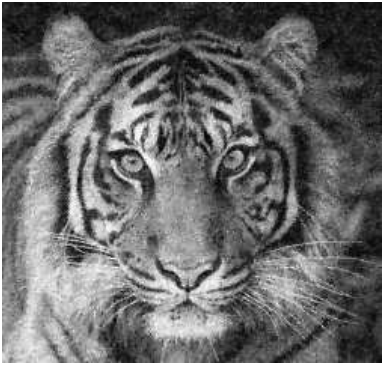




PSNR = 19.61

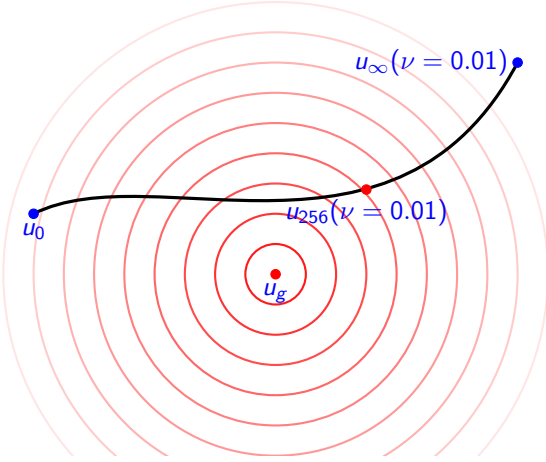
TV–L^2 with early stopping





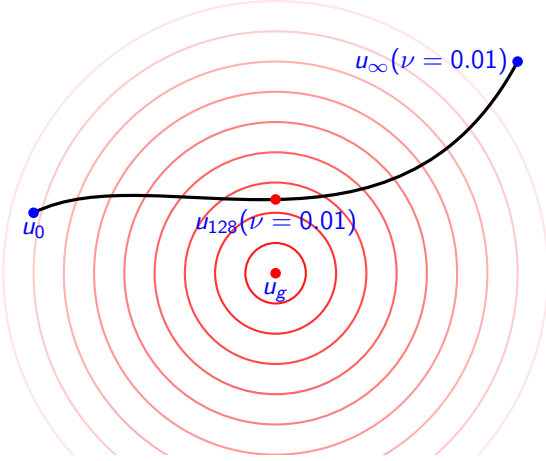
PSNR = 25.23

TV–L^2 with early stopping



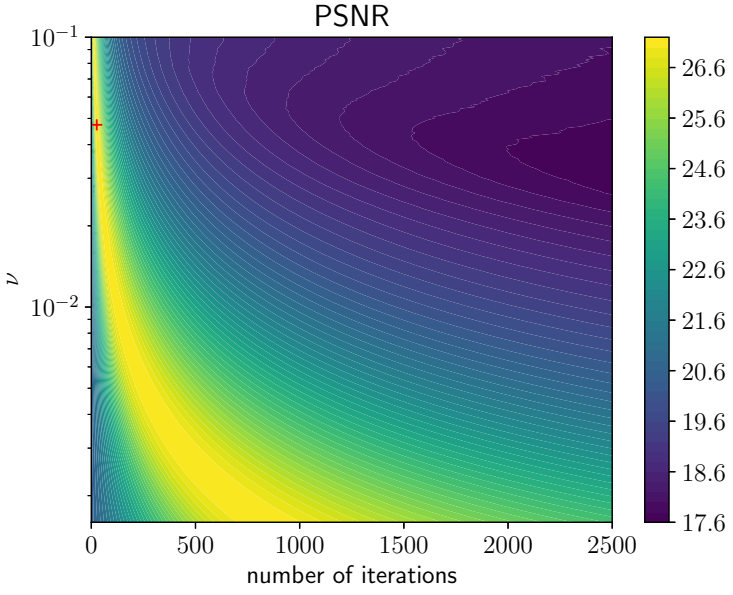
PSNR = 25.55

TV–L^2 with early stopping



PSNR = 27.19

Parameter search



- The best performance (red cross) is achieved for early stopping.
- When minimizing the energy exactly the solution is "oberfitted" to the variational model and gives inferior results.

A gradient flow perspective

Let $u \in \mathbb{R}^n$ be a data vector, then the variational energy is

$$\mathcal{E}[u] = \mathcal{D}[u] + \mathcal{R}[u].$$

A gradient flow perspective

Let $u \in \mathbb{R}^n$ be a data vector, then the variational energy is

$$\mathcal{E}[u] = \mathcal{D}[u] + \mathcal{R}[u].$$

Fields-of-Experts regularization:

$$\mathcal{R}[u] = \sum_{i=1}^{m} \sum_{k=1}^{N_K} \rho_k((K_k u)_i)$$

with $K_k \in \mathbb{R}^{m \times n}$ and associated nonlinear functions $\rho_k : \mathbb{R} \to \mathbb{R}$

A gradient flow perspective

Let $u \in \mathbb{R}^n$ be a data vector, then the variational energy is

$$\mathcal{E}[u] = \mathcal{D}[u] + \mathcal{R}[u].$$

Fields-of-Experts regularization:

$$\mathcal{R}[u] = \sum_{i=1}^{m} \sum_{k=1}^{N_K} \rho_k((K_k u)_i)$$

with $K_k \in \mathbb{R}^{m \times n}$ and associated nonlinear functions $\rho_k : \mathbb{R} \to \mathbb{R}$ data fidelity:

$$\mathcal{D}[u] = \frac{1}{2}\|Au - b\|_2^2$$

$A \in \mathbb{R}^{l \times n}$ and $b \in \mathbb{R}^l$ fixed

A gradient flow perspective

Gradient flow of energy $\mathcal{E}$ for a time $t \in (0, T)$:

$$\begin{aligned}\dot{\tilde{x}}(t) = f(\tilde{x}(t), (K_k, \Phi_k)_{k=1}^{N_K}) &= -D\mathcal{E}[\tilde{x}(t)] \\ &= -A^*(A\tilde{x}(t) - b) - \sum_{k=1}^{N_K} K_k^* \Phi_k(K_k \tilde{x}(t)), \\ \tilde{x}(0) &= x_0,\end{aligned}$$

with $\tilde{x} \in C^1([0, T], \mathbb{R}^n)$, $T \in \mathbb{R}_0^+$ and the functions $\Phi_k \in \mathcal{V}^s$ are given by

$$(y_1, \dots, y_m)^\top \mapsto (\rho_k'(y_1), \dots, \rho_k'(y_m))^\top,$$

with $\mathcal{V}^s$ finite dimensional subspace of $C^s(\mathbb{R}^m, \mathbb{R}^m)$

Optimal control problem

Reparametrization: $x(t) = \tilde{x}(t\,T)$

Optimal control problem

Reparametrization: $x(t) = \tilde{x}(tT)$
optimal control problem:

$$\min_{T \in \mathbb{R}, K_k \in \mathbb{R}^{m \times n}, \Phi_k \in \mathcal{V}^s} J(T, (K_k, \Phi_k)_{k=1}^{N_K})$$

Optimal control problem

Reparametrization: $x(t) = \tilde{x}(t\,T)$
optimal control problem:

$$\min_{T\in\mathbb{R},K_k\in\mathbb{R}^{m\times n},\Phi_k\in\mathcal{V}^s} J(T,(K_k,\Phi_k)_{k=1}^{N_K})$$

cost functional:

$$J(T,(K_k,\Phi_k)_{k=1}^{N_K}) \coloneqq \frac{1}{2}\|x(1)-x_g\|_2^2$$

Optimal control problem

Reparametrization: $x(t) = \tilde{x}(t\,T)$
optimal control problem:

$$\min_{T\in\mathbb{R},K_k\in\mathbb{R}^{m\times n},\Phi_k\in\mathcal{V}^s} J(T,(K_k,\Phi_k)_{k=1}^{N_K})$$

cost functional:

$$J(T,(K_k,\Phi_k)_{k=1}^{N_K}) \coloneqq \frac{1}{2}\|x(1)-x_g\|_2^2$$

constraints:

$$0 \le T \le T_{\max}, \quad \alpha(K_k) \le 1, \quad \beta(\Phi_k) \le 1, \quad K_k\mathbf{1} = 0 \in \mathbb{R}^m,$$

$\alpha : \mathbb{R}^{m\times n} \to \mathbb{R}_0^+$ and $\beta : \mathcal{V}^s \to \mathbb{R}_0^+$ are continuously differentiable, coercive functions

Optimal control problem

Reparametrization: $x(t) = \tilde{x}(tT)$
optimal control problem:

$$\min_{T\in\mathbb{R}, K_k\in\mathbb{R}^{m\times n}, \Phi_k\in\mathcal{V}^s} J(T, (K_k, \Phi_k)_{k=1}^{N_K})$$

cost functional:

$$J(T, (K_k, \Phi_k)_{k=1}^{N_K}) := \frac{1}{2}\|x(1) - x_g\|_2^2$$

constraints:

$$0 \le T \le T_{\max}, \quad \alpha(K_k) \le 1, \quad \beta(\Phi_k) \le 1, \quad K_k\mathbf{1} = 0 \in \mathbb{R}^m,$$

$\alpha : \mathbb{R}^{m\times n} \to \mathbb{R}_0^+$ and $\beta : \mathcal{V}^s \to \mathbb{R}_0^+$ are continuously differentiable, coercive functions
transformed state equation for $t \in (0, 1)$:

$$\dot{x}(t) = Tf(x(t), (K_k, \Phi_k)_{k=1}^{N_K}), \qquad x(0) = x_0$$

First order condition

Theorem (First order necessary condition)

Let $s \geq 1$. For each stationary point $(\overline{T}, (\overline{K}_k, \overline{\Phi}_k)_{k=1}^{N_K})$ of J with state $\overline{x}$

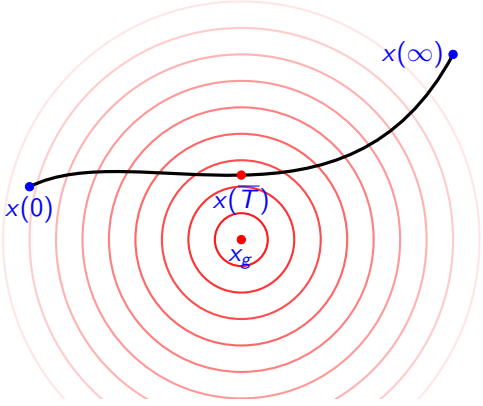
$$\int_0^1 \langle \overline{p}(t), \dot{\overline{x}}(t) \rangle \, \mathrm{d}t = 0$$

holds true. Here, $\overline{p} \in C^1([0,1], \mathbb{R}^n)$ denotes the adjoint state of $\overline{x}$, which is given as the solution to the ODE

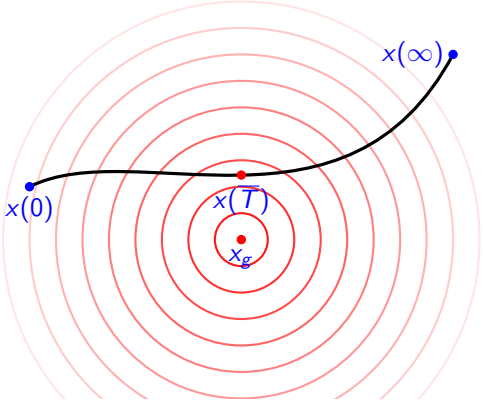
$$\dot{\overline{p}}(t) = \sum_{k=1}^{N_K} \overline{K}_k^* D\overline{\Phi}_k(\overline{K}_k \overline{x}(t)) \overline{K}_k \overline{p}(t) + A^* A \overline{p}(t)$$

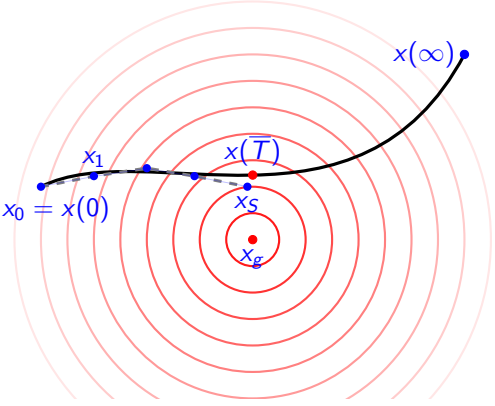
with terminal condition $\overline{p}(1) = x_g - \overline{x}(1)$.

Time discretization



Time discretization





Time discretization

Let $S \geq 2$ be a fixed depth and $\overline{\Theta} = ((\overline{K}_k, \overline{\Phi}_k)_{k=1}^{N_K})$.
State equation:
Explicit forward Euler:

$$x_{s+1} = x_s + \frac{T}{S} f(x_s, \overline{\Theta}) \quad s = 0, \ldots, S-1$$

Time discretization

Let $S \geq 2$ be a fixed depth and $\overline{\Theta} = ((\overline{K}_k, \overline{\Phi}_k)_{k=1}^{N_K})$.
State equation:
Explicit forward Euler:

$$x_{s+1} = x_s + \frac{T}{S} f(x_s, \overline{\Theta}) \quad s = 0, \ldots, S-1$$

Explicit 2$^{\text{nd}}$–order Heun:

$$x_{s+1} = x_s + \frac{T}{2S}\left(f(x_s, \overline{\Theta}) + f\left(x_s + \frac{T}{S} f(x_s, \overline{\Theta})\right)\right)$$

Time discretization

Let $S \geq 2$ be a fixed depth and $\overline{\Theta} = ((\overline{K}_k, \overline{\Phi}_k)_{k=1}^{N_K})$.
Adjoint state equation:

$$\dot{p}(t) = g(x(t), p(t), \overline{\Theta}) = \sum_{k=1}^{N_K} \overline{K}_k^* D\overline{\Phi}_k(\overline{K}_k x(t))\overline{K}_k p(t) + A^* A p(t)$$

Explicit forward Euler:

$$p_s = p_{s+1} - \frac{T}{S} g(x_{s+1}, p_{s+1}, \overline{\Theta})$$

Explicit 2nd–order Heun:

$$p_s = p_{s+1} - \frac{T}{2S}\left(g(x_{s+1}, p_{s+1}, \overline{\Theta}) + g\left(x_s, p_{s+1} - \frac{T}{S} g(x_{s+1}, p_{s+1}, \overline{\Theta}), \overline{\Theta}\right)\right)$$

Learning

For a given training set $(x_0^i, x_g^i)_{i \in \mathcal{I}}$, the loss for a batch $\mathcal{B} \subset \mathcal{I}$ is given by

$$J_{\mathcal{B}}(T, (K_k, w_k)_{k=1}^{N_K}) := \frac{1}{2|\mathcal{B}|} \sum_{i \in \mathcal{B}} \|x_S^i - x_g^i\|_2^2$$

Learning

For a given training set $(x_0^i, x_g^i)_{i\in\mathcal{I}}$, the loss for a batch $\mathcal{B} \subset \mathcal{I}$ is given by

$$J_{\mathcal{B}}(T, (K_k, w_k)_{k=1}^{N_K}) := \frac{1}{2|\mathcal{B}|} \sum_{i\in\mathcal{B}} \|x_S^i - x_g^i\|_2^2$$

subject to

$$K_k \in \mathcal{K} = \left\{K \in \mathbb{R}^{m\times n} : \alpha(K) \le 1, K\mathbf{1} = 0\right\},$$
$$w_k \in \mathcal{W} = \left\{w \in \mathbb{R}^{N_w} : \beta(w) \le 1\right\}.$$

Here

$$\rho'(x) = \sum_{j=1}^{N_w} w_j \psi_j(x)$$

with quadratic B–spline basis functions $\psi_j(x) \in C^1(\mathbb{R})$.

Image restoration

image denoising

$$b = x_g + n$$

where

$$n \sim \mathcal{N}(0, \sigma^2 I).$$

Default: $\sigma = 0.1$

Image restoration

image denoising

$$b = x_g + n$$

where

$$n \sim \mathcal{N}(0, \sigma^2 I).$$

Default: $\sigma = 0.1$

image deblurring

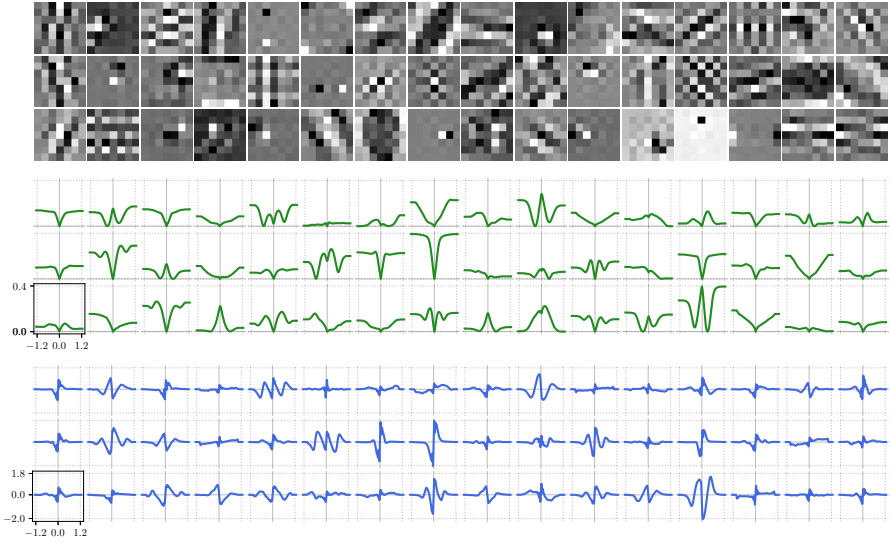
$$b = A_\tau x_g + n$$

with blur operator A_τ

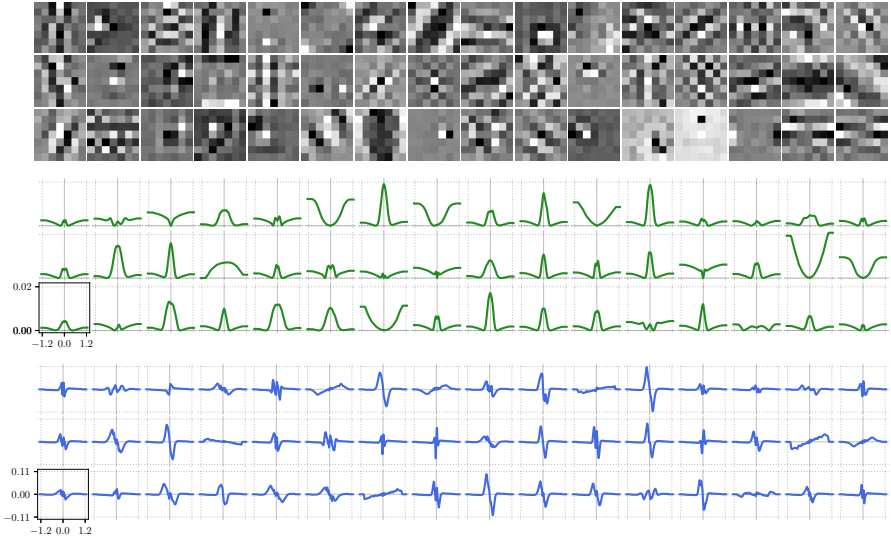
$$(x, y) \mapsto \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{x^2 + y^2}{2\tau^2}\right).$$

Default: $\tau = 1.5$ and $\sigma = 0.01$

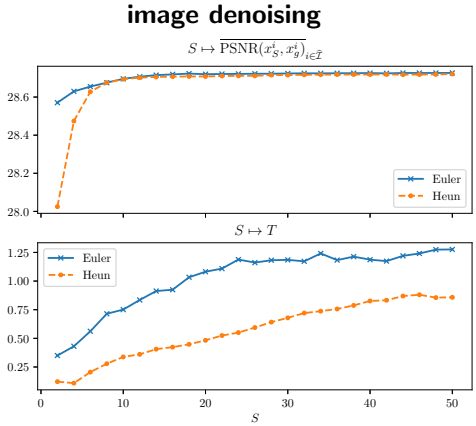
Regularization parameters - denoising



Regularization parameters - deblurring



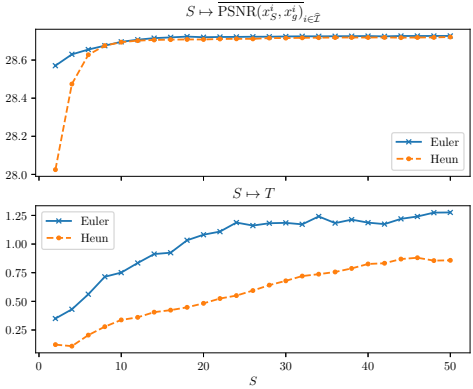
Early stopping



Inference speed: 5.694ms for $S = 20$

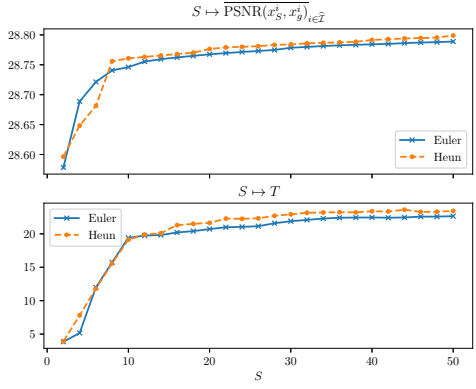
Early stopping

image denoising



Inference speed: 5.694ms for $S = 20$

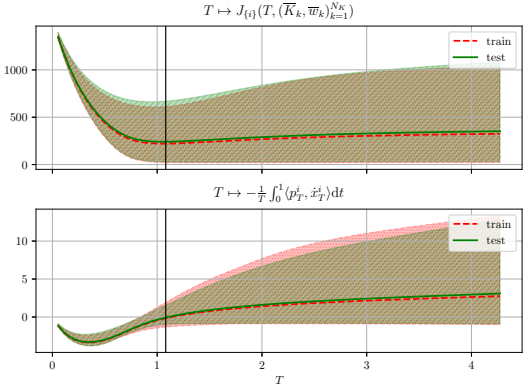
image deblurring



Inference speed: 8.687ms for $S = 20$

First order condition

image denoising



First order condition

image denoising

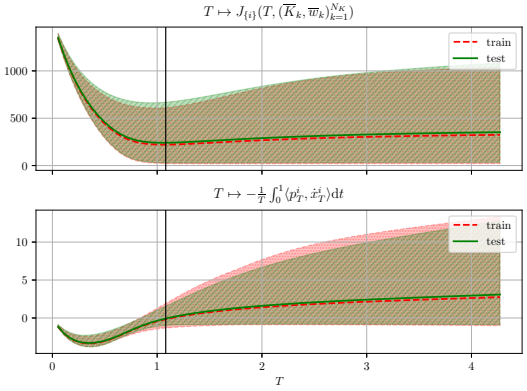
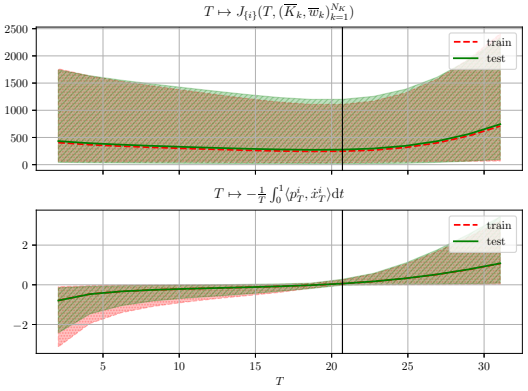


image deblurring



First order condition

PSNR=20.00

$S = 0$ — $T = 0$

PSNR=26.68

First order condition

PSNR=26.36

$S = 10$ $T = \frac{T}{2}$

PSNR=28.87

First order condition

PSNR=29.68

$S = 20$ $T = \overline{T}$

PSNR=29.52

First order condition

PSNR=29.15

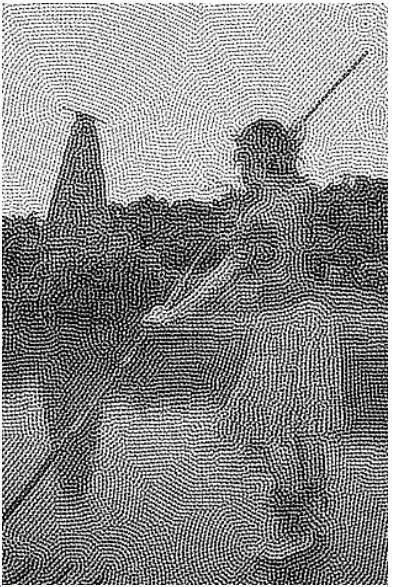
$S = 30$ $T = \frac{3\overline{T}}{2}$

PSNR=25.34

First order condition

PSNR=27.90

$S = 1000$ $T = 50\overline{T}$



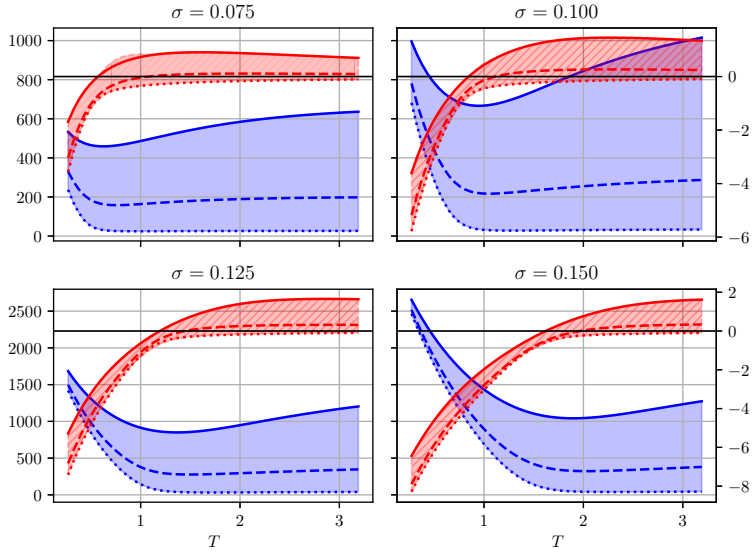
PSNR=5.13

Stopping time $\Longleftrightarrow$ Noise level

Does the stopping time depend on the noise level σ?

Stopping time $\iff$ Noise level

$$\mathcal{D}[x] = \frac{1}{2\sigma^2}\|x - b\|_2^2$$



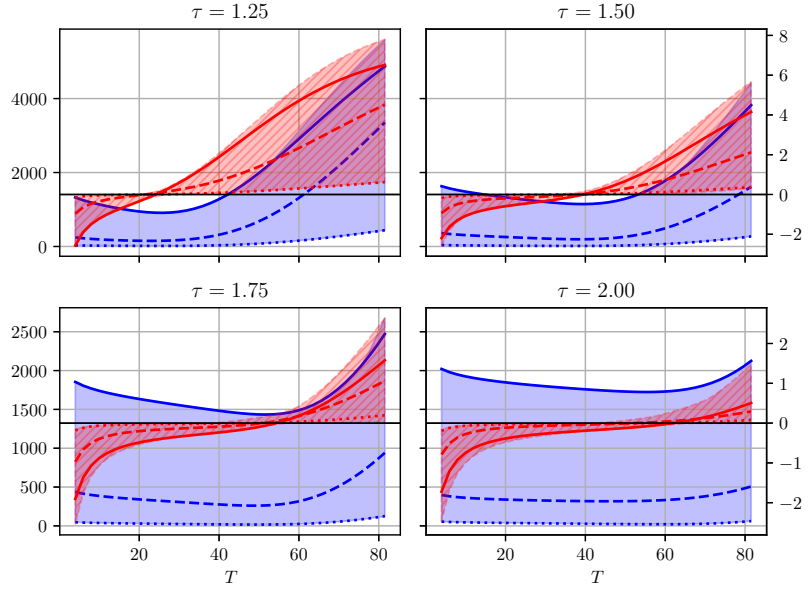
Stopping time $\Longleftrightarrow$ Noise level

	$\sigma = 0.075$		$\sigma = 0.1$		$\sigma = 0.125$		$\sigma = 0.15$	
	$\overline{\text{PSNR}}$	$\overline{T}$	$\overline{\text{PSNR}}$	$\overline{T}$	$\overline{\text{PSNR}}$	$\overline{T}$	$\overline{\text{PSNR}}$	$\overline{T}$
full optimization of all controls	30.05	0.724	28.72	1.082	27.72	1.445	26.95	1.433
optimization only of $\overline{T}$	30.00	0.757			27.73	1.514	26.95	2.055

Stopping time $\Longleftrightarrow$ Blur strength

Does the stopping time depend on the blur strength τ?

Stopping time $\Longleftrightarrow$ Blur strength



Stopping time $\Longleftrightarrow$ Blur strength

	$\tau = 1.25$		$\tau = 1.5$		$\tau = 1.75$		$\tau = 2.0$	
	$\overline{\text{PSNR}}$	$\overline{T}$	$\overline{\text{PSNR}}$	$\overline{T}$	$\overline{\text{PSNR}}$	$\overline{T}$	$\overline{\text{PSNR}}$	$\overline{T}$
full optimization of all controls	29.95	39.86	28.76	37.78	27.87	40.60	27.13	40.01
optimization only of $\overline{T}$	29.73	23.86			27.69	47.72	26.71	51.70

Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers

Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs $(\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n$ via

$$\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j$$

Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs $(\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n$ via

$$\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j$$

$\rightarrow$ forward Euler scheme reduces to

$$v_j - \frac{T}{S} \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,$$

Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs $(\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n$ via

$$\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j$$

$\rightarrow$ forward Euler scheme reduces to

$$v_j - \frac{T}{S} \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,$$

- contrast factor $(1 - \frac{\lambda_j T}{S})$ determines global contrast change

Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs $(\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n$ via

$$\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j$$

$\rightarrow$ forward Euler scheme reduces to

$$v_j - \frac{T}{S} \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,$$

- contrast factor $(1 - \frac{\lambda_j T}{S})$ determines global contrast change
- holds only for one step

Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs $(\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n$ via

$$\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j$$

$\rightarrow$ forward Euler scheme reduces to

$$v_j - \frac{T}{S} \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,$$

- contrast factor $(1 - \frac{\lambda_j T}{S})$ determines global contrast change
- holds only for one step
- eigenvalue determines contrast preservation

Estimation of generalized eigenpairs

Compute N_v generalized eigenpairs by solving

$$\min_{\{v_j\}_{j=1}^{N_v}} \sum_{j=1}^{N_v} \left\| \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) - \Lambda(v_j) v_j \right\|_2^2,$$

where

$$\Lambda(v) = \frac{\left\langle \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v), v \right\rangle}{\|v\|_2^2}$$

is generalized Rayleigh quotient.

Estimation of generalized eigenpairs

Compute N_v generalized eigenpairs by solving

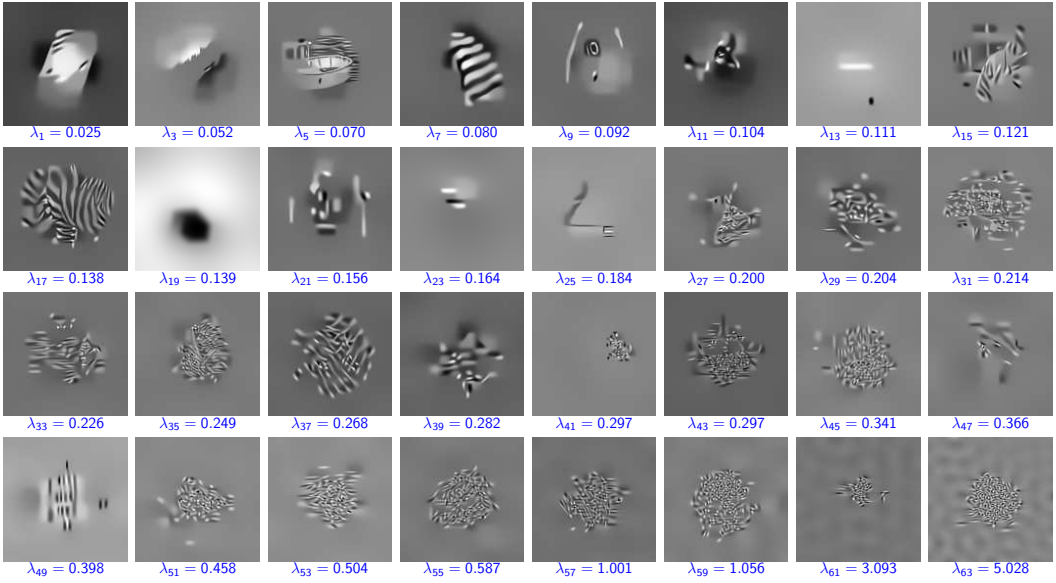
$$\min_{\{v_j\}_{j=1}^{N_v}} \sum_{j=1}^{N_v} \left\| \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) - \Lambda(v_j) v_j \right\|_2^2,$$

where

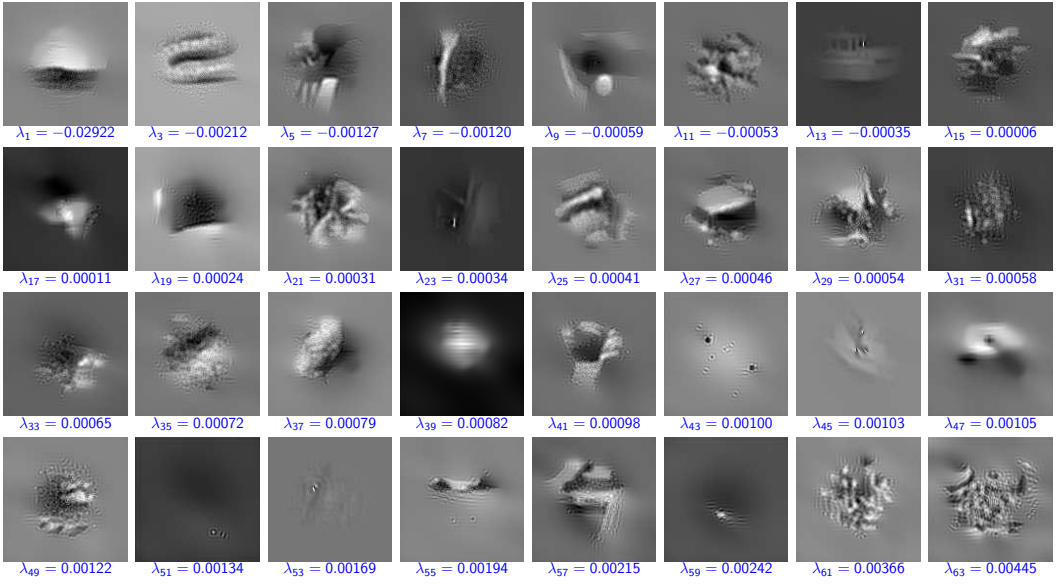
$$\Lambda(v) = \frac{\left\langle \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v), v \right\rangle}{\|v\|_2^2}$$

is generalized Rayleigh quotient.
Optimization by accelerated gradient method with backtracking.

Nonlinear eigenpairs for image denoising



Nonlinear eigenpairs for image deblurring



Overview

Variational Formulation of Linear Inverse Problems

- $x \in \mathbb{R}^{nC}$ restored image (size $n = n_1 \cdot n_2$, C channels)

$$x \in \operatorname*{argmin}_{\widehat{x} \in \mathbb{R}^{nC}} \left\{ \mathcal{E}(\widehat{x}, \theta, z) := \mathcal{D}(\widehat{x}, z) + \mathcal{R}(\widehat{x}, \theta) \right\}$$

- data fidelity term $\mathcal{D}(x, z) = \frac{1}{2}\|Ax - z\|_2^2$ for fixed task-dependent $A \in \mathbb{R}^{lC \times nC}$ and observed data $z \in \mathbb{R}^{lC}$
- total deep variation: parametric deep multi-scale regularizer $\mathcal{R}$ depending on learned training parameters $\theta \in \Theta \subset \mathbb{R}^p$

Variational Formulation of Linear Inverse Problems

- $x \in \mathbb{R}^{nC}$ restored image (size $n = n_1 \cdot n_2$, C channels)

$$x \in \operatorname*{argmin}_{\widehat{x} \in \mathbb{R}^{nC}} \left\{ \mathcal{E}(\widehat{x}, \theta, z) := \mathcal{D}(\widehat{x}, z) + \mathcal{R}(\widehat{x}, \theta) \right\}$$

- data fidelity term $\mathcal{D}(x, z) = \frac{1}{2}\|Ax - z\|_2^2$ for fixed task-dependent $A \in \mathbb{R}^{lC \times nC}$ and observed data $z \in \mathbb{R}^{lC}$
- total deep variation: parametric deep multi-scale regularizer $\mathcal{R}$ depending on learned training parameters $\theta \in \Theta \subset \mathbb{R}^p$

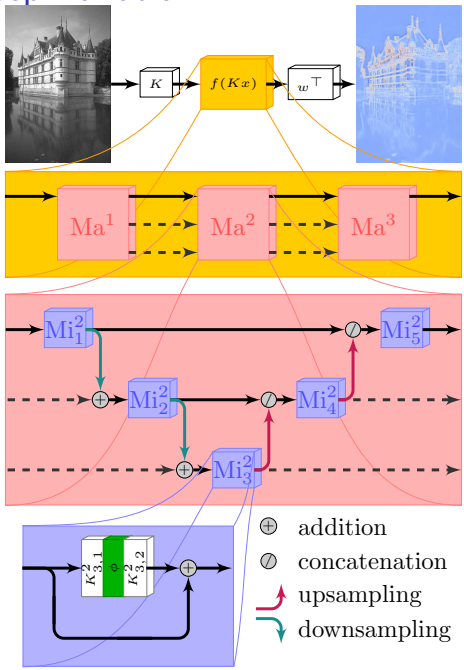
Gradient flow for $t \in (0, T)$:

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t), \theta, z) := -A^\top (A\tilde{x}(t) - z) - \nabla_1 \mathcal{R}(\tilde{x}(t), \theta),$$
$$\tilde{x}(0) = x_{\text{init}}$$

Reparametrization $x(t) = \tilde{x}(tT)$ results in equivalent gradient flow for $t \in (0, 1)$:

$$\dot{x}(t) = Tf(x(t), \theta, z), \qquad x(0) = x_{\text{init}}$$

Total Deep Variation



$K \in \mathbb{R}^{nm \times nC}$ learned convolution kernel ($\sum_{i=1}^{nC} K_{j,i} = 0$ for $j = 1, \ldots, nm$),

$f : \mathbb{R}^{nm} \to \mathbb{R}^{nq}$ multiscale convolutional neural network,

$w \in \mathbb{R}^q$ learned weight vector,

$\theta = (K, K_{s,t}^l, w)$ for $l \in \{1, 2, 3\}$, $s \in \{1, \ldots, 5\}$, $t \in \{1, 2\}$,

$r(x, \theta) := w^\top f(Kx)$,

Sampled Optimal Control Problem

Training set: $N \in \mathbb{N}$ triples $(x_{\text{init}}^i, y^i, z^i)_{i=1}^N$

- $x_{\text{init}}^i \in \mathbb{R}^{nC}$ initial image
- $y^i \in \mathbb{R}^{nC}$ ground truth image
- $z^i \in \mathbb{R}^{lC}$ observed data

Sampled Optimal Control Problem

Training set: $N \in \mathbb{N}$ triples $(x_{\text{init}}^i, y^i, z^i)_{i=1}^N$

- $x_{\text{init}}^i \in \mathbb{R}^{nC}$ initial image
- $y^i \in \mathbb{R}^{nC}$ ground truth image
- $z^i \in \mathbb{R}^{lC}$ observed data

Example (additive Gaussian image denoising): $A = I$, ground truth y^i corrupted by noise $n^i \sim \mathcal{N}(0, \sigma^2) \Rightarrow x_{\text{init}}^i = z^i = y^i + n^i$

Sampled Optimal Control Problem

Training set: $N \in \mathbb{N}$ triples $(x^i_{\text{init}}, y^i, z^i)_{i=1}^N$

- $x^i_{\text{init}} \in \mathbb{R}^{nC}$ initial image
- $y^i \in \mathbb{R}^{nC}$ ground truth image
- $z^i \in \mathbb{R}^{lC}$ observed data

Example (additive Gaussian image denoising): $A = I$, ground truth y^i corrupted by noise $n^i \sim \mathcal{N}(0, \sigma^2) \Rightarrow x^i_{\text{init}} = z^i = y^i + n^i$

Sampled optimal control problem with convex and coercive loss l:

$$\inf_{T \in [0, T_{\max}],\, \theta \in \Theta} \left\{ J(T, \theta) := \frac{1}{N} \sum_{i=1}^{N} l(x^i(1) - y^i) \right\}$$

s.t. state equation for each sample ($i = 1, \ldots, N$ and $t \in (0, 1)$)

$$\dot{x}^i(t) = Tf(x^i(t), \theta, z^i), \qquad x^i(0) = x^i_{\text{init}}$$

Sampled Optimal Control Problem

Training set: $N \in \mathbb{N}$ triples $(x^i_{\text{init}}, y^i, z^i)_{i=1}^N$

- $x^i_{\text{init}} \in \mathbb{R}^{nC}$ initial image
- $y^i \in \mathbb{R}^{nC}$ ground truth image
- $z^i \in \mathbb{R}^{lC}$ observed data

Example (additive Gaussian image denoising): $A = I$, ground truth y^i corrupted by noise $n^i \sim \mathcal{N}(0, \sigma^2) \Rightarrow x^i_{\text{init}} = z^i = y^i + n^i$

Sampled optimal control problem with convex and coercive loss l:

$$\inf_{T \in [0, T_{\max}],\, \theta \in \Theta} \left\{ J(T, \theta) := \frac{1}{N} \sum_{i=1}^{N} l(x^i(1) - y^i) \right\}$$

s.t. state equation for each sample ($i = 1, \ldots, N$ and $t \in (0, 1)$)

$$\dot{x}^i(t) = Tf(x^i(t), \theta, z^i), \qquad x^i(0) = x^i_{\text{init}}$$

Theorem

The minimum in the sampled optimal control problem is attained.

Discretized Optimal Control Problem

Discretized Optimal Control Problem

$$\inf_{T \in [0, T_{\max}],\, \theta \in \Theta} \left\{ J_S(T, \theta) := \frac{1}{N} \sum_{i=1}^{N} l(x_S^i - y^i) \right\}$$

subject to discrete state equation ($s = 0, \ldots, S-1$ and $i = 1, \ldots, N$)

$$x_{s+1}^i = x_s^i - \tfrac{T}{S} A^\top (A x_{s+1}^i - z^i) - \tfrac{T}{S} \nabla_1 \mathcal{R}(x_s^i, \theta),$$
$$x_0^i = x_{\text{init}}^i \in \mathbb{R}^{nC}$$

depth $S \in \mathbb{N}$ a priori fixed

Discretized Optimal Control Problem

Discretized Optimal Control Problem

$$\inf_{T\in[0,T_{\max}],\,\theta\in\Theta}\left\{J_S(T,\theta) := \frac{1}{N}\sum_{i=1}^{N} l(x_S^i - y^i)\right\}$$

subject to discrete state equation ($s = 0,\dots,S-1$ and $i = 1,\dots,N$)

$$\begin{aligned} x_{s+1}^i &= x_s^i - \tfrac{T}{S}A^\top(Ax_{s+1}^i - z^i) - \tfrac{T}{S}\nabla_1\mathcal{R}(x_s^i,\theta),\\ x_0^i &= x_{\text{init}}^i \in \mathbb{R}^{nC} \end{aligned}$$

depth $S \in \mathbb{N}$ a priori fixed
Equivalent state equation: $x_{s+1}^i = \widetilde{f}(x_s^i, T, \theta, z^i)$ with

$$\widetilde{f}(x,T,\theta,z) := (I + \tfrac{T}{S}A^\top A)^{-1}(x + \tfrac{T}{S}(A^\top z - \nabla_1\mathcal{R}(x,\theta)))$$

Discretized Optimal Control Problem

Let $(\overline{T}, \overline{\theta})$ be a pair of optimal control parameters with the corresponding state $\{\overline{x}_s^i\}_{s=0,\dots,S}^{i=1,\dots,N}$. We define the Hamiltonian

$$H : \mathbb{R}^{nC} \times \mathbb{R}^{nC} \times [0, T_{\max}] \times \Theta \times \mathbb{R}^{IC} \to \mathbb{R}$$
$$(x, p, T, \theta, z) \mapsto \langle p, \widetilde{f}(x, T, \theta, z) \rangle.$$

If $\nabla \widetilde{f}(\overline{x}_s^i, \overline{T}, \overline{\theta}, z^i)$ has full rank for all $i = 1, \dots, N$ and $s = 0, \dots, S$, then there exists an adjoint process $\{\overline{p}_s^i\}_{s=0,\dots,S}^{i=1,\dots,N}$ s.t.

$$\overline{x}_{s+1}^i = \nabla_2 H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i), \qquad \overline{x}_0^i = x_{\text{init}}^i,$$
$$\overline{p}_s^i = \nabla_1 H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i), \qquad \overline{p}_S^i = -\frac{1}{N} \nabla l(\overline{x}_S^i - y^i).$$

Finally, the solution is optimal in the sense that

$$\sum_{i=1}^{N} H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i) \geq \sum_{i=1}^{N} H(\overline{x}_s^i, \overline{p}_{s+1}^i, T, \theta, z^i)$$

for all $T \in [0, T_{\max}]$ and $\theta \in \Theta$.

Discretized Optimal Control Problem

discrete adjoint states p_s^i computed via discrete Pontryagin maximum principle:

$$\begin{aligned}\overline{p}_s^i &= \nabla_1 H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i) \\ &= (I - \tfrac{\overline{T}}{S}\nabla_1^2 \mathcal{R}(\overline{x}_s^i, \overline{\theta}))(I + \tfrac{\overline{T}}{S} A^\top A)^{-1} \overline{p}_{s+1}^i, \\ \overline{p}_S^i &= -\frac{1}{N}\nabla I(\overline{x}_S^i - y^i).\end{aligned}$$

Discretized Optimal Control Problem

discrete adjoint states p_s^i computed via discrete Pontryagin maximum principle:

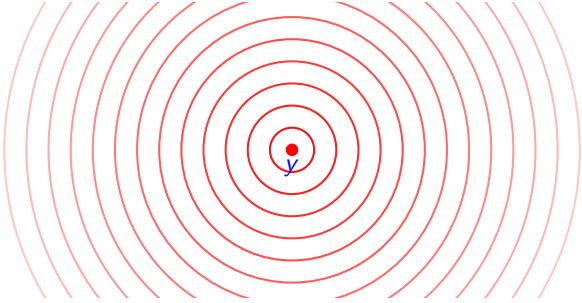
$$\begin{aligned}\overline{p}_s^i &= \nabla_1 H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i) \\ &= (I - \tfrac{\overline{T}}{S}\nabla_1^2 \mathcal{R}(\overline{x}_s^i, \overline{\theta}))(I + \tfrac{\overline{T}}{S} A^\top A)^{-1} \overline{p}_{s+1}^i, \\ \overline{p}_S^i &= -\frac{1}{N}\nabla l(\overline{x}_S^i - y^i).\end{aligned}$$

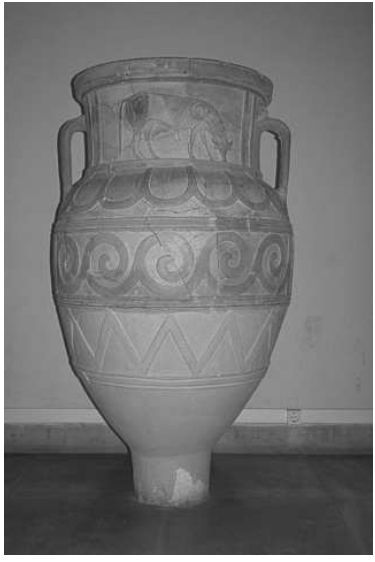
Theorem (Optimality condition)

Let $(\overline{T}, \overline{\theta})$ be a stationary point of J_S with associated states $\overline{x}_s^i$ and adjoint states $\overline{p}_s^i$. We further assume that $\nabla \widetilde{f}(\overline{x}_s^i, \overline{T}, \overline{\theta}, z^i)$ has full rank for all $i = 1, \ldots, N$ and $s = 0, \ldots, S$. Then, we have

$$-\frac{1}{N}\sum_{s=0}^{S-1}\sum_{i=1}^{N}\langle \overline{p}_{s+1}^i, (I + \tfrac{\overline{T}}{S} A^\top A)^{-1}(\overline{x}_{s+1}^i - \overline{x}_s^i)\rangle = 0.$$

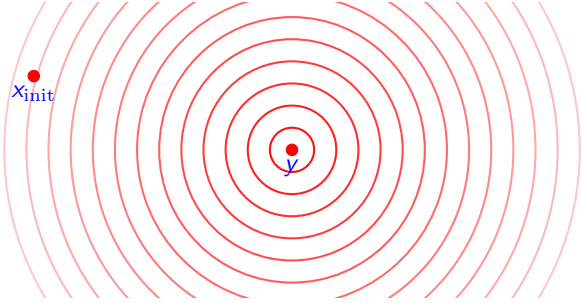
Importance of Early Stopping

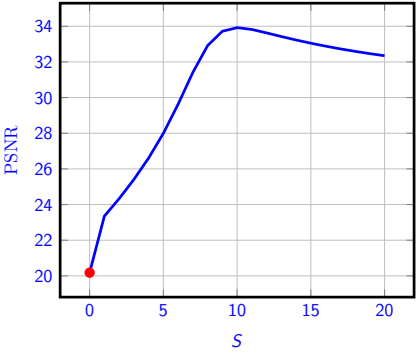


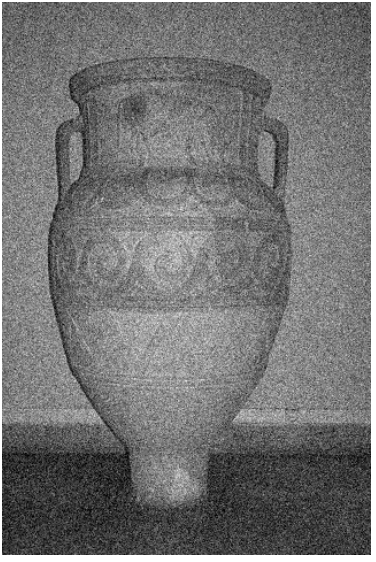


ground truth image

Importance of Early Stopping

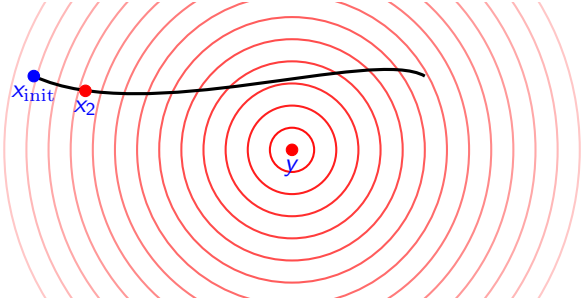


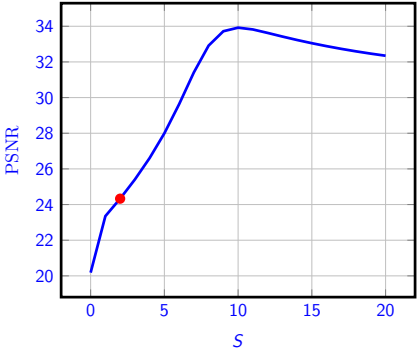


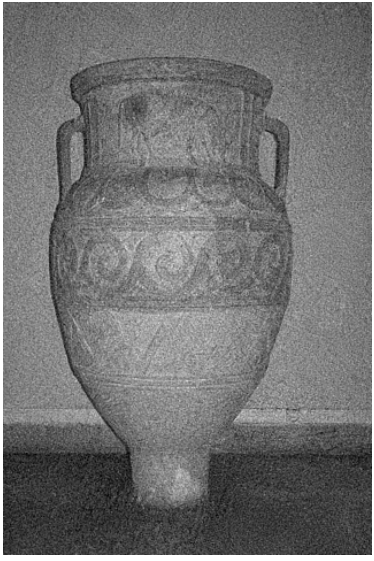


$\text{PSNR} = 20.18$

Importance of Early Stopping

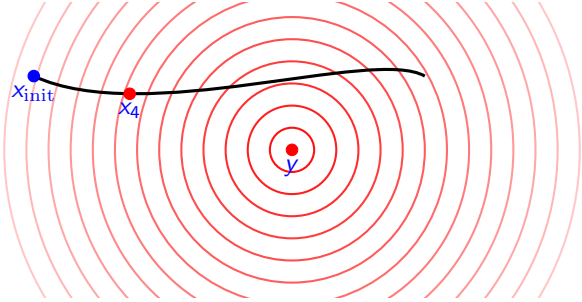


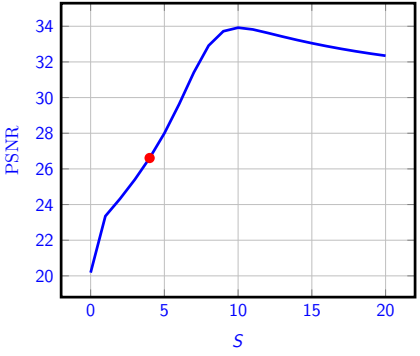


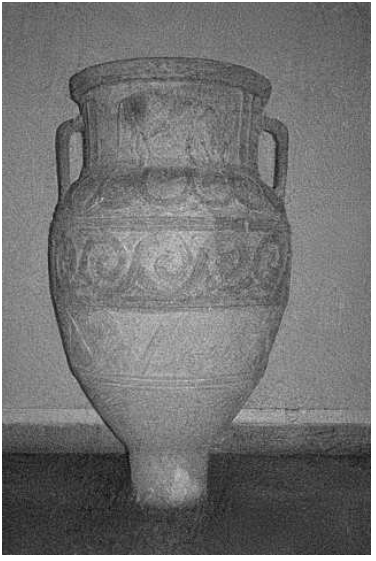


$\text{PSNR} = 24.33$

Importance of Early Stopping

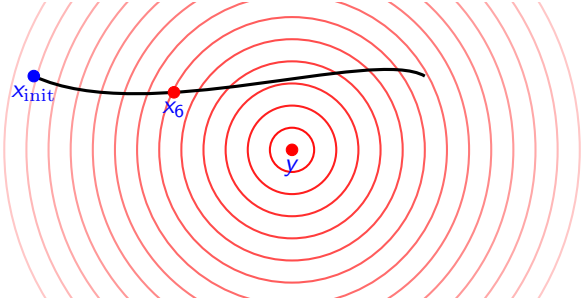


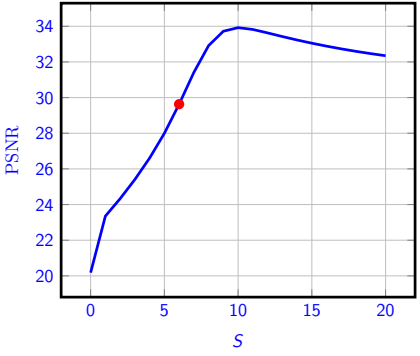


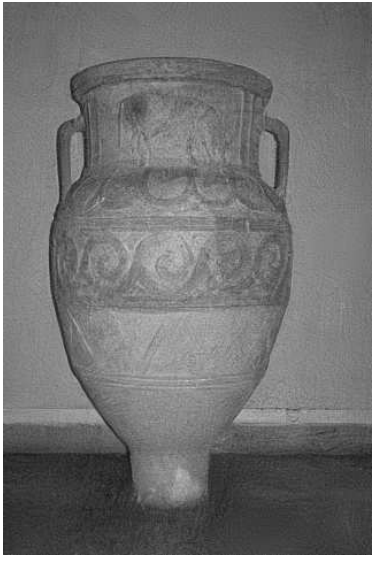


$\text{PSNR} = 26.61$

Importance of Early Stopping

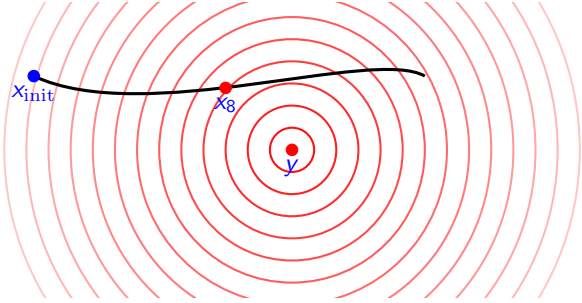


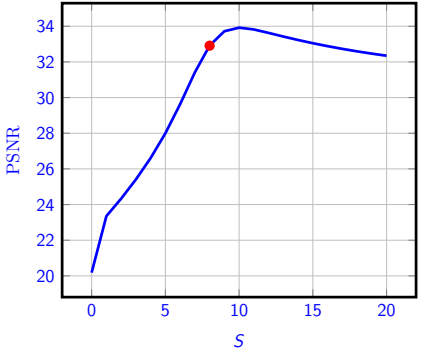




PSNR = 29.63

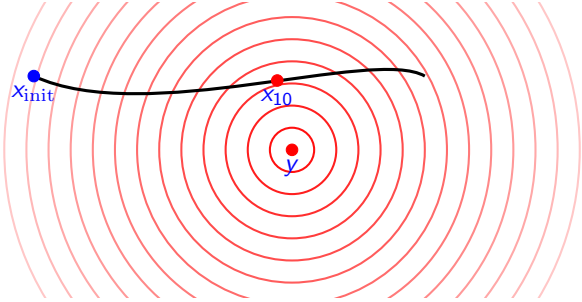
Importance of Early Stopping

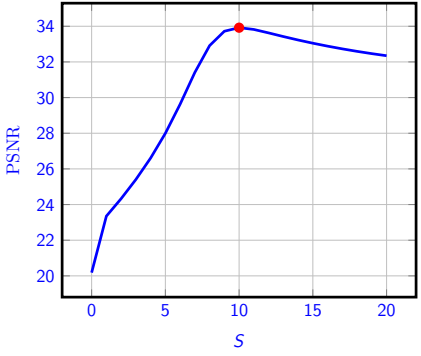




$\text{PSNR} = 32.91$

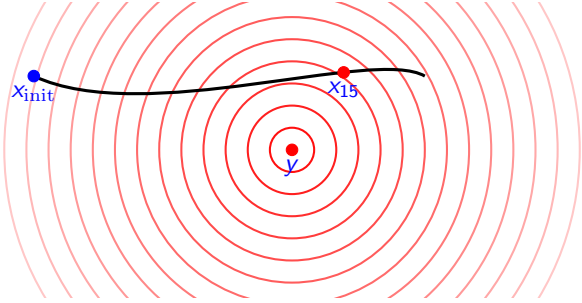
Importance of Early Stopping

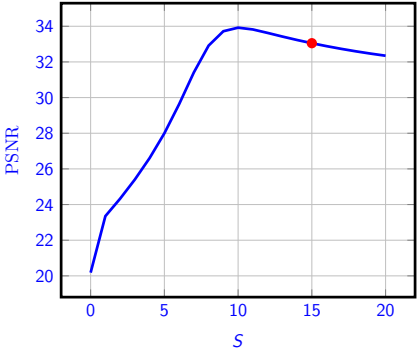




$\text{PSNR} = 33.92$

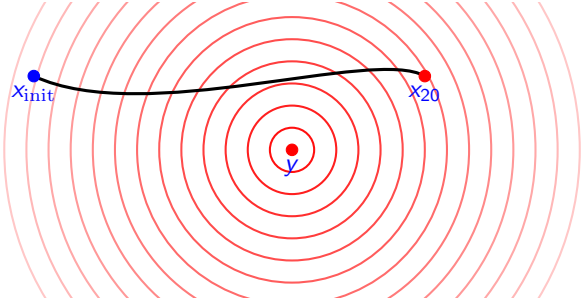
Importance of Early Stopping

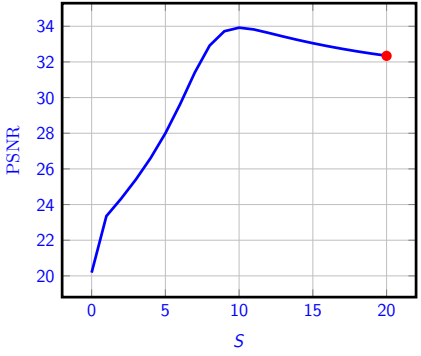




$\text{PSNR} = 33.04$

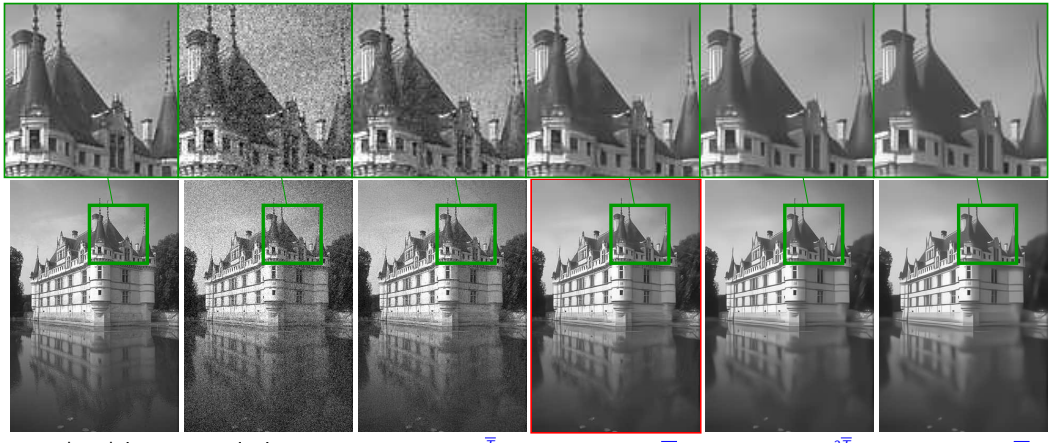
Importance of Early Stopping





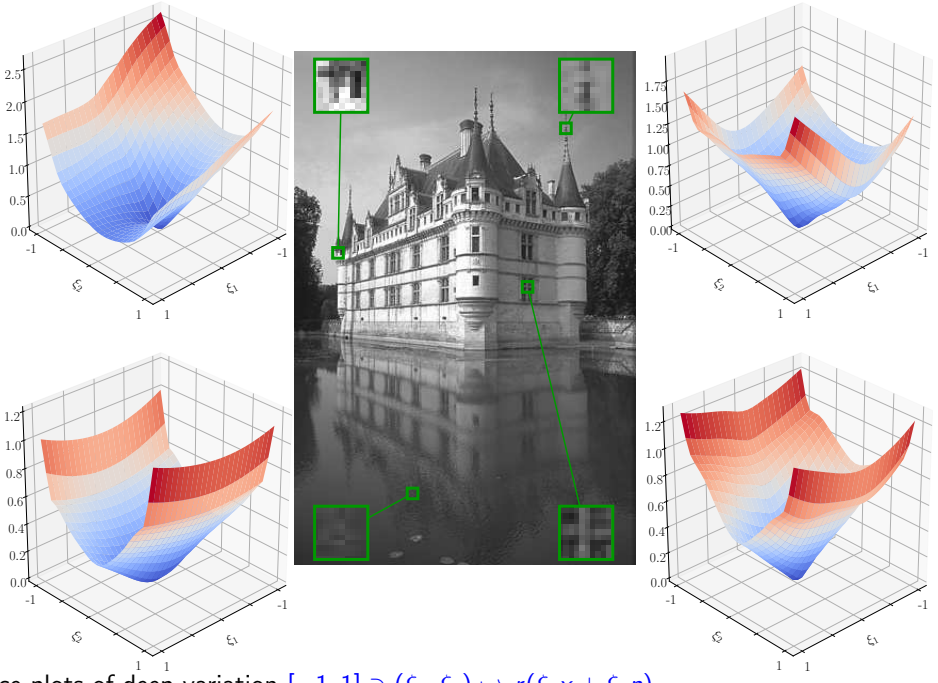
$\text{PSNR} = 32.34$

Numerical Results (Gaussian Image Denoising)



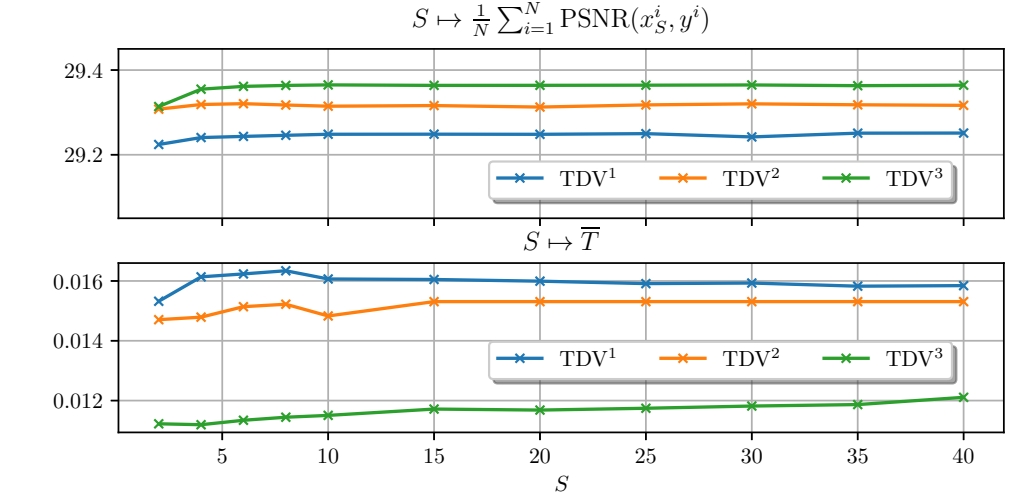
ground truth image | noisy image, PSNR = 20.19 | $S = 5,\ T = \frac{\overline{T}}{2}$, PSNR = 26.66 | $S = 10,\ T = \overline{T}$, PSNR = 30.29 | $S = 15,\ T = \frac{3\overline{T}}{2}$, PSNR = 29.47 | $S = 20,\ T = 2\overline{T}$, PSNR = 28.61

Energy



Surface plots of deep variation $[-1, 1] \ni (\xi_1, \xi_2) \mapsto r(\xi_1 x + \xi_2 n)_i$

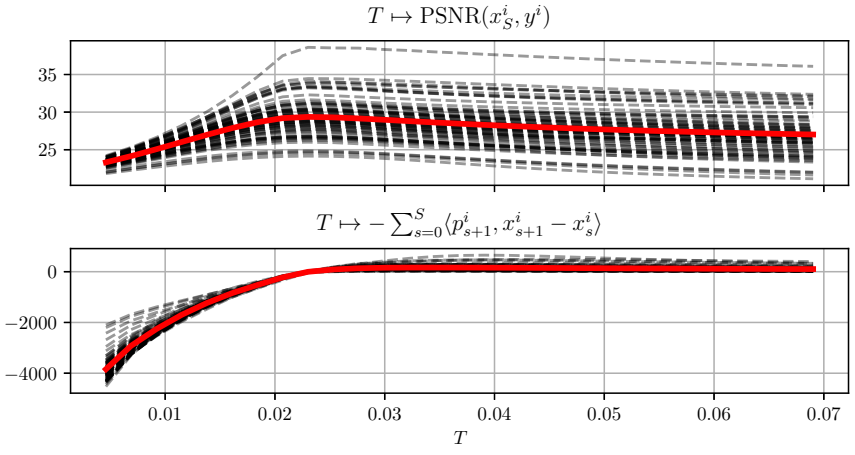
Numerical Results (Gaussian Image Denoising)



Top: $S \mapsto \frac{1}{N}\sum_{i=1}^{N} \mathrm{PSNR}(x_S^i, y^i)$

Bottom: $S \mapsto \overline{T}$

Numerical Results (Gaussian Image Denoising)



Top: $T \mapsto \mathrm{PSNR}(x_S^i, y^i)$

Bottom: $T \mapsto -\sum_{s=0}^{S} \langle p_{s+1}^i, x_{s+1}^i - x_s^i \rangle$

Averages across samples are depicted by red curves

Numerical Results (Gaussian Image Denoising)

Data set	σ	BM3D	TNRD	DnCNN	FFDNet	N^{3}Net	**FOCNet**	TDV^3
Set12	15	32.37	32.50	32.86	32.75	-	**33.07**	33.01
	25	29.97	30.05	30.44	30.43	30.55	**30.73**	30.66
	50	26.72	26.82	27.18	27.32	27.43	**27.68**	27.59
BSDS68	15	31.08	31.42	31.73	31.63	-	**31.83**	31.82
	25	28.57	28.92	29.23	29.19	29.30	**29.38**	29.37
	50	25.60	25.97	26.23	26.29	26.39	**26.50**	26.45
Urban100	15	32.34	31.98	32.67	32.43	-	**33.15**	32.87
	25	29.70	29.29	29.97	29.92	30.19	**30.64**	30.38
	50	25.94	25.71	26.28	26.52	26.82	**27.40**	27.04
# Parameters			26,645	555,200	484,800	705,895	**53,513,120**	427,330

TDV^3 slightly worse than FOCNet (state-of-the-art), but

- FOCNet only applicable for denoising,
- TDV^3 has less than 1 % of the parameters of FOCNet,
- rigorous mathematical theory for TDV^3 available

Understanding TDV

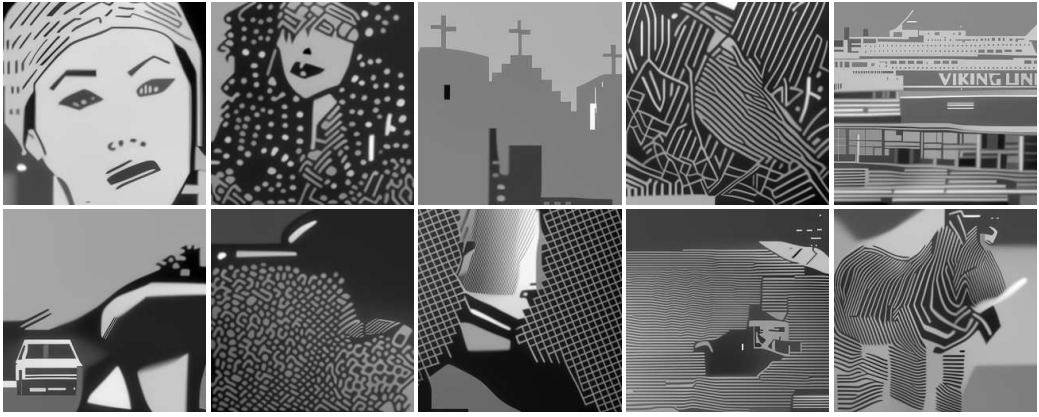
Nonlinear eigenmode analysis for TDV[3]:

$$\overline{x} \in \underset{x}{\operatorname{argmin}}\, \mathcal{R}(x, \theta) \quad \text{s.t. } \|x\|_2 = \|x_{\text{init}}\|_2$$

Understanding TDV

Nonlinear eigenmode analysis for TDV[3]:

$$\overline{x} \in \underset{x}{\operatorname{argmin}}\, \mathcal{R}(x, \theta) \quad \text{s.t. } \|x\|_2 = \|x_{\text{init}}\|_2$$

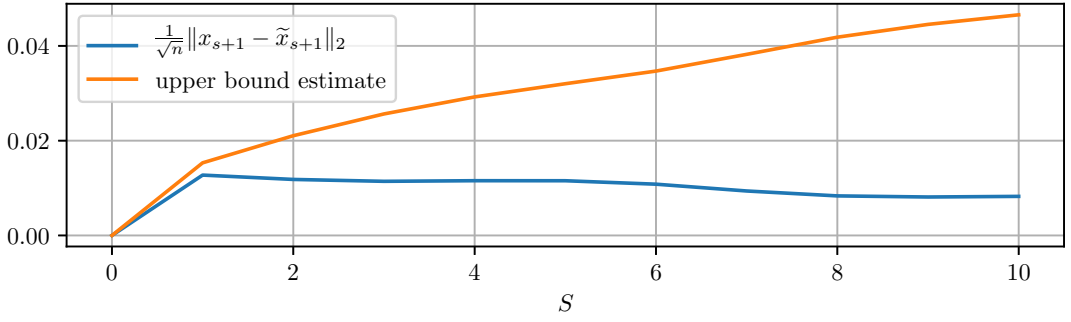


Sensitivity Analysis (Gaussian Image Denoising)

- $(T, \theta), (\widehat{T}, \widehat{\theta})$ two pairs of control parameters (obtained from two different training datasets)
- $x, \widetilde{x} \in (\mathbb{R}^{nC})^{(S+1)}$ two solutions of state equation with same observed data z and initial condition x_{init}, i.e.

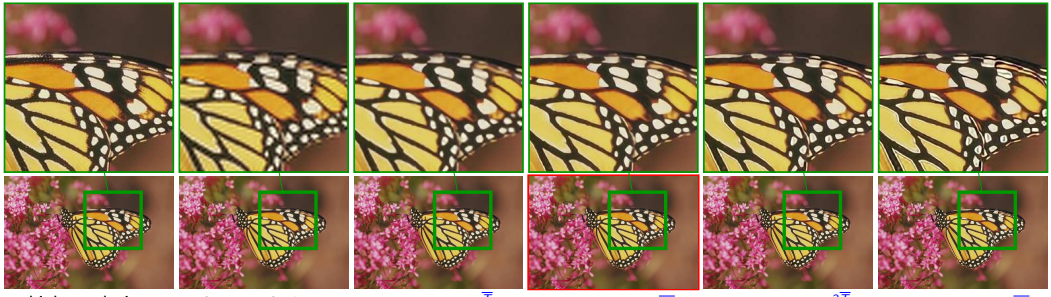
$$x_{s+1} = \widetilde{f}(x_s, T, \theta, z), \quad \widetilde{x}_{s+1} = \widetilde{f}(\widetilde{x}_s, \widehat{T}, \widehat{\theta}, z)$$

  for $s = 1, \ldots, S-1$ and $x_0 = \widetilde{x}_0 = x_{\text{init}}$
- upper bound estimate by ODE theory



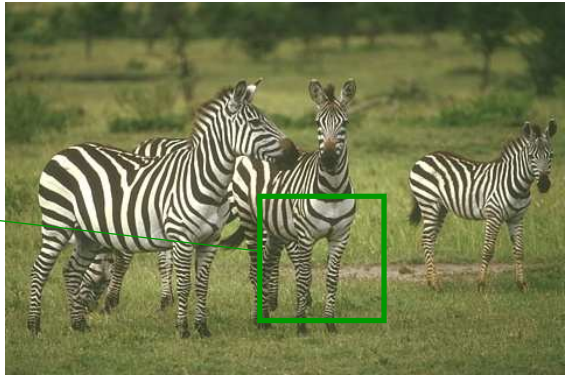
Numerical Results (Single Image Super-Resolution)

- single image super-resolution with scale factor $\gamma \in \{2, 3, 4\}$
- full resolution ground truth image $y^i \in \mathbb{R}^{nC}$
- linear downsampling operator A as implementation of scale factor-dependent interpolation convolution kernel in conjunction with stride
- observed low resolution image $z^i = Ay^i \in \mathbb{R}^{nC/\gamma^2}$



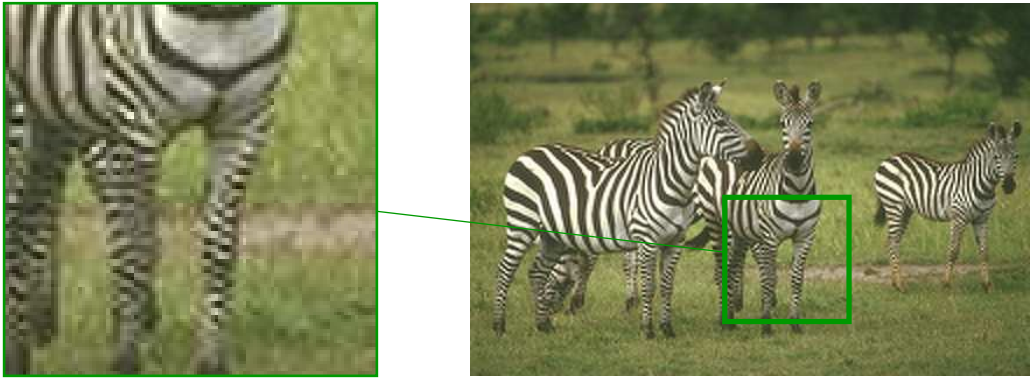
high resolution | low resolution, PSNR = 28.32 | $S = 5$, $T = \frac{\overline{T}}{2}$, PSNR = 31.41 | $S = 10$, $T = \overline{T}$, PSNR = 32.56 | $S = 15$, $T = \frac{3\overline{T}}{2}$, PSNR = 31.24 | $S = 20$, $T = 2\overline{T}$, PSNR = 28.47

Numerical Results (Single Image Super-Resolution)



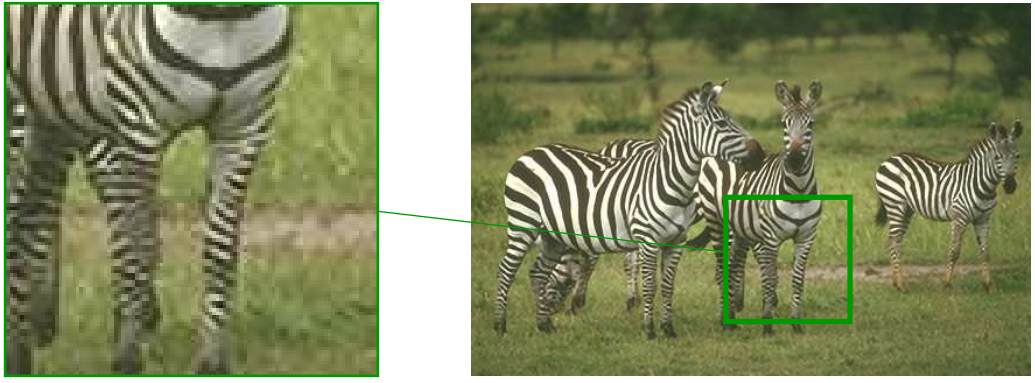
ground truth image

Numerical Results (Single Image Super-Resolution)



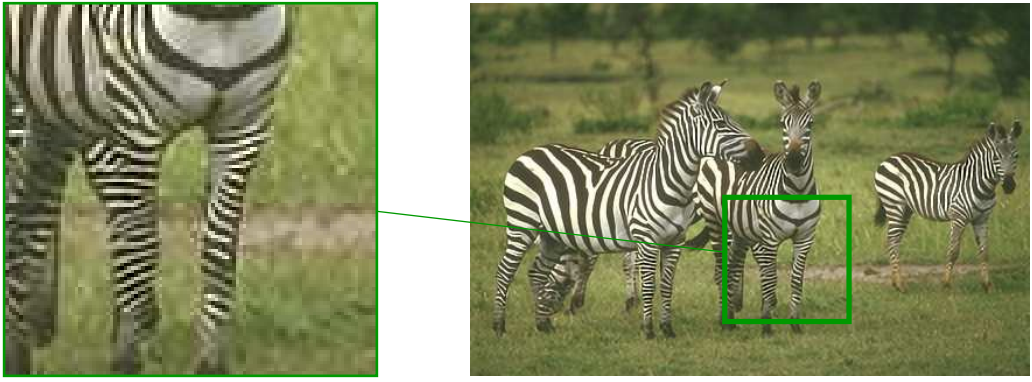
noisy

Numerical Results (Single Image Super-Resolution)



TDV

Numerical Results (Single Image Super-Resolution)



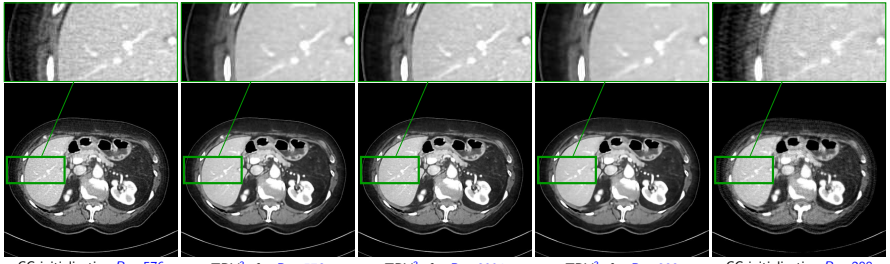
TDV (continued)

Numerical Results (Single Image Super-Resolution)

Data set	Scale	MemNet	VDSR	DRRN	OISR-LF-s	TDV[3]
Set14	×2	33.28	33.03	33.23	33.62	33.35
	×3	30.00	29.77	29.96	30.35	29.96
	×4	28.26	28.01	28.21	28.63	28.41
BSDS100	×2	32.08	31.90	32.05	32.20	32.18
	×3	28.96	28.82	28.95	29.11	28.98
	×4	27.40	27.29	27.38	27.60	27.50
# Parameters		585,435	665,984	297,000	1,370,000	428,970

Transfering TDV to medical imaging

2D CT reconstruction for angular undersampling



CG initialization $R = 576$
PSNR = 34.17

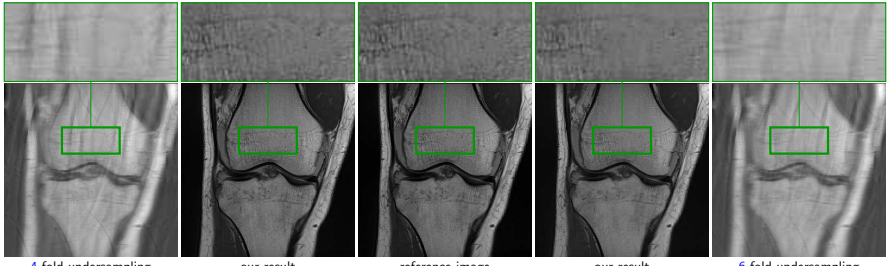
TDV^3_{25} for $R = 576$
PSNR = 43.51

TDV^3_{25} for $R = 2304$

TDV^3_{25} for $R = 288$
PSNR = 35.03

CG initialization $R = 288$
PSNR = 27.75

undersampled MRI reconstruction



4-fold undersampling
PSNR = 24.98

our result
PSNR = 48.29

reference image

our result
PSNR = 38.55

6-fold undersampling
PSNR = 24.07

Overview

Learning with graphical models

- Finally, we consider learning parameters with graphical models for image labeling.
- For many years, both the unary terms and binary terms have been computed based on handcrafted functions.
- Cannot compete with recent deep-learning methods.
- We propose to ...

Learning with graphical models

- Finally, we consider learning parameters with graphical models for image labeling.
- For many years, both the unary terms and binary terms have been computed based on handcrafted functions.
- Cannot compete with recent deep-learning methods.
- We propose to ...

(i) Learn neural networks that compute $\theta = (\theta_i, \theta_{i,j})$ from the input.

(ii) Adopt the graphical model as an inference layer in the network.

Application to Stereo

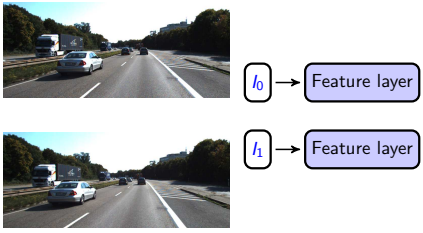
- Given a stereo image pair, compute the disparity (inverse depth)

I_0

I_1

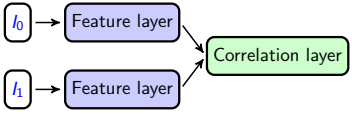
Application to Stereo

- Given a stereo image pair, compute the disparity (inverse depth)



Application to Stereo

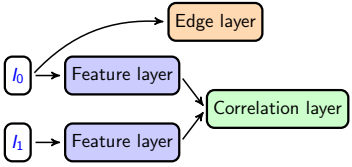
- Given a stereo image pair, compute the disparity (inverse depth)



$$\theta_i(x_i)$$

Application to Stereo

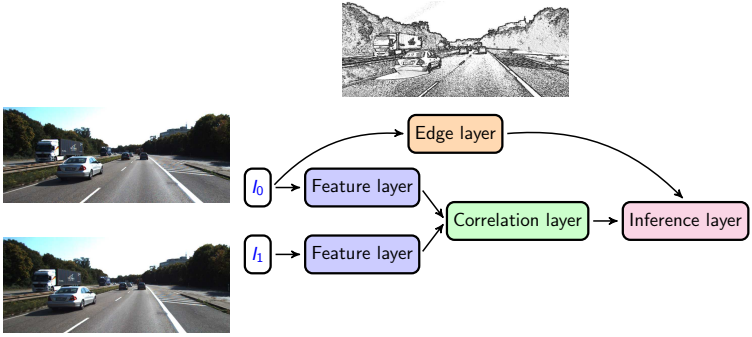
- Given a stereo image pair, compute the disparity (inverse depth)



$\theta_i(x_i)$ $\theta_{i,j}(x_i, x_j)$

Application to Stereo

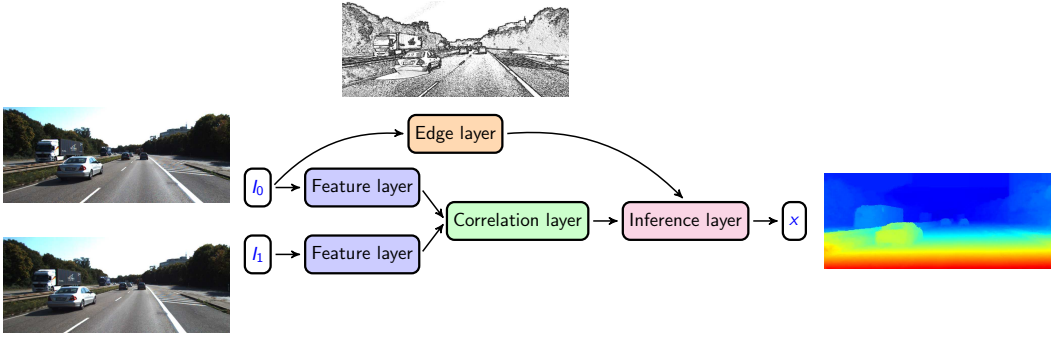
- Given a stereo image pair, compute the disparity (inverse depth)



$$\sum_{i \in \mathcal{V}} \theta_i(x_i) \; + \; \sum_{(i,j) \in \mathcal{E}} \theta_{i,j}(x_i, x_j)$$

Application to Stereo

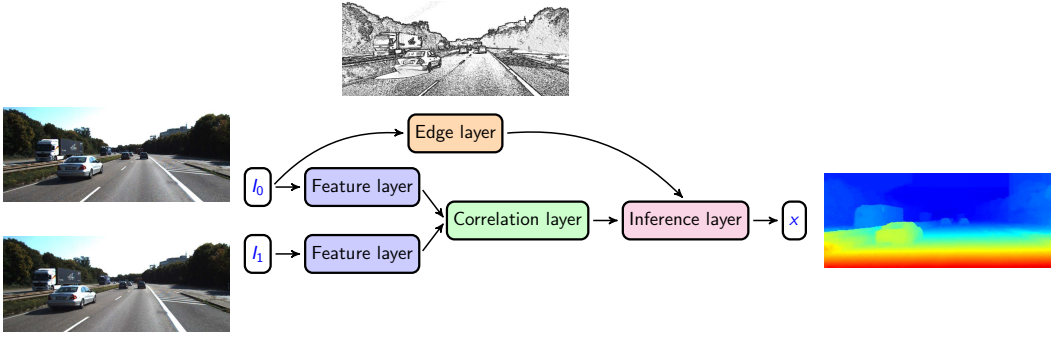
- Given a stereo image pair, compute the disparity (inverse depth)



$$\min_{x \in \mathcal{L}} E(x, \theta) := \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{i,j}(x_i, x_j)$$

Application to Stereo

- Given a stereo image pair, compute the disparity (inverse depth)



$$\min_{x \in \mathcal{L}} E(x, \theta) := \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{i,j}(x_i, x_j)$$

- Two main issues:

(i) Efficient solution of the inference layer

(ii) End-to-end learning

Learning

- For learning the parameters ϑ of the neural network, we consider a loss function ℓ that compares the output of the image labeling problem $x(\vartheta)$ with the ground truth labels $x^\dagger$, e.g.

$$\ell(x(\vartheta), x^\dagger) = \left\| x(\vartheta) - x^\dagger \right\|_1$$

- The learning problem represents a bilevel optimization problem:

$$\min_{\vartheta} \ell(x(\vartheta), g), \quad \text{s.t. } x(\vartheta) \in \arg\min_{x \in \mathcal{L}} \in E(x, f(\vartheta)),$$

- Hard to solve, because $x(\vartheta)$ does not continuously depend on ϑ.
- We approximate the problem by constructing a differentiable upper bound similar to the structured output SVM [Tsochantaridis et al. '04]

Convex upper bound

- We use the following chain of upper bounds:

$$\max_{x \in \arg\min_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) \quad \leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger,f)} \ell(x, x^\dagger)$$

Convex upper bound

- We use the following chain of upper bounds:

$$\begin{aligned}\max_{x \in \arg\min_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) &\leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\ &\leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f)\end{aligned}$$

Convex upper bound

- We use the following chain of upper bounds:

$$\begin{aligned}
\max_{x \in \arg\min_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) &\leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\
&\leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
&\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f)
\end{aligned}$$

Convex upper bound

- We use the following chain of upper bounds:

$$\begin{aligned}
\max_{x \in \arg\min_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) &\leq \max_{x \in \mathcal{L} : E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\
&\leq \max_{x \in \mathcal{L} : E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
&\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
&= \psi(\hat{x}, x^\dagger, f)
\end{aligned}$$

  where $\hat{x} = \arg\max_{x \in \mathcal{L}} \ell(x, x^\dagger) - E(x, f)$.

- The function ψ is linear in f (in the lifted space), hence it is a maximum over linear functions $\rightsquigarrow$ convex.

Convex upper bound

- We use the following chain of upper bounds:

$$\begin{aligned}\max_{x \in \arg\min_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) &\leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\ &\leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\ &\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\ &= \psi(\hat{x}, x^\dagger, f)\end{aligned}$$

  where $\hat{x} = \arg\max_{x \in \mathcal{L}} \ell(x, x^\dagger) - E(x, f)$.

- The function ψ is linear in f (in the lifted space), hence it is a maximum over linear functions $\rightsquigarrow$ convex.
- Computing the upper bound requires to solve the labeling problem but with loss-augmented unary terms.
- The resulting surrogate function is differentiable with respect to the unaries f_i

$$D_{f_i} \psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i) \quad \rightsquigarrow$$

Convex upper bound

- We use the following chain of upper bounds:

$$\begin{aligned}\max_{x \in \arg\min_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) &\leq \max_{x \in \mathcal{L} : E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\ &\leq \max_{x \in \mathcal{L} : E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\ &\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\ &= \psi(\hat{x}, x^\dagger, f)\end{aligned}$$

  where $\hat{x} = \arg\max_{x \in \mathcal{L}} \ell(x, x^\dagger) - E(x, f)$.

- The function ψ is linear in f (in the lifted space), hence it is a maximum over linear functions $\rightsquigarrow$ convex.
- Computing the upper bound requires to solve the labeling problem but with loss-augmented unary terms.
- The resulting surrogate function is differentiable with respect to the unaries f_i

$$D_{f_i}\psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i) \quad \rightsquigarrow \quad \underbrace{D_{f_i}\psi(\hat{y}, y^\dagger, f) = y_i^\dagger - \hat{y}_i}_{\text{in the lifted space}}$$

Convex upper bound

- We use the following chain of upper bounds:

$$\begin{aligned}\max_{x \in \arg\min_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) &\leq \max_{x \in \mathcal{L} : E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\ &\leq \max_{x \in \mathcal{L} : E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\ &\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\ &= \psi(\hat{x}, x^\dagger, f)\end{aligned}$$

  where $\hat{x} = \arg\max_{x \in \mathcal{L}} \ell(x, x^\dagger) - E(x, f)$.

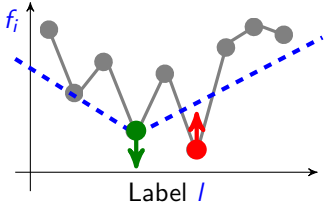
- The function ψ is linear in f (in the lifted space), hence it is a maximum over linear functions $\rightsquigarrow$ convex.
- Computing the upper bound requires to solve the labeling problem but with loss-augmented unary terms.
- The resulting surrogate function is differentiable with respect to the unaries f_i

$$D_{f_i}\psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i) \quad \rightsquigarrow \quad \underbrace{D_{f_i}\psi(\hat{y}, y^\dagger, f) = y_i^\dagger - \hat{y}_i}_{\text{in the lifted space}}$$

- Similar formula for the binary weights $f_{i,j}$.

Graphical Explanation

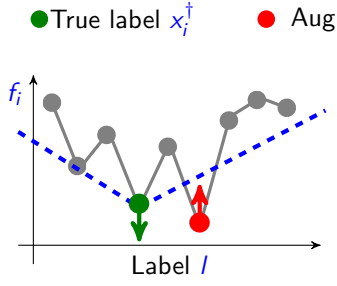
True label $x_i^\dagger$ Augmented label $\hat{x}_i$ Loss margin



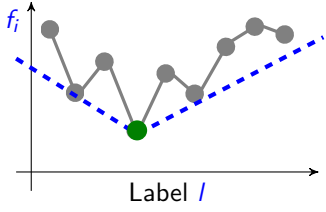
$\rightsquigarrow$

$D_{f_i}\psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i)$

Graphical Explanation



$\rightsquigarrow$



$$D_{f_i}\psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i)$$

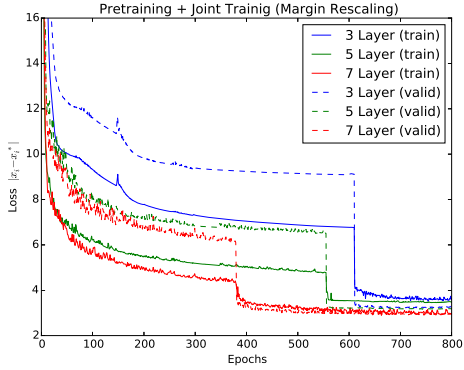
Training

Data bases

- Middlebury Stereo - Version 3
- KITTI 2015

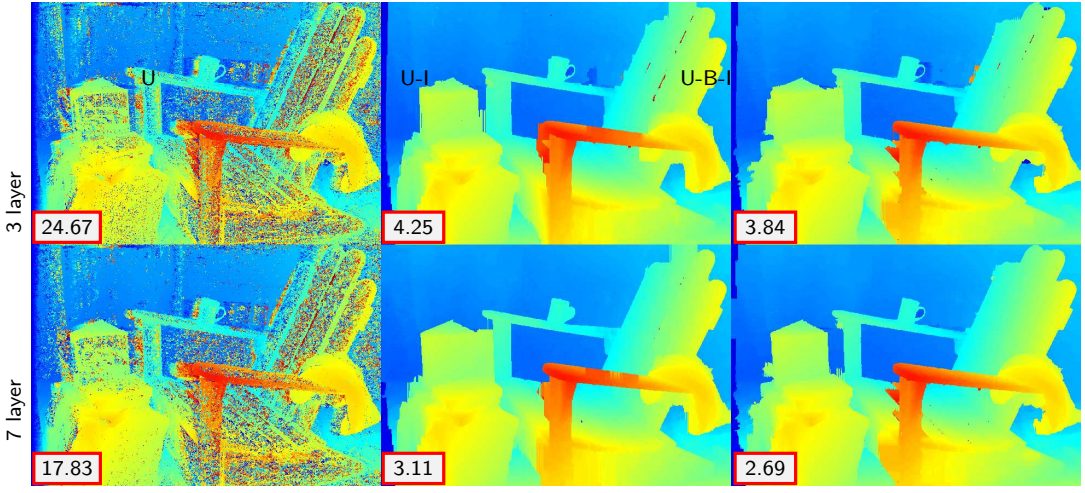
Training

- Learning is performed using stochastic subgradient descent with momentum
- First, we perform a CNN-only pre-training, followed by a joint training

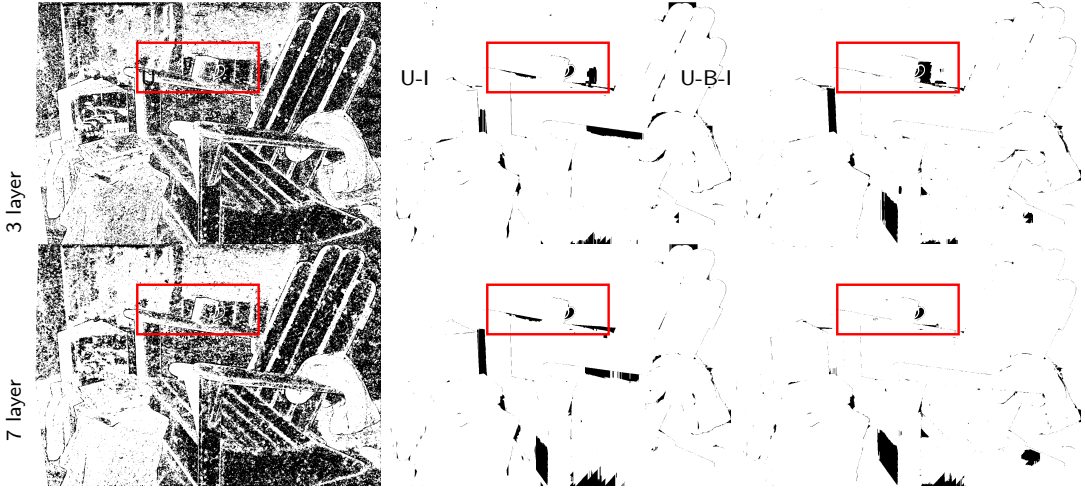


Benchmark	Method	CNN	+CRF	+Joint	+PW
Middlebury	CNN3	23.89	11.18	9.48	9.45
	CNN7	18.58	9.35	8.05	7.88
Kitti 2015	CNN3	28.38	6.33	6.11	4.75
	CNN7	13.08	4.79	4.60	4.04
	[28]	5.99	4.31	-	-
	[55]	13.56	4.45	-	-

Experiments - Middlebury Stereo

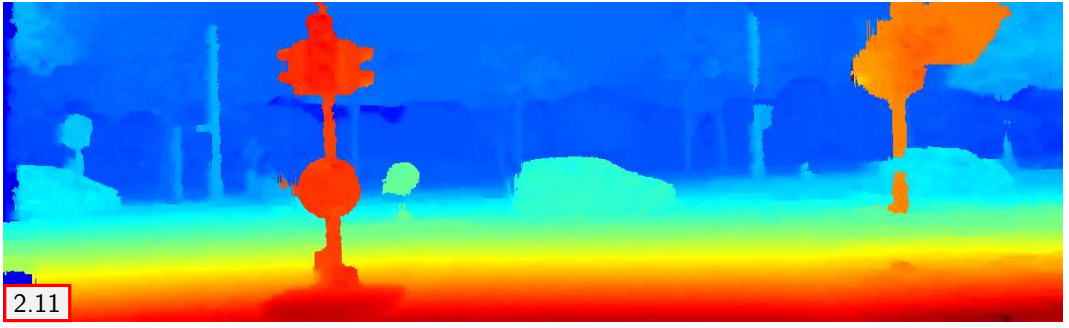


Experiments - Middlebury Stereo



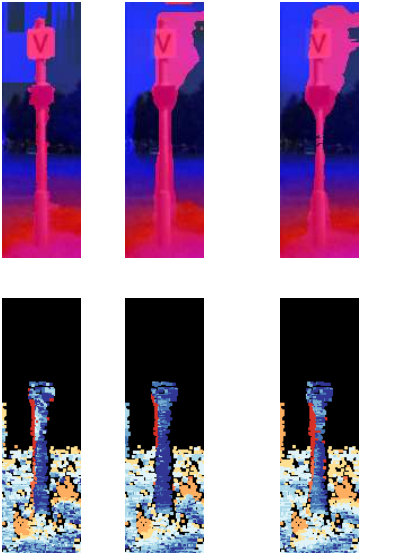
Experiments - Kitti 2015

Experiments - Kitti 2015



Kitti 2015 - Quality of groundtruth

Ours MC-CNN ContentCNN



Ours MC-CNN ContentCNN

Extension to motion estimation (Sintel)